

# ACOUSTICAL AWARENESS FOR INTELLIGENT ROBOTIC ACTION

A Dissertation  
Presented to  
The Academic Faculty

By

Eric Martinson

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in Computer Science

Georgia Institute of Technology

December, 2007

Copyright © 2007 by Eric Martinson

# ACOUSTICAL AWARENESS FOR INTELLIGENT ROBOTIC ACTION

Approved by:

Dr. Ronald Arkin, Advisor  
School of Interactive Computing  
*Georgia Institute of Technology*

Dr. Frank Dellaert  
School of Interactive Computing  
*Georgia Institute of Technology*

Dr. David Anderson  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. Thad Starner  
School of Interactive Computing  
*Georgia Institute of Technology*

Dr. Tucker Balch  
School of Interactive Computing  
*Georgia Institute of Technology*

Date Approved: November 9, 2007

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my wife, Lilia Moshkina, for the countless hours she spent helping me out in many different ways. She was always encouraging, helped out with the robots when it was needed, was a good sounding board for refining ideas, and was an excellent editor for this dissertation. In general, Lilia was invaluable to this work, and this dissertation would likely not have been completed without her.

Second, I would like to thank my advisor, Ron Arkin, along with the rest of my committee: Dave Anderson, Tucker Balch, Frank Dellaert, and Thad Starner. Ron provided me with the freedom to explore this subject and truly identify a topic of my own choosing, while the committee was enthusiastic and helpful all the way to the end.

Finally, I would like to thank the people at the Naval Research Laboratory Center for Artificial Intelligence. In particular, thanks to Alan Schultz and Derek Brock for their early acceptance of this line of research. Their experience and general support was a great asset towards refining the work and getting it published. Thanks also to Ben Fransen, a fellow intern, for being a patient listener on numerous occasions throughout two summers.

# TABLE OF CONTENTS

<b>Acknowledgements</b>	<b>iii</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>x</b>
<b>Summary</b>	<b>xv</b>
<b>CHAPTER 1 - Introduction</b>	<b>1</b>
<b>1.1 Terminology</b>	<b>3</b>
<b>1.2 Research Question</b>	<b>4</b>
<b>1.3 Contributions</b>	<b>6</b>
<b>1.4 Dissertation Overview</b>	<b>6</b>
<b>CHAPTER 2 – Related Work</b>	<b>8</b>
<b>2.1 Sound Source Localization</b>	<b>9</b>
<b>2.2 Natural Language Interfaces</b>	<b>17</b>
<b>2.3 Audio Classification</b>	<b>23</b>
<b>2.4 Audio Probing</b>	<b>26</b>
<b>2.5 Sound Vocalization</b>	<b>27</b>
<b>2.6 Summary of Application Domains</b>	<b>29</b>

<b>CHAPTER 3 – Acoustical Awareness</b>	<b>30</b>
3.1 Types of Awareness	31
3.2 Knowledge for Acoustical Awareness	35
3.3 Mathematical Framework for Sound Propagation	42
3.4 Chapter Summary	66
<b>CHAPTER 4 – Acquiring Knowledge about the Auditory Scene</b>	<b>69</b>
4.1 Building Blocks	70
4.2 Representations for Characterizing Sound Sources	84
4.3 Path Information	141
4.4 Building Maps without Models	153
4.5 Chapter Summary	158
<b>CHAPTER 5 – The Autonomous Mobile Security Robot</b>	<b>162</b>
5.1 Related Work in Security Robotics	163
5.2 Monitoring the Auditory Scene	165
5.3 Improving the Signal-To-Noise Ratio	203
5.4 Chapter Summary	230
<b>CHAPTER 6 – The Stealthy Approach Scenario</b>	<b>232</b>
6.1 How to Hide a Noisy Robot	235

6.2	Experimental Results _____	242
6.3	Discussion of Results _____	253
6.4	Chapter Summary _____	265
<b>CHAPTER 7 – Acoustical Awareness for Human-Robot Interaction</b>		<b>267</b>
7.1	A Model of Human Acoustical Awareness _____	269
7.2	Robotic Adaptations _____	274
7.3	Combining Types of Awareness _____	285
7.4	Chapter Summary _____	288
<b>CHAPTER 8 – Summary and Contributions</b>		<b>290</b>
8.1	How Can Acoustical Awareness be Applied to Mobile Robotics?	291
8.2	Contributions _____	301
8.3	Conclusion _____	305
<b>Appendix A - Software Design</b>		<b>307</b>
A.1.	Hardware _____	307
A.2.	Software Processes _____	309
A.3.	Database Design _____	318
<b>Appendix B - Knowledge Gathering Tools</b>		<b>330</b>
B.1.	Spatial Likelihoods _____	330

<b>B.2.</b>	<b>Auditory Evidence Grids</b>	<b>333</b>
<b>B.3.</b>	<b>Directivity Models</b>	<b>339</b>
<b>B.4.</b>	<b>Mel Frequency Cepstral Coefficients</b>	<b>342</b>
<b>B.5.</b>	<b>Creating Direct Field Maps</b>	<b>346</b>
<b>B.6.</b>	<b>Ray-Tracing for Direct and/or Reverberant Field Maps</b>	<b>347</b>
<b>B.7.</b>	<b>Sampled Data Noise Maps</b>	<b>354</b>
<b>Appendix C - Guiding Robotic Movement</b>		<b>358</b>
<b>C.1.</b>	<b>Clear-Space Map</b>	<b>358</b>
<b>C.2.</b>	<b>Patrolling the Environment</b>	<b>360</b>
<b>C.3.</b>	<b>Investigation of a Sound Source</b>	<b>368</b>
<b>Appendix D - HRI Application</b>		<b>372</b>
<b>D.1.</b>	<b>Selecting Speech Volume</b>	<b>374</b>
<b>D.2.</b>	<b>Pausing for Interruptions</b>	<b>375</b>
<b>D.3.</b>	<b>Rotating to Face the Listener</b>	<b>376</b>
<b>D.4.</b>	<b>Relocating the Robot</b>	<b>377</b>
<b>References</b>		<b>380</b>

## LIST OF TABLES

Table 4.1. The results of all phase 1 auditory evidence experiments. _____	94
Table 4.2. Mean localization error when auditory evidence grids are used with data collected by a moving robot. _____	113
Table 4.3. Mean localization and orientation error as produced by the discovery process. _____	115
Table 4.4. Mean error in identifying the direction of maximum volume, as produced by an area coverage task. _____	124
Table 4.5. Localization and orientation accuracy of the two source discovery process	126
Table 5.1. List of trials completed by the robot for this scenario. All used the same patrol route, but varied in the types and numbers of active sources in the environment. _____	174
Table 5.2. The relative performance of using the proposed maximum likelihood approach for detecting each type of source in the environment as a new source. _____	179
Table 5.3. Illustrates successes in detecting and localizing radios playing music, compared across different environment types. _____	185
Table 5.4. Performance of both the detection and localization algorithms for environments with at least one new sound source present. _____	187
Table 5.5. Summary of belief states used for each patrol run through the environment.	194
Table 5.6. Results of the source change detection algorithms, compared across different numbers of changes in the environment. _____	196
Table 5.7. Average reduction in noise levels using different relocation strategies to avoid music sources _____	211
Table 5.8. Results of the adaptive waypoint following algorithm averaged over the entire path. _____	225
Table 5.9. Results of the adaptive waypoint following algorithm for a $3 \times 3\text{-m}^2$ region in front of the sound source. _____	226
Table 6.1. Acoustic hiding results in the presence of a 67-dB radio source. _____	246

Table 6.2. Acoustic hiding results in the presence of a 67-dB radio source and a loud reverberant field. _____	248
Table 6.3. Acoustic hiding results in the presence of a 67-dB radio source located near a wall. _____	251
Table 6.4. Acoustic hiding results in the presence of a 54-dB filter source. _____	252

## LIST OF FIGURES

Figure 3.1. Basic reactive acoustically aware system. _____	32
Figure 3.2 Reactive acoustically-aware system with behavioral coordination. _____	32
Figure 3.3 Hybrid architecture for supporting acoustical awareness. _____	34
Figure 3.4. 3D model of the Aware Home Laboratory used for estimating reverberation effects and general sound propagation. _____	39
Figure 3.5 Direct vs. indirect paths from source to receiver. _____	45
Figure 3.6 Relating acoustic entities to sound fields when building noise estimates. _	48
Figure 3.7 The information from each acoustic entity necessary for building a direct field estimate. _____	51
Figure 3.8 Modeling the image source method. _____	53
Figure 3.9. The information from each acoustic entity necessary for building a reverberant field estimate. _____	60
Figure 3.10. An example outer wall of a house used in estimating transmission of sound. _____	62
Figure 3.11. The information from each acoustic entity necessary for building a transmitted sound estimate. . _____	64
Figure 4.1. The B21r mobile robot and the obstacle map it created using the continuous localization algorithm . _____	72
Figure 4.2. The Pioneer2-dxe robot and the map of the Mobile Robot Lab at Georgia Tech created using the PMAP software. _____	75
Figure 4.3. Measurable time delay between signals arriving at each microphone vs. the angle of incidence. _____	77
Figure 4.4. A contour plot of a spatial likelihood result for detecting human speech. Light areas are considered more likely. . _____	82
Figure 4.5. Contour plot of an auditory evidence grid localizing two radios. _____	90
Figure 4.6. Fully equipped B21r mobile robot used for phase 1 testing. _____	92

Figure 4.7. Spatial evidence grid used by the robot for localization with source positions shown relative to the obstacle positions in the room. _____	93
Figure 4.8. Auditory evidence grids localizing two speech sources (a stationary human speaker and a tape player ) from 463 data points collected at 6 positions. ____	98
Figure 4.9. Stepping through the iterative clustering process, first round. _____	103
Figure 4.10. Stepping through the iterative clustering process, second round. _____	105
Figure 4.11. Stepping through the iterative clustering process, conclusion. _____	108
Figure 4.12. An overlay of the NRL environment, showing an example waypoint path, set of area coverage target points, and source locations. _____	111
Figure 4.13. Auditory evidence grid created from 137 samples collected during a directed investigation of a source using an area coverage heuristic. _____	116
Figure 4.14. Comparison of robot-created directivity models using different reverberation assumptions, with a hand-measured directivity model. _____	120
Figure 4.15. Hand coded obstacle map used by the pioneer for navigation in an environment with two sources. _____	125
Figure 4.16. Direct field estimates created from a single source of arbitrary volume_	127
Figure 4.17. Process of creating sound propagation models from sampled area coverage data, part 1. _____	129
Figure 4.18. Process of creating sound propagation models from sampled area coverage data, part 2. _____	130
Figure 4.19. Process of creating sound propagation models from sampled area coverage data, part 3 - combined direct field for both sources _____	131
Figure 4.20. Weight vs frequency plot of a mel-scale filter bank. _____	133
Figure 4.21. Classification results vs. predicted direct field volumes for two sources, a filter and a fountain, at regular intervals around the room. _____	138
Figure 4.22. Comparison of a spatial evidence grid collected by the robot for localization purposes to a thresholded evidence grid used for reverberant field estimates_	143
Figure 4.23. Maps of sound propagation created using a 2D robot-created evidence grid of the obstacles in the environment. _____	146

Figure 4.24. Map of the auditory scene combining a simplified direct field model with the reverberant field.	151
Figure 4.25. Estimated sound levels for the combined direct and reverberant fields, created from purely robot-collected information.	153
Figure 4.26. Comparison of the interpolated noise map (left) to the sound fields model (right) for the fan source.	156
Figure 4.27. Comparison of the interpolated noise map (left) to the sound fields model (right) for the radio source.	157
Figure 4.28. Demonstration of the effects of poor reverberation models in the NRL AI Center.	157
Figure 5.1. Pictures of the sound sources dominating the auditory scene in the Mobile Robot Laboratory for the acoustic monitoring task.	168
Figure 5.2. The obstacle layout used for the acoustic monitoring task. Within this environment, there were two sources whose positioned never changed.	170
Figure 5.3. Discretized obstacle map through which a patrol route has been identified.	172
Figure 5.4. Graphical comparison of different relocation strategies the robot can use to avoid a sound source when correcting for a poor initial acoustic location.	207
Figure 5.5. Positions of the 3 different ambient noise sources and radios within the testing environment for improving a poor initial acoustic location.	210
Figure 5.6. Predicted noise map of the poor initial acoustic location testing area modeling the effects of the three ambient noise sources on the auditory scene. This map assumes that each source is omni-directional.	211
Figure 5.7. Environment for testing the improved SNR movement strategies.	215
Figure 5.8. Paths taken by the different movement strategies overlayed on the robot-discovered noise map: the path through the grid-cell centers, the path chosen to avoid loud locations.	218
Figure 5.9. A noise map created from hand collected samples is converted to a vector field representation, where strength is indicated by arrow size.	219
Figure 5.10. The behavioral controller used to reactively follow gradients along a waypoint path.	220

Figure 5.11. Direct field map created from hand-measured data used in testing the improved SNR movement strategies. _____	223
Figure 5.12. Direct field maps created from two different robot-measured data sets used in testing the improved SNR movement strategies. _____	224
Figure 5.13. Histogram of all data volumes in a 3x3-m <sup>2</sup> region in front of the radio source collected by the robot during the improved SNR movement strategy trials. _	227
Figure 6.1. Contour map of the estimated noise at the observer due to the robot. ____	238
Figure 6.2. Contour map showing estimated impact on an observer due to a robot at any reachable location in the environment. _____	241
Figure 6.3. Layout of the acoustic hiding scenario. _____	243
Figure 6.4. Comparison of the angular detection energy observed while the robot was taking the shortest path vs. the acoustic hiding path for the first environmental layout containing a 67-dB source. _____	246
Figure 6.5. Bar chart comparing the angular detection energy recorded by the observer for each robot path. _____	247
Figure 6.6. The second environmental layout used to test acoustic hiding performance. _____	249
Figure 6.7. Comparison of the angular detection observed while the robot was taking the shortest path vs. the acoustic hiding path for the second environmental layout containing a 67-dB source. _____	251
Figure 6.8. Comparison of the angular detection observed while the robot was taking the shortest path vs. the acoustic hiding path for the second environmental layout containing a 54-dB source. _____	253
Figure 6.9. Change in total volume plot, as predicted by ray-tracing models, for the first room layout with no ambient noise sources. _____	257
Figure 6.10. Sound intensity profile at the observer's location due to a 67-dB radio. _	259
Figure 6.11. Contour plot of the revised approach to estimating directional cues using ray-tracing. _____	260
Figure 6.12. Contour plot of the maximum angular impact of the robot for scenario 2 with a loud reverberant field. _____	262
Figure 6.13. General shape of the volume vs. frequency plot for sounds masked by a single tone. _____	264

Figure 7.1. B21R robot from iRobot, outfitted with a four microphone overhead array, bicolops stereo vision system, and monitor for visual feedback.	276
Figure 7.2. Stereo vision results.	278
Figure 7.3. The sequence of steps the robot takes while reading a story to a human listener.	286
Figure A.1. The network configuration of the 4 computers used in the acoustically aware experiments.	308
Figure A.2. The software configuration used for acoustically-aware navigation in this dissertation.	310
Figure A.3. Groups of information in the database are grouped by the entities they relate to: sound sources, environments, listeners, and representations of the auditory scene.	318
Figure A.4. A graphical description of the tables/relationships that make up the sampled data entity in the database.	320
Figure A.5. The three tables describing environmental information in the database.	323
Figure A.6. Tables storing sound source information in the database.	325
Figure A.7. Summary of the database implementation used in this dissertation. All of the tables seen in previous sections are included.	329
Figure C.1. A Finite State Automaton guiding a robot through a series of three arbitrary waypoints in the environment.	361

## SUMMARY

With the growth of successes in pattern recognition and signal processing, mobile robot applications today are increasingly equipping their hardware with microphones to improve the set of available sensory information. However, if the robot, and therefore the microphone, ends up in a poor location acoustically, then the data will remain noisy and potentially useless for accomplishing the required task. This is compounded by the fact that there are many bad acoustic locations through which a robot is likely to pass, and so the results from auditory sensors often remain poor for much of the task.

The movement of the robot, though, can also be an important tool for overcoming these problems, a tool that has not been exploited in the traditional signal processing community. Robots are not limited to a single location as are traditionally placed microphones, nor are they powerless over to where they will be moved as with wearable computers. If there is a better location available for performing its task, a robot can navigate to that location under its own power. Furthermore, when deciding where to move, robots can develop complex models of the environment. Using an array of sensors, a mobile robot can build models of sound flow through an area, picking from those models the paths most likely to improve performance of an acoustic application.

In this dissertation, we address the question of how to exploit robotic movement. Using common sensors, we present a collection of tools for gathering information about the auditory scene and incorporating that information into a general framework for acoustical awareness. Thus equipped, robots can make intelligent decisions regarding control strategies to enhance their performance on the underlying acoustic application.

# CHAPTER 1

## INTRODUCTION

Audition on mobile robots has long been passed over in favor of vision, the argument being that if we could only decipher an image, then vision has all of the data necessary for highly successful navigation. But the proponents of audition have been successfully reversing this trend in recent years by arguing that there is a wealth of information available to the robot outside the narrow confines of a camera's view-space. If nothing else, the omni-directionality of incoming acoustic information can be used to direct more data-rich directional sensors to intriguing or suspicious locations. Beyond that, researchers are also adding microphones to augment human-robot interfaces [Fong et al. 2003], improve security [Huang et al. 1997], localize themselves [Martinson and Dellaert 2003; Hu et al. 2006], and a variety of other applications.

For robots, audition is a relatively young field. Elsewhere, however, it is by no means understudied. Electrical engineering and digital signal processing (DSP) have made great strides over the last 30 years in using static mounted microphones, hand-held microphones, and microphone arrays. It is a great testament to their success in areas such as speech recognition, classification, and source localization that roboticists are now considering equipping their mobile platforms with these sensors. But as researchers have discovered with other sensory modalities, microphones on robots constitute a different problem than other microphone scenarios.

Where traditional microphone mountings have often been subject to environmental interference ranging from ambient noise, to high and low frequency

echoes, and overlapping sound sources, robots add their own set of problems to the list. Many of the techniques developed to counter these problems, including filters, *do* work on mobile robots, but they are not as successful when the platform: (1) moves around the environment, changing its proximity to different sources; (2) generates its own noises, wheel and motor, which vary with the executed action; and (3) has limited computational and power resources, but needs to process the data in real time. These inherent problems of mobile robotics, combined with the general problems associated with using microphones, produce daunting obstacles confronting the developers of acoustic applications for these platforms.

Mobile robotics though has unique advantages all its own, which have not been exploited in the traditional signal processing community. The key advantage is that robots can move. They are not limited to a single location as are traditional microphone mountings, nor are they powerless over to where they will be moved as with wearable computers. If there is a better location for performing their task, they can navigate to that location under their own power. Furthermore, we are not limited to a single robot. Robot teams add extra dimensions of control, by allowing fully dynamic microphone arrays that are not limited by a rigid internal structure, nor stuck in randomly distributed locations. The potential that mobility alone adds to acoustical applications is enormous, but we first need to figure out how to best exploit that potential.

In this work, it is our supposition that acoustical awareness is the key to successful development of mobile robotic applications involving sound. Acoustical awareness is defined here as the coupling of action with knowledge about the acoustic environment, where said knowledge could be in the form of maps, rules, measurements,

predictions, or anything that indicates how sound flows or will flow through the environment. The underlying premise is that the more acoustical knowledge the robot uses, the better its global performance will be on an acoustic application. The questions of how much knowledge is necessary, and how it is to be integrated into the robotic controller are central to the proposed research.

## 1.1 TERMINOLOGY

*Acoustics* is defined as “the science of sound”<sup>1</sup>. Both auditory (listening) and sound generating applications are acoustic, because they work with sound. In the grand picture, the two areas differ by focusing on either the receiver or the source. In either case, the same principles of sound propagation through an environment apply, and an acoustically-aware application would require much of the same information.

*Acoustical Awareness* is the coupling of action with knowledge about the acoustic environment, specifically anything that indicates how sound flows or will flow in the physical world.

*Audition* refers to the act of hearing. Like cameras with vision, microphones are the instruments we employ for recording sound, defined as “the mechanical energy transmitted by longitudinal pressure waves in a material medium (like air).”<sup>2</sup> As with computer vision, however, we are not limited to what can be sensed by people. While humans can hear sounds in a frequency range from 20-20000 Hz, microphones can be used to “listen” to much lower or higher frequencies.

---

<sup>1</sup> Raichel, D. p.4

<sup>2</sup> <http://www.webster.com>, Accessed 8/29/04

The *Auditory Scene* contains all aspects of that which effect what a listener somewhere in the environment can hear. It includes the sound sources generating the noise, the environment through which the sound travels, and, ultimately, the listener itself.

*Noise* is an application-specific term referring to unwanted sound. Even if the sound is desired or needed for some other purposes, but is interfering with the intended application, it is called noise.

The *Soundscape* refers to that which can be heard. Although often used interchangeably with the term *Auditory Scene*, the soundscape is a narrower definition, referring specifically to what can be heard at any location in the environment, independent of the listener.

*Vocalization* refers to the creation of sound by the robot. The emitted sound could be speech, or just noise. It could target either human or robot listeners, or may not target anyone, as does the incidental creation of noise which often accompanies mechanical motion.

## 1.2 RESEARCH QUESTION

*How can acoustical awareness be effectively incorporated into a navigational controller?*

In exploring this principal question, we address the following set of three subsidiary questions:

- **What a priori information or sensory data is useful for a mobile robot performing an acoustic application?**

Acousticians have developed a large body of research on the flow of sound. Potentially everything about the environment, ranging from construction material performance and architectural features, to speaker and microphone models, is useful for a mobile robot. However, not all of it is feasibly acquired, much less usable, given the limited computational, or acoustic processing resources onboard the robot. Therefore, we need to determine which information can be reasonably collected for, or by, a mobile robot, and whether or not that information is usable in a given situational context.

- **How can we combine sensory data from multiple sources to build effective representations of the acoustic environment?**

Using just a single microphone located on the robotic platform provides a wealth of sensory information available for assisting navigation: sound pressure level, the frequencies present and their loudness, impulse noises vs. continuous streams, classification results, etc. Beyond that a priori knowledge of the environment, other sensory data such as vision, other microphones, or even data from other robotic platforms might be available. Somehow this data needs to be fused together to build effective and coherent representations of the acoustic environment in which the robot resides.

- **How does acoustical awareness change with control over the source vs. the receiver?**

Vocalization and audition vary only in their control over the sound source. In vocal applications, the robot itself is the source. In auditory applications, the robot is the receiver. In some instances, a robot may need to be both. In many cases, both forms

share the same goal, which is to control the sound being heard by the receiver (human or microphone). How these goals are achieved may also be similar across vocalization and audition.

### **1.3 CONTRIBUTIONS**

It is believed that the pursuit of answers to these questions should result in significant contributions to the mobile robotics community, with application to other fields including signal-processing, acoustical engineering, and human-computer interfaces. In particular, this research will provide:

- A conceptual and architectural framework for incorporating acoustical awareness into a navigational controller.
- A novel approach for the storing, retrieval, and fusion of acoustic knowledge for mobile robotic applications.
- Guidelines for applying the resulting framework to acoustic applications. In particular, matching the data available to the task at hand given a particular situational, intentional, and environmental context.

### **1.4 DISSERTATION OVERVIEW**

This dissertation is described with 8 chapters. Chapter 2 covers the types of acoustical applications for robots, and what work has already been done. Chapter 3 discusses the nature of being acoustically aware, identifying what information is needed to understand sound flow in the environment, and how it will be used. Chapter 4 then delves into the robotic question of representations and control for acquiring the necessary information using a mobile platform. Chapters 5-7 describe applications of acoustical

awareness to different robotic application domains, including robotic security, stealth robots, and human-robot interaction. Finally, Chapter 8 summarizes the dissertation and outlines the contributions from this dissertation.

## CHAPTER 2

### RELATED WORK

Acoustic applications on mobile robots, while increasing in number, are still relatively rare. But with the ready availability of microphones, and the increasing processing power of computers and even microcontrollers, the area is primed for an application explosion. One sure sign of this potential is the growing interest in acoustical domains by robotics hobbyists. Voice commands [Williams 2004], noise following [Predko 2003], and sound synthesis [Jones et al. 1999], are all popular applications in the area. Still, the work by hobbyists tends to be overly simple algorithmically. But if enough people become interested in the area, then we will see microphones and speakers on robots become commonplace.

What is making acoustical applications in robotics difficult is the underlying complexity of the acoustical domain. The soundscape is always changing with time, more so than even the visual domain tends to, and the sensors currently available for sampling the soundscape are noisy and only capture a relatively small selection of the soundscape. Additionally, the soundscape itself is not straightforward, and varies significantly from environment to environment, even when the same types of noise sources are present. Altogether, this creates a very hostile perceptual domain for a robot, which is already struggling to successfully handle routine navigational tasks.

In order to overcome these problems in the acoustic domain, much research in mobile computing (including robotics), as well as biology, and digital signal processing, has concentrated on developing task specific solutions that take advantage of the nature

of sound flow through the environment in order to improve performance. This chapter presents the current state of the art in robot acoustics. The general organization of the chapter is by broad application categories explored in current research: specifically, sound source localization, natural language interfaces, classification, vocalization, and audio probing.

## **2.1 SOUND SOURCE LOCALIZATION**

Sound source localization applications are primarily concerned with identifying where a sound is coming from, including determining its exact position, or simply the angles to the source(s), or distance estimates. Most sound localization work is not even concerned with what is making the sound, and in fact, a common laboratory assumption is that whatever sound is being heard is the one that the robot is interested in. This category is probably where the greatest amount of acoustic research on mobile robots has occurred to date.

The localization problem can be roughly divided into two areas of interest. The first problem is localizing individual sources in the environment. The second lies in creating maps of sound sources, possibly for a robot to localize itself using a set of detectable noise sources. In either of these areas, an awareness of sound propagation through the environment may potentially assist improving accuracy, reducing false positive responses, and improving the general applicability of the developed algorithms.

### **2.1.1 TRADITIONAL LOCALIZATION**

Localizing sources in the environment may not seem like a difficult task to a person, but that is because the mechanisms for doing so are built into a humans' auditory

system. People are equipped with two ears for the purpose of localization, as are most mammals and birds. With two receivers (ears) physically separated from each other on opposite sides of the head, sound arrives at different times, phases, and intensities in each ear. These inter-aural time differences (ITD), phase differences, and intensity differences (IID) can be used to calculate left-to-right angular location of the sound. There are different models of exactly how these features are calculated biologically [Jeffress 1948; Shamma 1989], but these are the physical properties available in the incoming sound stream, and both models make use of them.

If there are three or more receivers, then ITD's alone can estimate the angle of incidence for an arbitrary source location, and this has in fact been used on a number of robots [Yamasaki 1995; Huang et al. 1997; Young and Scanlon 2001]. With only two receivers, however, ITD's can provide only an  $180^\circ$  estimate on the horizontal plane towards the location of the sound. Localizing on front-to-back or elevation is not possible without additional information. The biological solution is found in the shape of the head. Sounds traveling around the back of the head arrive at different times and intensities than when they come from the front. The pinna, or fleshy parts of the ears, also filter or focus sounds depending on which direction they come in from. The shape of the head and the pinna make up the Head Related Transfer Function, or HRTF. The HRTF is a function, different for each individual, which the brain learns in order to pick out localization cues that would otherwise be hidden. Although the use of an HRTF does not guarantee the same resolution at all angles around the head, it does effectively localize sounds from any direction and has been applied in humanoid robotics [Nakadai 2003; Hornstein et al. 2006].

Noise, however, remains a problem with ITD-based solutions. Environmental noise, as well as robot-generated ego noise, can generate misleading measurements or mask the signal of interest. A common feature, for this reason, among many robotics solutions is the use of higher quality microphones with frequency ranges that limit interference, and to mount them high above the robot base, away from motors and other noise-producing equipment. Even then, the noise can cause problems. Furthermore, on some platforms and with some applications, it is not feasible to deploy expensive microphones far from sources of robot ego-noise. Humanoid platforms are a case in particular. Most current humanoid robots are limited to inexpensive microphones mounted within the head, close to the internal machinery of the robot.

A sound source localization solution that has been put forward by a couple of different robotics groups is to physically move the microphones. Barbara Webb [Webb 1998] explored this approach using small robots with 2 microphones. The robot would always turn in the direction of the highest volume, and although the path was not entirely straight, the robot could find the sound source when echoes or obstacles did not interfere. Another more traditional solution using ITD's, was conducted by Nakadai et. al. [Nakadai 2001], which would turn their humanoid robot's head in the direction of the loudest sound as estimated by ITD's. By doing so, the robot could effectively ignore the internal noises when they do not change direction with the rotation, allowing the robot to accurately focus on the noise source.

In addition to robot and environmental noise, another problem confronting real-time implementations of sound localization algorithms is that the accuracy of ITD's is limited in highly echoic environments. If an echo bounces off the nearby floor, or wall,

then separating it from the real signal is very difficult, and leads to errors in localization. Unfortunately most indoor environments (especially hard floored ones), unless specially padded, have this echo problem in one place or another. The most common solution in robotics is to add another sensor with a different vulnerability. Cameras are the obvious alternative, as a microphone can assist in orienting a camera towards the object of interest [Huang et al. 1997; Strobel 2001; Blisard et al. 2007], although heat sensors have also been used when applicable [Yamasaki 1995]. Noise remains a problem with the auditory sensing, but its use has been restricted to initializing the application.

Without adding another sensor, another solution that has worked well in echoic environments is to make use of the precedence effect. The precedence effect is a psychoacoustic phenomenon where an acoustic signal arriving first at the ears, suppresses the ability to hear any other signals for the next 40ms, in order to reduce the interference with echoes. Initial work by Jie Huang [Huang 1997] generated robots that were successfully able to locate visually obstructed sound sources in different rooms using ITDs, but still resorted to vision as the primary sensor once the object became visible. More recent work by Martin Heckmann [Heckmann et al. 2006], however, has improved on this early work for binaural microphone arrays to dramatically reduce the need for the visual localization on their Honda robot, Asimo. Their model of the precedence effect combines interaural time, intensity, and envelope differences with an echo suppression mechanism to allow Asimo to localize a human speaker under a large variety of ambient noise and environmental conditions.

### 2.1.2 CREATING AUDITORY SPATIAL MAPS

Up until this point, we have been assuming that the sound localization algorithm returns angular measurements to the robot. While, theoretically, ITD-based algorithms could also estimate distance from the robot, the closer together the microphones are, the greater the error in the distance measurement. People, too, have a similar problem in estimating distance for similar reasons (the ears are too close together). Nevertheless, if there are multiple sound sources in a room, people are reasonably good at localizing them in space, if not instantaneously, then at least over time. To explain how people and animals can be good at this despite the signal processing limitations, some researchers in biology have proposed the notion of an auditory spatial map that maps multiple sensing modalities in the brain into a single ego-centric representation of stimuli position.

The barn owl has one of the best-studied auditory systems of any species. Researchers of the Owllab at the University of Oregon have, as stated on their webpage, concentrated on "studying the neural mechanisms of auditory localization in barn owls", addressing questions such as how the owl localizes, how it reacts to multiple sound sources [Takahashi and Keller 1994], what the HRTF [Keller et al. 1998] is, etc. What they have discovered is that within the inferior colliculus of the brain, individual neurons become attached to specific locations in the surrounding environment, only firing when a noise is determined to have originated from that location. These neuronal spatial maps, however, are not being constructed from auditory information alone. Hyde and Knudson at Stanford University claim that the Optic Tectum (OT), which directs the eyes toward sensory cues, is critical for the construction of an auditory map [Hyde 2000]. Their studies with barn owls have indicated that without the OT, the owl's brain is no longer

capable of updating and aligning the separate maps of space. Hyde and Knudson have also constructed a general framework for the calibration process that applies to both avian and mammalian species [Hyde 2000].

Rucci, Tononi and Edelman [Rucci et al. 1997] at the Neurosciences Institute in San Diego have developed an alternative model specifically for the barn owl, arguing that the existing model does not include enough information. Since the barn owl cannot see without turning its head, they argue that the model should actually be a sensorimotor model. The system takes as input ITDs and visual information to construct the visual and auditory maps. The model developed is neuronal and is trained using "value-dependent learning" and tested in simulation under a variety of inputs.

While this neuronal spatial map of the auditory scene suggests a tantalizing approach for mobile robotics, it should be understood that this is not the only biological model of the auditory scene. In some animals, including people, recent research has failed to reveal any notion of spatial maps [Goldstein 2007]. In some other animals, such as guinea pigs, maps have been found at an early age, but seem to disappear as they get older [Ingham et al. 1998]. This does not invalidate the idea of an auditory spatial map, as the barn owl and the guinea pig have very clearly demonstrated the existence of such maps in some species, but it does suggest that there may be more than one way to track or at least store this information in our brains.

In robotics, however, the idea of localizing all of the sound sources in space relative to the robot has more often been developed as a straightforward extension of the earlier angular sound localization problem, rather than as a mapping problem. To alleviate the problem of microphone proximity in determining distance, the natural

solution is to distribute the microphones over a wider area of the environment. Using a microphone array embedded into the walls of a room [Nakadai et al. 2006], some researchers were able to localize multiple simultaneously talking human speech sources, and determine the directivity. This information can then be passed along to a mobile robot for interactive or avoidance purposes. A similar setup in a home environment [Bian et al. 2005] extracted 3-dimensional coordinates of sound sources and estimated the type human activity occurring. Another possibility, when the environment cannot be engineered, is to place the microphones on separate robots [Girod and Estrin 2001]. As the robots separate in space, differences in arrival time become more pronounced, so distance measurements should ideally become more accurate. But for this technique to work the audio streams from the separate microphones need to be synchronized to better than millisecond accuracy. For each millisecond of error in synchronization, roughly 34-cm of error are introduced in the sound source location. Furthermore, the robots need to be localized accurately in space (overhead cameras and/or laser-based obstacle maps), relative to each other, or the time-delay measurements will be incorrectly determined, even with accurate synchronization.

While research into accurate localization of the sound sources remains an active field, there is also work progressing in the opposite direction. If the robot has a map of where the sounds are as it moves through the environment, then the robot could localize itself using the map and the sounds that it can currently perceive. Such work on the surface seems very similar to other landmark-based navigation strategies using databases of vision, laser and/or sonar measurements [Thrun et al. 2001]. In practice, there have been some difficulties in applying the same technique to the auditory domain. Most

similar to the idea of landmark-based localization is work by Jwu-Sheng Hu with a Sony AIBO robot [Hu et al. 2006]. Rather than using existing environmental sources, however, the AIBO emits the sound being tracked, a barking sound, and then records the results. Combined with a database of pre-recorded barks from different parts of the environment, along with a history of the robot's movement, they can estimate position and orientation of the legged platform. Also utilizing robot emitted sounds, work by Dellaert et. al. [Dellaert et al. 2003] used distances to other robot noise sources to create maps of where robots had traveled through the environment. Further work by Sebastian Thrun [Thrun 2005] extended a similar methodology to dynamically localizing microphones in an array using a series of easily recognizable impulse noises (finger snaps, clapping, etc.) in the environment.

While all of these probabilistic or landmark-based approaches could potentially be applied to the general mapping problem around passive environmental sources, all of them suffer from a number of problems when applied to a real and naturally occurring auditory scene. The transitory nature of sound sources is one problem, making comparisons between old and new data difficult. Another problem is environmental echoes, which produce large defects in even small maps. To overcome this problem, the robot needs a representation that still has some meaning in the environment despite some changes to the auditory scene. This representational issue is exactly that the problem on which this thesis is focusing. As such, we will discuss this matter in further depth in the next chapter.

## 2.2 NATURAL LANGUAGE INTERFACES

Natural Language Interfaces refer to speech interfaces between computers/robots and humans. This could entail the robot speaking to a person, or the person speaking to the robot, or both. People are so used to communicating with each other by speaking that it is only natural that we would want the same interface for communicating with our robots. The argument is still open, however, as to how much of a speech interface is actually necessary. Most would agree though that many areas of robotics that require human-robot interaction would benefit from a real-time speech interface.

The domain of natural language interfaces can be roughly divided into two parts: listening, and speaking. Both parts are necessary for a full interface, but current research is still a long way from a completely integrated solution. As will be discussed in both sections, problems that repeatedly trouble natural language interfaces are the quality of the detected or generated speech, and the effects of masking noise on intelligibility. These are often environmental effects that could be minimized by an acoustically-aware robot utilizing knowledge of sound flow. Chapter 5 will discuss such an application.

### 2.2.1 THE ROBOT LISTENS

With the increased use of Hidden Markov Models (HMM), speech recognition rates have improved dramatically over the last decade. Just within the last several years, software has become readily, even freely, available [Lamere et al. 2003] for anybody to add real time speech recognition to their computer application. More and more roboticists are in fact, starting to do just that. The general approach that works best divides all human speech into 40-50 similar sounds, called phonemes. Each word in the

English language can be represented as an ordered set of these phonemes. Using measured transition probabilities between phonemes as input to an HMM, we can identify which phonemes the speaker pronounced, and try to reconstruct the spoken words from those. Russell and Norvig [Russell and Norvig 1995] estimate 80-98% accuracy in the best speech recognition systems, depending on the length of the input, the size of the vocabulary to be recognized, the variety of speakers, and the signal quality.

On robots, however, the word recognition rate tends to be lower than that achieved using just a microphone. Just as with sound localization, the internal noises from the robot and the variety of environments they can be located in can cause problems for robust audition. Sometimes, however, very simple interaction techniques can be used to compensate for the lower recognition rates. One such method, used on the robot HERMES [Bischoff 2000], was to ask the user for confirmation of the spoken command when the recognition results were poor. More complex solutions have combined the spoken command with visual cues, like pointing or other gestures that the human can easily perform [Perzanowski et al. 2000; Kettebekov 2002]. Both of these solutions work best with a limited command set.

Beyond recognizing a limited vocabulary, the next step involves actually understanding the semantic meaning of the speech, so that the computer/robot can respond appropriately. Just because one can recognize words does not imply natural language understanding. Russell and Norvig [Russell and Norvig 1995] state the following regarding natural language understanding :

*“We are given a set of ambiguous inputs, and from them we have to work backwards to decide what state of the world could have created the inputs”.*<sup>3</sup>

This is a very hard problem, and one that we are not close to solving in the general case. However, one technique that has been employed effectively on a number of robots [Roy et al. 2000; 2004]. Often called something like dialog-driven interaction, it implies a script that both the robot and the human are expected to follow. The person says one thing, the robot recognizes that it is the next line in the script, and then “says” its part. Using grammatical formalisms, some variability in the allowable words can be introduced to the script, but the programmer has to anticipate what sentences a human user might respond with, and the human has to stick to the script without changing topics.

Outside of natural language understanding, there is a second problem involving computer audio that is on the DSP side of speech recognition. Stream segregation is the ability to separate two or more different speakers from each other, and from background noises, so as to identify which person is saying what. When multiple people are speaking, the robot/computer has to decide which person to respond to, and remove all of the excess information. Due to the nature of sound propagation, all of that speech information from everyone in the room is normally combined in the stream picked up by each receiver. Unless somehow processed to separate the speakers, that stream is unintelligible as far as computer speech recognition is concerned. This is classically defined as the cocktail party effect [Bregman 1990]. Although the full solution is still

---

<sup>3</sup> Russell and Norvig, p 654

beyond our abilities, there do exist partial solutions. When the number of talkers is relatively few, independent component analysis (ICA) has demonstrated reasonable performance at separating multiple speech streams and overcoming their masking effects using a binaural microphone array [Takeda et al. 2006]. Traditional ICA, however, is easily confused by reverberant environments and can have difficulties when the people speaking are located too close together. A variant of ICA that works with only a single microphone to separate speech streams [Smaragdis 2001] has partially overcome these difficulties, by focusing on only the speech related cues. This method still suffers in the presence of reverberation, however, as it now includes reverberant effects in the resulting speech stream, making speech recognition difficult. In the case of either method, they remain computationally expensive, and are still limited by the length of audio segmented, the number of speakers or sound sources in the area, and some environmental effects.

An alternative solution to ICA, is to incorporate the geometry of the speaker's location into the algorithm. In the simplest case, where only one speech streams needs to be recognized at a time, a computationally simpler approach mentioned earlier [Nakadai 2003], is for the robot to rotate to face one speaker, and use directional microphones to amplify only that person's speech. Another partial solution based on geometric locations does the same without involving any motor actions by using a microphone array [Claudio and Parisi 2001; Argentieri et al. 2006]. If where the people speaking are located can be identified, then one can mathematically construct and amplify a narrow beam from the streams of multiple microphones. Recent work by Valin successfully applied this approach on a mobile robot, for separating up to three simultaneous speech streams [Valin 2005]. Unfortunately, these geometric-based approaches still have difficulties

when the two people speaking are located too closely together, or there is significant reverberation from nearby surfaces. In that case, the speech streams remain too intertwined for a location-based solution to separate, and may require repositioning the microphone array intelligently with respect to the auditory scene to change the relative angles to the speakers, or somehow limit the effects of reverberation.

### **2.2.2 THE ROBOT SPEAKS**

If no speech recognition is required, then robot speech can be a surprisingly easy concept to implement. The simplest systems use pre-recorded sound bites that are activated by a robotic behavior. A human can provide the recordings, and the robot only needs to interject them at the correct point during its human-robot interaction. How the robot chooses the correct sound bite could be in response to a single sensory stimuli like a clap [Jones et al. 1999], or it can even be learned by the robot. Minerva [Schulte et al. 1999], a robotic tour guide, adapted to the environment by monitoring how its voice commands affected the people densities around it.

An alternative approach to pre-recorded audio is text-to-speech (TTS). First, the text is converted to a phonetic code. Next, pitch, intonation, pausing, and rate are added. Finally, the audio is created from the parameterized phonetic code either by grabbing phonemes from a database, or synthesizing speech from basic phonetic units [Venkatagiri 2003]. The resulting speech is clearly understandable, and the newer software available continues to improve in quality. TTS was used on GRACE [Simmons et al. 2003] for the AAI robot challenge to deliver a talk by the robot at the conference. The greatest advantage of TTS is that the speech can be changed on the fly. Work in affective

robotics (i.e. emotions, moods, etc.) has demonstrated the advantages of dynamically adjusting the pitch, pausing, and rate of the synthesized speech (i.e., the robot's prosody) to give a robot personality [Breazeal 2001; Scheutz et al. 2006]. Work in more traditional human computer interfaces [Dusan and Flanagan 2002] changes not just the quality of the speech, but adapts the text input itself sent to the synthesizer by constructing grammars from human speech.

Still, TTS systems are not without their problems. One persistent issue is environmental noise. Although commercial systems are very good in quiet environments, TTS in noisy surroundings can still be hard to understand [Venkatagiri 2003]. The studies on this subject to date, however, have been performed using isolated phrases. With some context behind the phrases, people may be better able to understand the current systems. The problem of context though, is really a full interface problem. Both speech recognition and speech synthesis are necessary to provide the human participant with context. Unfortunately, the two problems are unequally advanced in achieving human-level performance. A study of natural language interfaces at Eurospeech '97 [Bloothoof and Os 1997], suggested that the existing speech recognition capability was holding interfaces back. Some users of the latest systems were at times completely fooled by the accurate TTS, forgetting that they were talking to a machine. But then when speech recognition failed, people fooled by the TTS system took longer than other participants in recovering from the error.

## **2.3 AUDIO CLASSIFICATION**

Passive sound classification is the problem of recognizing sounds and/or indexing sounds in the sound stream. Detection of air ducts, computer fans, machinery, a human voice, music, etc. are all examples of sounds that a robot may encounter in the sound stream and that it may be expected to recognize. To go even further, if we can recognize all of these factors, can we recognize the type of the environment itself? From the robotic perspective, this is a potentially huge field, but can be limited by the sheer processing power available onboard a mobile platform.

### **2.3.1 ENTITY CLASSIFICATION**

The oldest studied problem in audio classification is simply recognizing the start of the interesting signal. The voice/unvoiced/silence problem was the earliest classification problem developed for the telephone industry. The majority of the time in phone conversations is actually silence, or lack of audio content. So if the industry could separate silence from content, they would not have to transmit the entire signal across their lines, therefore increasing the number of calls they could handle simultaneously given a limited bandwidth. To compress the signal even further, they needed to also distinguish between voiced and unvoiced sounds. Because people are so good at recognizing speech, the telephone companies wanted to automatically recognize a minimum signal that could still be understood by the listener, so as to remove all of the excess auditory information from the stream resulting in an extremely compressed signal. The most well-known solution was developed by Atal and Rabiner [Atal and Rabiner 1976]. This approach detects vowel sounds in an audio stream. Later research showed

that provided the vowel sounds were present, human listeners could recognize with great accuracy the words in the conversation even when much of the remaining signal was removed. The resulting algorithm, called Linear Predictive Coding [Rabiner and Schafer 1978], compressed the audio stream and reconstructed the stream.

Much of the work in the Voiced/Unvoiced/Silence problem led to useful sets of features that could be extracted from an audio stream and applied to the more general classification problem, like mel-cepstrum coefficients [Quatiri 2002]. Ideally, we would like to classify any type of audio that is put before the computer, but it turned out that successful classification is highly dependent on the mathematical features extracted from the audio sample. Incorrectly chosen features will allow different audio samples in different classes to appear similar [Duda et al. 2001]. For example, water leaks can be detected in pipe networks by listening for certain frequencies (GMIC) [Hetek 2004]. However, the robot needs to know that it is near a pipe network and should be listening for these leaks. The same is true for identifying bird song [Kogan and Margoliash 1998], where the robot has to be told when to start listening for birds. For this reason, classification approaches currently have to be application specific, first choosing a set of classes, and then finding a good feature set and algorithm.

Although classification on robots in general, outside of speech recognition, has remained relatively unexplored thus far, research is already exploding in related mobile computing applications. Current classification applications using cell phones [Philips 2004] and wearable computers [Stager et al. 2003; Lukowicz et al. 2004] may in the long run be applicable to robotics and autonomous control.

### 2.3.2 AUDITORY SCENE ANALYSIS

One reason for the successes achieved in silence detection involved the telephones themselves. With a telephone, the microphone is placed right next to the person's mouth where the sounds are loudest. As a result, the microphone itself does not pick up much of the background noise. If forced to use a different microphone, however, which picked up more noise, then the classification problem becomes substantially harder. How can an arbitrary noise stream be separated from the "good" stream? What is needed is knowledge of the auditory scene, or contextual information regarding the signal currently being recorded. For instance, if the robot knows it is outside rather than inside, then it can use different algorithms to recognize the start and stop points, and use different filters on the incoming data.

This problem is called auditory scene analysis, and people do this all the time. They listen for different types of events given different auditory scenes in which they are located, and in general, people are relatively good at recognizing the different scenes. Given the task of recognizing the surrounding environment from an audio clip, out of context, people could identify the class of surroundings 70% of the time, and often in 20sec or less [Peltonen et al. 2002]. In comparison, a machine given a much smaller set of classes took almost 2 minutes to identify a much smaller, 5 vs. 25, set of classes. An even smaller set of classes, however, may also be useful in providing some auditory scene analysis. Work by Christophe Couvreur [Couvreur 1998] using HMM's, has attempted to classify environmental audio into a set of 5 categories using transportation-related sounds. It is too small a set of categories for solving the general segmentation problem, but such a set could be useful to a robot which is exposed, as part of its job,

primarily to transportation-related auditory scenes. In such a case, transportation sounds could still provide some useful auditory context for identifying the current scene and improving general classification results.

Dividing the world into classes for the machine to recognize is not the only solution to the auditory scene analysis problem. Work from Carnegie Mellon uses a wrist based light and audio sensor to detect places the wearer had been before [Maurer et al. 2006]. Alternatively, another method has been to recognize distinct sounds within the sound stream, and then derive auditory context, or the auditory scene, from those sounds. A good example is, if you hear a car horn, you are probably on the street [Clarkson et al. 1998]. Another, more hardware intensive solution, borrowed a wireless sensor network to assist the traveling microphone/processor. If the sensors know where they are, then they can broadcast this information, providing both context and localization information [Schiele and Antifakos 2002].

## **2.4 AUDIO PROBING**

The field of audio probing includes anything that requires an action by the robot to receive some form of audio feedback. Ultrasonic sensing is one example of audio probing. The robot emits a high-frequency sound click into the environment, and waits for the echo of the sound bouncing off objects in its path. It works on the same principle as bat echolocation, providing the robot with estimated distances to hard surfaces around it. Ultrasonic probing has been used for a variety of purposes including mapping, classification, and, of course, obstacle avoidance. For a more detailed description of how ultrasonic sensors have been used in mobile robot navigation see [Arkin 1998].

Ultrasound, however, is only a very simple form of audio probing. It takes the reflected sound pulse, and translates it into a single unit of distance. More complicated forms of audio probing audio probing can extract different information from the returned sound. For instance, by generating a loud impulse noise at some frequency, computers have already been able to estimate acoustic properties such as reverberation time in indoor environments [O'Keefe 1998]. Perhaps more compelling for robotics is the use of tools to generate noise. By using metal poles to hit surfaces, robots can try to classify the materials used [Krotkov 1995; Femmam et al. 2001], and/or identify other structural characteristics of the object being probed [Amsellem et al. 2006].

More complex forms of audio probing however, such as the classification of material surfaces, tend to have the same problem as classification algorithms. The advantage in audio probing is that there exists a little more context for completing the application. The object being classified is often in a known location, with suspected material properties. However, the breadth of the classification algorithm is still quite limited. In the general case, either more context is needed to narrow the range of possible results, or the classifier needs to be retrained for each application.

## **2.5 SOUND VOCALIZATION**

The final application area is that of sound vocalization. For statically located speakers, the vocalization problem is a highly specialized field. The goal is to provide the best acoustical experience for the listeners, whether they are in a concert hall, or sitting at home in front of the television. Toward that end, researchers have generated a wide number of acoustical parameters and simulation technologies that can assist the

creation of excellent acoustics in a variety of spaces. On robots, however, sound vocalization research has remained more limited. Typical robot audio today consists of prerecorded sound bytes and text to speech, played back for the user when and where the program says. Some robots even do this automatically. For example, ActivMedia's AmigoBot plays sounds on startup and shutdown, and Sony's AIBO can beep to indicate current conditions on the robot. Proper sound vocalization, however, should also involve sensing the environment. Humans, for example, will actually adapt their speech, and their singing or music to overcome environmental noise, or to tone down the speech when they are disturbing others. Good vocalization involves analyzing the environmental feedback to determine how the sound output should be modified for the greatest effect.

Sound vocalization research involving robots has largely been limited to two primary areas. The first area is natural language interfaces, which were previously discussed, and which have not traditionally involved adapting the volume, or tone of the generated sound. The second area is in music generation. Waseda University in Japan has been a hotbed for musical robots. Under the guidance of Makoto Kajitani, a series of WAM (Waseda Automated Manipulator) robots that could play pre-programmed routines on the piano were developed through the 70's. A full humanoid, the Wabot-2 [Waseda 2000], could play a keyboard while following a simple score using a camera for input. Several MUBOT's [Kajitani 1989], or musician robots, were also developed which could play recorder, violin, or cello. The latest work has been on a Flutist Robot that could not only play the flute, but also perform trills and vibrato on its instrument [Waseda 2000].

The original Waseda robots, however, did not actually adapt to the sounds they were playing. They figured out what note should be played when, and used planning to

reach the necessary keys in time. ISAC however, developed at Vanderbilt University [Alford 1999], actually adapted to the sound being played to improve performance on a Theremin. By listening to the output from the instrument, ISAC could achieve perfect pitch using a reactive control system and could play a sequence of notes in time with a human keyboardist. More recently in the same direction, a robot flutist has been added to the band that listens to its pitch and changes its blown air speed [Isoda et al. 2003].

## **2.6 SUMMARY OF APPLICATION DOMAINS**

What this collection of work should demonstrate is that the need for robot acoustics is already large, and only getting larger. Already, robots need to utilize sound for a large variety of applications, many for interacting with people, but also for improving quality in manufacturing and sensing, and even localizing the robot itself. In nearly all of these cases, however, the role of audio is limited by the complexity of the domain itself. As such, designers spend a great deal of effort building filters, and adaptive algorithms to remove the noise. For the remainder of this dissertation, we are going to focus on an alternative to filtering and noise removal. That alternative is acting with respect to the auditory scene. An acoustically-aware robot can utilize knowledge of sound flow to recognize where the noise is coming from and minimize its effects on performance.

## CHAPTER 3

### ACOUSTICAL AWARENESS

The concept of awareness has many definitions (see below<sup>4</sup>). Typically, though, it implies some knowledge of the surroundings at a conscious level. This could be specific knowledge about a particular object [Dourish and Bellotti 1992], or it could be raw data which generates an appropriate response [Kaelbling and Rosenschein 1991]. More specifically, however, being aware suggests an explicit recognition and understanding of the meaning behind what is sensed. Not only that, but once the recognition and understanding are in place, being aware means being capable of acting upon that sensed information. That is how being aware is different from simply recording the data, analyzing it, and filing it away for other truly aware beings to use.

American Heritage Dictionary <sup>4</sup> Having knowledge or cognizance. Vigilant; watchful.	L. Kaelbling and S. Rosenschein A tight coupling between sensing and action [Kaelbling and Rosenschein 1991]
P. Dourish and V. Bellotti “An understanding of the activities of others, which provides a context for your own activities.” [Dourish and Bellotti 1992]	M. Endsley “The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future” [Endsley 1988]

<sup>4</sup> The American Heritage® Dictionary of the English Language, Fourth Edition Copyright © 2000 by Houghton Mifflin Company. Published by Houghton Mifflin Company.

Awareness means acting with respect to the knowledge of the surroundings. Acoustical awareness, therefore, is coupling action with knowledge about the acoustic environment, specifically anything that indicates how sound flows or will flow in the physical world.

### **3.1 TYPES OF AWARENESS**

The general definition of awareness, as suggested by the previous definitions, may be too broad for our use. As awareness is defined, it can be applied to any application that perceives the environment, or has prior knowledge of it, and makes control decisions based on that information. But there are at least two different levels of being aware. First, there is simply reacting to a stimulus from the environment. Most of the robotics applications discussed in chapter 2 fall into this category. Second, there is understanding the context and the nature of the perceptual stimuli, and then making an intelligent, informed decision based on that knowledge. In robotics, this has classically been termed as reactive versus deliberative control, and both should have their place in a real acoustically-aware system.

#### **3.1.1 REACTIVE ACOUSTICAL AWARENESS**

Common acoustics applications in robotics today mostly fall under the category of reactive acoustical awareness. Simply stated, all reactive processes can be defined in terms of connections between 4 parts. At any time  $t$ , the signal ( $d_t$ ) from the acoustic perceptual hardware is transformed by some perceptual software processes, referred to here as perceptual schemas ( $P$ ). Then the perceptual schema results are fed into a motor schema, or behavior ( $B$ ), which transforms the results of the perceptual schema into commands for the motor controller ( $M_t$ ). This process, involving 4 parts can be seen in

Figure 3.1. Ideally, the transformation into action would be instantaneous, but in practice, the resulting motor command is generated as a result of the sensor data at some time removed from the original stimulus.

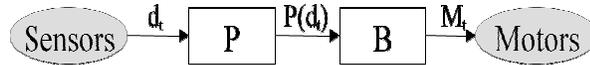


Figure 3.1. Basic reactive acoustically aware system.

Behaviors typically are relatively simple processes. An example behavior (phonotaxis) which uses acoustic perceptual data is moving in the direction of a sound source [Webb 1998]. By inserting one additional component into this architecture, the acoustically-aware behaviors can be easily combined with a multitude of other arbitrary behaviors. Described in [Arkin 1998], a behavioral coordination component (C), accepts the inputs of all behaviors at time  $t$ , and outputs the commands for the motor controller (Figure 3.2).

The general drawback to purely reactive systems, and this includes reactive acoustically-aware systems, is that there is no memory, so all of the knowledge for

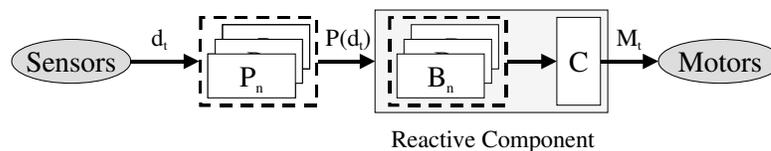


Figure 3.2 Reactive acoustically-aware system with behavioral coordination.

completing the task (both a priori and historical), which cannot be directly perceived by the robot, has to be built into the design of the controller component. This could be algorithmic manipulation of the perceptual data, or it could be facilitated by enhanced behavioral coordination mechanisms [Arkin 1998]. In either case, it places a large onus on the system designer, who must not only know the task extremely well, but also have to predict large numbers of failure situations and design control solutions that do not interfere with each other when operating simultaneously. It is still an open question as to whether or not general awareness can be achieved using sufficiently complex reactive systems, but in today's practice the required complexity for such a task is beyond the scope of a human designer.

### **3.1.2 KNOWLEDGE BASED ACOUSTICAL AWARENESS**

For more complex tasks, some form of internal state needs to be added to the system. Starting with the reactive aware system in Figure 3.3, this can be accomplished by adding a knowledge component to the architecture. The defining characteristics of a knowledge component are: 1) it takes as input data from the perceptual system and the memory of its previous state(s), and 2) outputs the result of some computational processing to the reactive component (behaviors or coordination mechanism). Without the output to the reactive component, the knowledge component is limited to logging and monitoring tasks only. The resulting combined controller forms a hybrid reactive-deliberative robotic controller (Figure 3.3). In this figure, the knowledge component has been further subdivided into two pieces: (1) a knowledge planner (K), responsible for maintaining world models and creating plans for active perception, and (2) a task

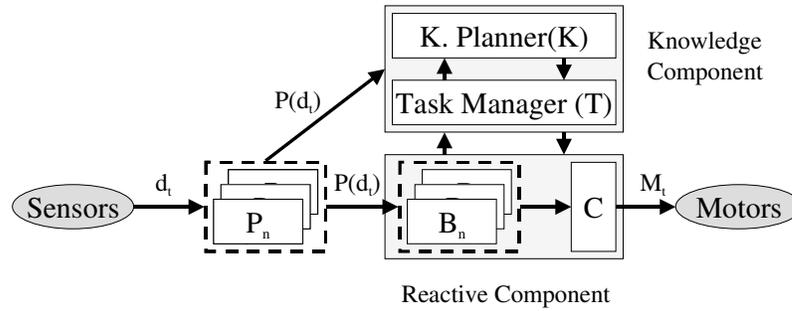


Figure 3.3 Hybrid architecture for supporting acoustical awareness.

manager (T), responsible for selecting, modifying, and following a plan. This subdivision of the deliberative component is equivalent to subdivisions made in earlier hybrid reactive-deliberative robotic architectures such as AuRA [Arkin 1998] or Atlantis[Gat 1991]. In these architectures, perceptual schema data is traditionally passed through the reactive component before reaching the knowledge component, but this is equivalent to the direct link between perception and the knowledge component, which is common in other knowledge acquisition architectures [Johansson 2003] so as to emphasize the importance of data collection.

The drawbacks to a knowledge component within the controller are synchronization, and general processing speed difficulties. As there is no limit on the amount of data stored in the knowledge components memory, there is no guarantee that any processing on that data can be completed fast enough to facilitate real time control. The solution, as proposed by some in the robotic architecture community [Gat 1991], is to run the knowledge acquisition components and motor control components asynchronously. At any given time-step, the motor controller has available to it the perceptual data processed at that time-step, and the most recent result of the knowledge

acquisition system. With this design, the knowledge component facilitates the performance of the reactive system, assisting where possible.

In this dissertation, we are focusing on the development of this second type of acoustical awareness. As discussed in chapter 2, there has already been extensive work in developing behaviors that connect acoustical sensors with action, and there is likely to be much more in the future as acoustical applications become more commonplace. What has not received much attention is the augmentation of these behavioral systems with the deliberative form of acoustical awareness. Where behavioral systems can fail because of the difficult nature of acoustic inputs in arbitrary environments, supplemental knowledge of how sound flows through the environment can be used to suggest actions for avoiding failure cases and overcoming local minima inherent to the acoustic domain.

In the remainder of this chapter, we will concentrate primarily on identifying the type of information needed to guide an acoustically-aware robot through this type of deliberative robotic architecture. This corresponds to the first of the three critical sub-questions described at the beginning of this thesis: What types of data and information about the auditory scene are useful for an acoustically-aware robot? After identifying the set of potentially useful information, we will then describe how this information can be utilized in a mathematical framework for estimating sound flow through an environment.

## **3.2 KNOWLEDGE FOR ACOUSTICAL AWARENESS**

Acoustics is an entity-driven perceptual domain. The problem of sound flow through the environment can be thought of in terms of its primary entities and how they interact: (1) Where is the sound coming from? (sources) (2) How will it travel around the

environment? (paths) and ultimately, (3) How will it appear to the listener? (receiver) Improving robotic applications requiring sensing or transmitting sound requires at least a basic understanding of all of these parts.

*“Every building acoustics problem, whether the enhancement of desired sounds, or the control of undesired sounds (noise), can be considered in terms of a system of sound sources, paths, and receivers.”<sup>5</sup>*

For the remainder of this section, we will discuss the various aspects of each of these three primary acoustic entities. What are the types of information that an acoustically-aware robot may want to know? In the next section, we will discuss how this information can be combined together to model sound flow through an environment, and discuss why all of this information may or may not be needed for guiding an acoustically-aware robot.

### **3.2.1 SOUND SOURCE MODEL**

The sound source is the most obvious of the three acoustic entities. These are the objects that emit that sound which ultimately arrives at a microphone, or receiver. Furthermore, they are also the entities that are most likely to change over the time during which a robot is executing some task. As such, it is imperative that a robot be able to acquire as much information as possible about the sound sources that may be effecting

---

5

Cavanaugh, W. (1999) , p.3

the environment in which it is situated. For each source in the environment, this includes information about:

- *Position* – coordinates of the sound source's location and orientation in environment.
- *Directivity* – the variation in amplitude of the outgoing signal due to angle of departure. This variation in amplitude may vary with frequency, as some frequencies may be better absorbed or transmitted by the materials from which the sound source is constructed.
- *Sound Function* – the sound produced by the sound source. This includes frequency, volume, and changes in these properties over time.

Of these three types of information, position is the only one that must always be known to some degree of accuracy. If the source position is not known, then the robot cannot predict anything about its effects on the environment. In the absence of further information, simplified models can be substituted for the other two properties. Without directivity information, the source can be approximated as omni-directional, invariant to both frequency and angle. Without specific knowledge of the wave function, a robot can approximate the sound source as a constant volume pink noise source. A pink noise source assumes equal energy frequency bands, and produces sound similar to fan or wind noise.

Given enough time, all three of these pieces of information about a source are determinable to some extent by a mobile robot. Chapter 4 discusses in more detail how this is possible, and Chapters 5 and 6 apply this information to robotic applications.

### 3.2.2 ENVIRONMENTAL INFORMATION - PATHS

The environment ultimately controls how much of the sound emitted by the sound sources reach the receiver. If there are walls in between the source and the receiver, then the sound will have to either travel around them, or through them. Either method reduces the volume, or otherwise changes the sound arriving at the receiver. How much that sound is changed is dependent upon the following properties of the path model:

- *Geometrical layout*: includes obstacle positions, walls, and all other surfaces in the environment (Figure 3.4). Used for predicting reflections, and regions of acoustic shadow.
- *Material Properties*: what materials are each of the surfaces made of, and what are the acoustic properties of those materials? Used for determining how much sound is transmitted through a wall, absorbed by the wall, or reflected from it.
- *Structural Composition*: what does the support structure of the building look like, such as would be found on an architectural blueprint. Used to determine absorption rates when calculating the strength of transmitted sound.

If this information is not provided a priori to the application, then acquiring detailed information about the environment can be very difficult (although not impossible) for a mobile robot. Luckily, simplistic yet reasonable assumptions are available for all of these. The simplest geometrical layout is an unobstructed outdoor environment. For indoor applications, sound propagation models generated under this assumption are better when the walls are far away, and the environment is relatively uncluttered, but they may still have some value depending upon the application. For better performance, what is needed is a map of the primary architectural features of the

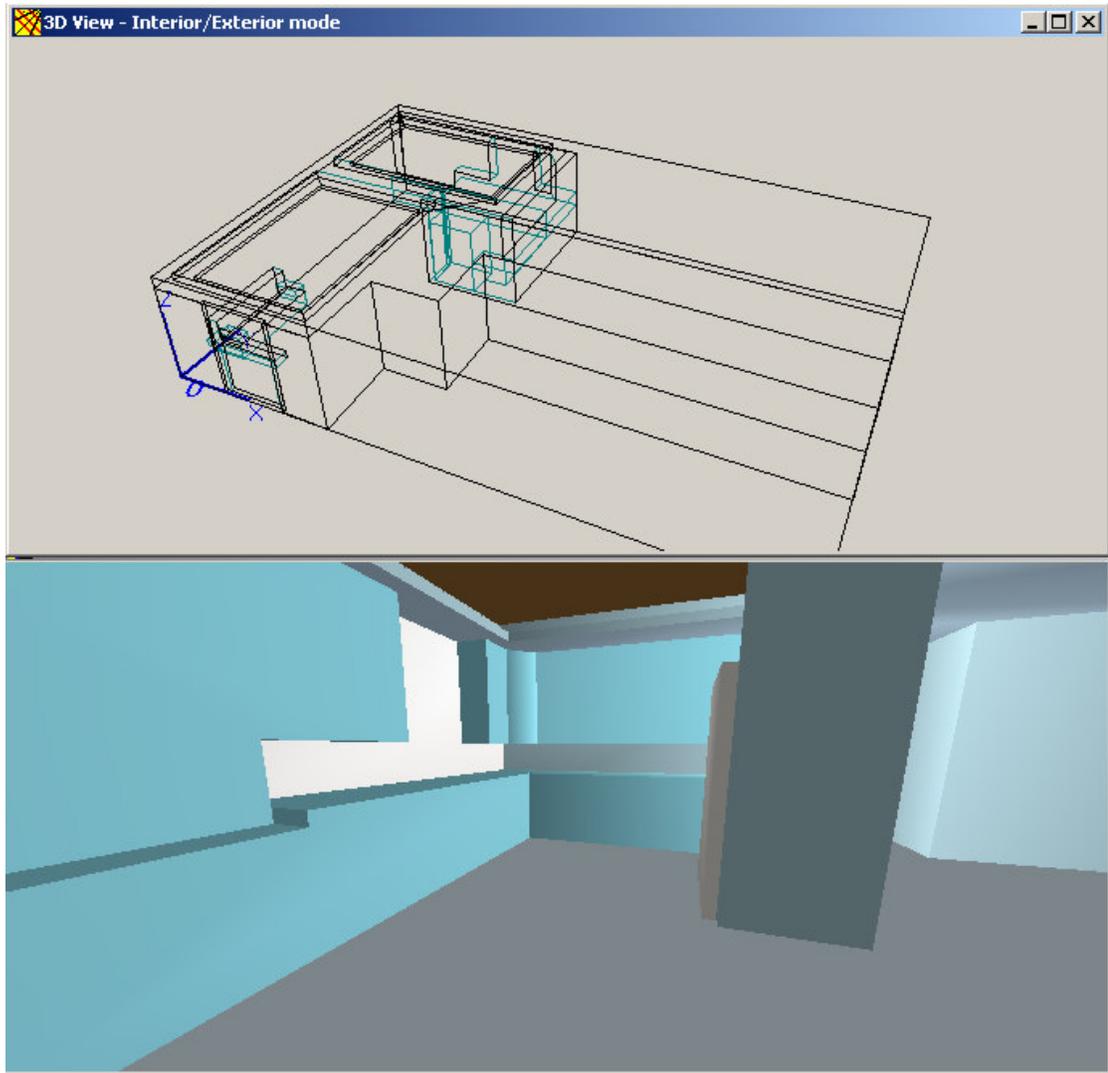


Figure 3.4. 3D model of the Aware Home Laboratory used for estimating reverberation effects and general sound propagation. Small obstacles in the environment, such as plants, dishes, etc. are less important for reverberation models than architectural specifications of surrounding floors, walls, and ceilings. The visualization was created by the software, Odeon 6.5 Combined.

environment, including large pieces of furniture and/or machines. The larger the object the more effect it will have on sound propagation, but also the less likely it will be moved around. For instance, a small houseplant, which could block the path of a robot moving through the environment, does not affect sound flow very much even if it is moved from place to place. Figure 3.4 demonstrates an example 3D geometric layout used with the Odeon acoustical modeling software for sound flow through a kitchen/living room. In Chapter 4, we will discuss the acquisition and use of the geometric layout for an acoustically aware robot. Chapter 6 will then discuss the application of a 2D model to real robotic scenarios.

Without knowledge of the geometrical layout, there is no need for material properties or structural composition of the environment. If the layout is available, however, but material and structural properties are not (a very common occurrence), then a simplifying assumption that can be used is that the walls are thick and solid (i.e. non-transmitting), and that all materials are perfectly reflective. In this dissertation, we will always be using these simplifying assumptions in our sound flow models, since material and structural composition are usually unavailable to the robot. As discussed in chapter 2, however, some material properties are potentially determinable by the robot, as may some structural composition. In future applications of acoustical awareness, additional knowledge, even if pertaining only to individual surfaces or walls, could still be used in conjunction with these simplifying assumptions to improve the accuracy of overall sound flow models.

### 3.2.3 RECEIVERS

The final acoustic entity needing some description is the receiver. Usually, with a robotic application, this will refer to a microphone, or array of microphones, situated on a mobile robot. Sometimes, however, a robot might be producing noise, or speech, for a human listener to hear, in which case the receiver would refer to the human listener's auditory system. In this chapter of the dissertation, we are interested in building a model of sound flow through the environment, so as to predict what a listener will hear at any location in the environment. In order to estimate what the listener might hear, however, the robot also needs a model of that receiver. While the range of information that can be stored and collected for each receiver will vary significantly between applications, the following are typical attributes for a receiver model that a robot might need:

- *Position* – coordinates of the receivers location in environment
- *Directivity*: the variation in the perceived amplitude of the incoming signal due to angle of incidence.
- *Frequency Response*: a range of frequencies the receiver can detect, and the relative amplitude across frequency bands.

As with sound sources, the position of the receiver is a critical piece of information. Without knowing anything about the position of the receiver, it is impossible to guess what the receiver might hear. Since the receiver is often mobile, however, the position of the receiver is often initially estimated as a set of possible positions, or an area over which the receiver might move about. This allows a broad initial estimate in the form of a map, or guide, that a robot can then use to predict what it will hear at any given reachable location.

The second two attributes of the receiver, directivity and frequency response, are less important to the sound flow estimation process. Without further information, the receiver can be simply modeled as an ideal point sampler. That is, any pressure changes occurring at that location are assumed to be perfectly recorded by the microphone, regardless of the frequency or the angle of incidence to the receiver. In this dissertation, our receiver information is always restricted to position only, as we used omni-directional microphones in all of the experiments, and did not separate out frequencies.

### **3.3 MATHEMATICAL FRAMEWORK FOR SOUND PROPAGATION**

The previous section identified a large amount of knowledge that can be collected from each of the three primary acoustic entities in the soundscape: sources, paths, and receivers. That knowledge, however, does not by itself indicate what a robot would experience as it moves about the environment. Ideally, an acoustically-aware robot should be able to predict what it will hear so that it can make decisions about either avoiding the noise, or moving towards the noise, so as to improve its performance at some acoustic task. How do we bridge the gap between the information available to the robot, and this predictive capability? Thankfully, this problem is already of great interest to another research community, the field of architectural acoustics. When designing a building, engineers often need to consider the ramifications of their design choice on the flow of sound. In the case of concert halls, the aim is to aid sound propagation, so that the sound reaches more people, and does so in a fashion befitting the type of music being played. In more typical mundane buildings, such as offices or homes, the goal is usually the opposite, trying to minimize the effects of ambient noise on people trying to work or

live. Whatever the target building of the acoustical engineer, the goal requires the same knowledge, knowledge about the flow of sound from source to receiver, through the environment. This is exactly what our acoustically-aware robot needs.

Of the methods for modeling, or understanding, sound flow through an environment, the theory of sound fields is one of the most commonly utilized. The theory itself is based on the physical principles of sound propagation, as laid out in common acoustic textbooks [Wilson 1994; Cavanaugh 1999; Raichel 2000]. Although this is not the only method for predicting sound flow through an environment, it is particularly suited to mobile robotics as it provides a framework into which many different types of information can be inserted. Furthermore, the resulting framework allows for unattainable knowledge, breaking down gracefully in the presence of unknown quantities. If the robot does not have available to it some knowledge, either a priori or through self-acquisition, then the resulting estimates of sound flow can still guide a robot to or away from sound sources, improving performance over an uninformed robot (see Chapter 5). Other solutions are not quite as appropriate for robotics. In particular, it is possible to solve the wave equation directly using certain assumptions and approximations for unknown quantities. However, solving the equation typically trades speed for accuracy, and degrades quickly with missing information. As computation is still at a premium for a mobile platform, and missing information is common, this solution may not be as applicable in robotics where an estimation of sound flow across large areas may need to be known.

### 3.3.1 THEORY OF SOUND FIELDS

The theory of sound fields [Svensson 2002] is used to make predictions about the soundscape given the knowledge available. Although called a theory, it is more of methodology that separates different types of acoustic knowledge from each other, so as to make useful estimates about the sound present at any given location in the environment even with only partial information.

To describe the theory, let us first make the following assumptions:

1. Assume there exists a function  $S_n(x,y,z,t)$ , for each sound source ( $n$ ) in the environment, which can determine the instantaneous pressure generated by that sound source at time ( $t$ ) and location ( $x,y,z$ ) in the environment.
2. Assume that all functions  $S_n$  in the environment are independent of each other. In the case of sounds in the audible range at typical volumes, the flow of sound due to each source is generally unaffected by other sources.

Using these assumptions, the theory of superpositioning says that the total pressure at a given time and location ( $d_{x,y,z,t}$ ) can be estimated as the ambient pressure ( $P_0$ ) plus the sum of the effects of each sound source in the environment.

$$d_{x,y,z,t} = P_0 + \sum_{i=0}^n S_i(x, y, z, t) \quad \text{Equation 3.1}$$

But what is the nature of the sound source function  $S$ ? Although the effects of each sound source in the environment can be separated from each other, the function  $S$  cannot be separated from the physical environment itself. If we have a spherical sound source, which generates a single pressure pulse, then that pulse will propagate

spherically, decaying in amplitude, until it hits a physical obstacle. At that point, on a smooth surface in a perfectly rigid environment, the entire pulse is reflected from the wall back into the environment at some angle relative to the angle of the incoming pressure wave. If the surface is not smooth however, then some scattering will occur at the impact point, generating uneven reflection from the wall. Moreover, if the surface is not perfectly rigid, then the wall will absorb some of the sound, and some will be transmitted through to the other side.

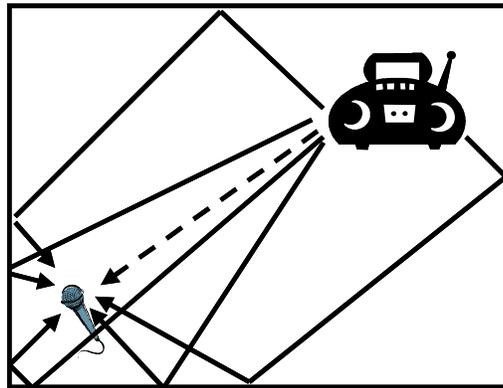


Figure 3.5 Direct (dashed) vs. indirect paths from source to receiver.

Fortunately, however, the nature of sound again promotes the use of superpositioning. Where before the soundscape was split into component sound functions, now the component sound functions may be split into their component parts.

In this case, the separable parts of interest are:

- *Direct Sound* – sound waves with an unobstructed path from source to receiver.
- *Diffraction* – the bending of sound around barriers/obstacles.

- *Reflection* – throwing back sound waves from a surface. Includes absorption and scattering effects due to rough, non-rigid surfaces.
- *Structure-Borne Vibration* – sound absorbed by walls/obstacles may generate vibrations that can travel a long way through a solid medium before re-entering the air as sound. A good example is a vibration that travels from the basement to the third floor of a high rise, using the steel skeleton of the building as a conduit.
- *Transmission* – sound waves that continue through the wall/obstacle to emerge as waves on the opposite side.

It is more common in the acoustics literature [Raichel 2000], however, to group a number of these effects together into separable sound fields. A *sound field* can be described loosely as the sound in the region of interest around the source. The *direct field* is then the sound field created only by direct sound. A *reverberant field* includes diffraction, reflection, and surface diffusion effects. Since these fields are assumed to be independent of each other, the sound function ( $S_n$ ) can then be described as the summation of the effects of each field on the location  $(x,y,z)$  plus some transmission effects at given time  $t$ .

$$S_n = D + F + T \quad \text{Equation 3.2}$$

Where:

$D$  = direct field.

$R$  = reverberant field, including reverberations, surface diffusion, and diffraction effects.

$T$  = transmitted and structure-borne sound

Although the exact form of these sound fields is impossible to determine at any given point in time, some reasonable estimates may be generated for a number of these fields using existing tools. As will be discussed in the following sections, the selection of tools, how they are applied, and even what information they need actually varies significantly from field to field. However the estimates for each field are generated, the beauty of the sound fields model is that as long as the effects being modeled are independent of each other (or have relatively small effect on each other), they can still be summed together to estimate the whole. Furthermore, if some effect or sound source is not currently being modeled, for whatever reason, then while the accuracy will decrease, the knowledge about the flow of sound in the environment is still correct, and still potentially useful for effecting robotic movement.

While constructing this model of separable sound fields, we have assumed that all of the fields are independent of one another, it should be noted that in reality the transmission and structure-borne effects are not actually independent of either the direct or reverberant fields. The origin of the sound to be transmitted has to come from somewhere, and may originate in either the direct or reverberant field. Then, any sound that is actually transmitted, i.e., reaching the other side of an obstruction, will continue to reflect around the environment and contribute to the reverberant field. In practice, however, transmitted sound is not included in either field because the calculation methods for estimating either transmitted or structure-borne sound differ substantially from those for calculating the other fields.

In Figure 3.6, we demonstrate the overall relationship between the separable sound fields model and the previously described acoustic entities. The robot will be

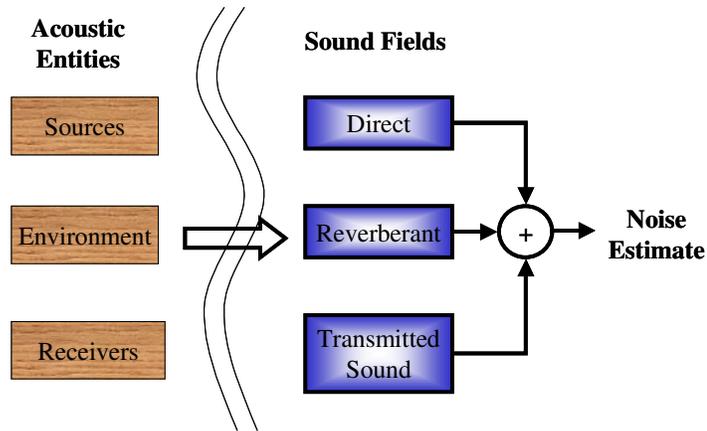


Figure 3.6 Relating acoustic entities to sound fields when building noise estimates. Each of the sound fields makes use of information from all three of the acoustic entities to estimate sound propagation.

gathering as much information as it is able to about each of the three acoustic entities. That information will then be used to construct each of the sound fields that build our final noise estimate.

### 3.3.2 DIRECT FIELD

To describe the direct field, let us start by describing the source itself. Using Fourier's theorem, any waveform can be described as the super-positioning of some number (N) of harmonic waves at different amplitudes, frequencies, and phases.

$$\vec{p}(t) = \vec{A} \cdot \sin(\vec{n}\omega t + \vec{\varphi}) = \vec{C} \cdot e^{j\vec{n}\omega t} \quad \text{Equation 3.3}$$

Where:

$\vec{A}$  = amplitude vector of length N

$\omega$  = fundamental frequency

$\vec{n} = [1, 2, 3, 4, \dots, N]$

$\vec{\varphi}$  = phase angle vector

$\vec{C}$  = complex amplitude vector.

Determining the effects of the direct field on the receiver is the simplest sound field to estimate. Assuming an ideal spherical source at a distance ( $l$ ) from an ideal point receiver, and a constant speed of sound ( $c$ ), then the amplitude of the incident wave can be approximated by a linear drop off with the distance ( $l$ ). If no unobstructed path exists between source and receiver, then  $l$  approaches infinity.

$$\bar{D}_n(l, t) = \frac{1}{l} \bar{p} \left( t - \frac{l}{c} \right) \quad \text{Equation 3.4}$$

In practice, however, receivers are never ideal. A real microphone is limited in the detectable frequency range. Microphones detect minute changes in pressure (sound waves) only when those changes cause mechanical elements inside the microphone to move, which in turn generates an electric signal. However, the shape and the material properties determine the frequency range to which the microphone responds, and there is no single element design that resonates in response to all frequencies. Even those frequencies that do cause resonance in the sensing element, vary in the size of the resulting amplitude of vibration, in turn affecting the value of the “sensed” signal. This is called the microphones frequency response ( $\bar{r}$ ), and is usually provided by the manufacturer.

The other adjustment that needs to be made for real microphones, is the variation in directivity. An ideal microphone is omni-directional, reacting equally to signals arriving from all directions. A real microphone is not. Even the microphones sold as omni-directional usually have blind spots where the physical cable connects to the

microphone. In practice, this can be adjusted for using a scalar adjustment ( $Q_{\theta,\phi}$ ), depending on the incident angle ( $\theta,\phi$ ). In reality, the directivity value at a given angle should probably be a vector that varies with frequency, but the amount of error introduced by the scalar assumption is relatively low in comparison to other assumption-induced error.

To define the resulting direct field effect, let us first define an element-wise vector product operation (\*), where, for all elements in vector  $\vec{B} = \{B_1, B_2, B_3, \dots, B_N\}$ :

$$\vec{B} = \vec{C} * \vec{D} \quad \text{implies} \quad B_i = C_i \cdot D_i$$

Then, then the direct field effects from an omni-directional point source, on a directional microphone can be estimated as the element-wise vector product of the magnitude of the individual frequency components, and the directivity pattern ( $Q_r$ ) of the microphone at that angle towards the source ( $\theta,\phi$ ).

$$\bar{D}_n(t) = \frac{Q_r(\theta,\phi)}{l} \left( \bar{r} * \bar{p} \left( t - \frac{l}{c} \right) \right) \quad \text{Equation 3.5}$$

Where:

$l$  – path length from source to receiver

$Q_r(\theta,\phi)$  – directivity adjustment, for incident angle ( $\theta,\phi$ )

$\bar{r}$  – frequency response of the receiver

$\bar{p}(t)$  – sound function generated by the source at time  $t$

$c$  – speed of sound

Receivers are not the only directional entity in the equation. Sources, too, are directional, and are typically modeled by a similar directivity function ( $Q_s(\theta,\phi)$ ). In the

source case however,  $(\theta, \phi)$ , are measured from the front of the source to the receiver instead of vice versa. So if we differentiate between the angle of departure from the source  $(\theta_s, \phi_s)$ , and the angle of incidence on the receiver,  $(\theta_r, \phi_r)$ , Equation 3.5 can be re-written to include both source and receiver directivity:

$$\bar{D}_n(t) = \frac{Q_r(\theta_r, \phi_r) \cdot Q_s(\theta_s, \phi_s)}{l} \left( \vec{r} * \vec{p} \left( t - \frac{l}{c} \right) \right) \quad \text{Equation 3.6}$$

So, to summarize the estimation of the direct field, there are a number of different pieces of information that can be used. A source location is necessary to calculate the path length to each location in the field, and a source function is necessary to scale the response at each possible receiver location. Even the source function, however, does not

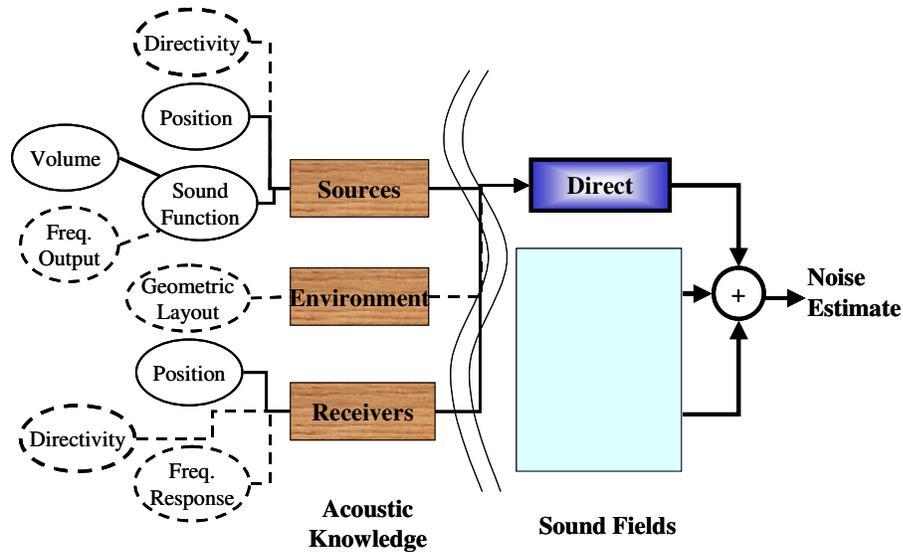


Figure 3.7 The information from each acoustic entity necessary for building a direct field estimate. Only the position and volume of the sound source, and the position of the receiver (solid circles) are absolutely necessary. Other information that can also assist in the calculation (dashed circles) includes directivity of the source and receiver, frequency output of the source, the frequency response of the receiver, and the geometric layout.

need to be complete, as a simple volume is good enough for making an estimate. Everything else is optional. Equation 3-4, which uses just path length and source function is probably the simplest estimate of the direct field to use with new sources. Then, as more information becomes known about the microphone and the directionality of the source, these can also be incorporated into the equation to hopefully improve the results. Figure 3.7 graphically displays the set of information that needs to, or just can, be included in the direct field calculations. Unlike the other fields, recalculating the direct field is a real-time operation even at its most complex that can be done quickly whenever new information becomes available. In Chapter 4, we will discuss the creation of direct field estimates from source position, directivity, and volume using this approach. It is left for future work the incorporation of time varying sound functions and microphone information into the field calculations.

### **3.3.3 REVERBERANT FIELD**

A reverberant sound field can be described as a field created from the reflection, diffusion, and diffraction of sound waves in an environment with physical obstructions to the flow of sound. Without the physical obstructions, it would be only the direct field, but with them, sound waves can take many alternate paths to reach the same location in the environment. When a sound wave hits a hard, smooth surface, it will bounce off at an angle of reflectance ( $\theta_r$ ) equivalent to the angle of incidence ( $\theta_i$ ), see Figure 3.8. If the surface is perfectly rigid, then no energy is lost at the transmission point, and only the phase of the wave is changed. Otherwise, the amplitude of the wave is affected relative

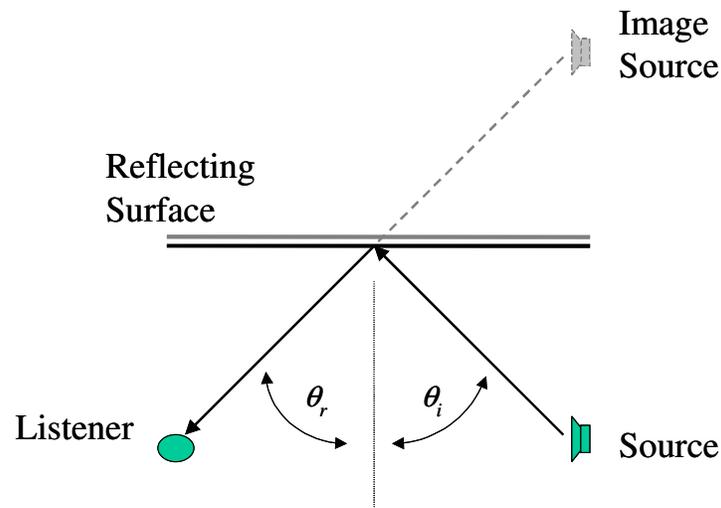


Figure 3.8 Modeling the image source method. In a normal reflection, an incident wave is reflected off of smooth surface at incident angle  $\theta_i = \theta_r$ . The image source method uses this principle to replace reflections with equivalent sources having the same angle of incidence as the reflection

to the frequency of the incident wave. How many of these reflected waves ultimately reach the receiver is determined by the shape of the environment.

### Image Source Method

The simplest model of reverberant environments in fact uses this relation of equal angles to describe the reverberations as the effects of “mirror-image” sources. Figure 3.8 describes the idea. Each reflecting surface can be replaced by an equivalent source ( $S'$ ), which is located at a mirror-image position about the reflecting surface from the original source. The summation of the direct field effects from all of these sources then constitutes an estimate of the reverberant field. This is called the image source method (ISM) [Savioja 1999].

In Equation 3.7 the image source method is used to estimate the combined effects of all 1<sup>st</sup> order reflections from a single source  $n$  to a receiver located in the environment. The order in this case, refers to the number of reflections that have occurred in the path from source to receiver, so 1<sup>st</sup> order means only one reflection between the source and the receiver.

$$\bar{R}_n(t) = \sum_{i=1}^{\# \text{sources}} \frac{Q_r(\theta_{r,i}, \phi_{r,i}) \cdot Q_{s,n}(\theta_{s,i}, \phi_{s,i})}{l_i} \left( \bar{E}_i * \bar{r} * \bar{p}_n \left( t - \frac{l_i}{c} \right) \right) \quad \text{Equation 3.7}$$

Where:

$l_i$  = path length from the real source  $n$  to the receiver.

$Q_r(\theta_r, \phi_r)$  = directivity adjustment at microphone, for incident angle  $(\theta_{r,i}, \phi_{r,i})$

$Q_{s,n}(\theta_s, \phi_s)$  = directivity adjustment due to source  $n$ , for departure angle  $(\theta_{s,i}, \phi_{s,i})$

from the image source.

$\bar{r}$  = frequency response of the receiver

$\bar{p}_n(t)$  = sound function generated by the source at time  $t$

$c$  = speed of sound

$\bar{E}_i$  = Environmental adjustment to the amplitude, depending upon the materials reflected off along the path.

The image source method however, is not without problems, largely because of its simplicity. For one, each reflecting surface needs to be flat in order to estimate the reverberant effects as that of a mirror-image source. Second, the surface also needs to be smooth, otherwise diffusion (or scattering) effects occur as a multitude of waves are generated from each intersection point, which image source methods cannot accurately

model. Thirdly, image-source methods do not scale well for calculating higher-order reflections. Even with just 1<sup>st</sup> order reflections, the positions for all mirror image sources have to be calculated first, before directivity and path lengths can be determined. As the order of the reflections increases, however, the number of image sources grows exponentially. Furthermore, the computational effort required for estimating the position, path length, and directivity of each higher-order source also grows along with the order, so even though the final summation of results is relatively easy, the finding of the image-sources themselves is highly inefficient.

Another problem not already mentioned is that the image-source method is very inefficient in estimating the entire field. For each possible position of the receiver an entirely new set of image-sources has to be calculated. Now, if the original source position is known at compile time, the positions of all image-sources for all receiver locations can actually be calculated in advance (which also alleviates the computational inefficiency problem), and then stored for quick access in a database, making the image-source method feasible in real-time. However, if the original source position is not known, then calculating all of the image-sources for all positions in the environment when it does become known will most likely require further compromises in accuracy to speed up the calculation time. Such compromises may include reducing the order of reflections calculated and estimating the field in a more limited area, perhaps for making spot updates to a pre-existing reverberant field model. For this reason, as well as the earlier problems, the image-source method was not implemented as part of this dissertation to describe the reverberant field. Instead, it was only included because its simplicity may

prove useful to other robot designers. A greater description of the image source method and its effectiveness can be found in [Allen and Berkely 1979; Svensson 2002].

### Ray-Tracing

The image source method is loosely based on the concept of wave propagation. Instead of handling individual wave reflections though, it models the reflections as sources. Although it has been revealed to be computationally very expensive it was originally proposed as a faster alternative to a ray-tracing model of wave propagation[A. Krokstad 1968], which was capable of modeling surface scattering. Modern processing speeds however have made ray-tracing feasible and it has been implemented in a number of commercial platforms including Catt-Acoustic, Ease/Ears, and Odeon.

The basic idea behind ray-tracing is that a number of rays (often in the form of cones) are generated at random angles from the source into the room. When a ray hits a surface, it is reflected either specularly (as in image-source methods) or diffusely, depending on the scattering coefficient of the intersecting surface. Greater accuracy could be received by generating a new batch of rays at each surface to reflect on the level of scattering, but doing so quickly escalates computationally as the number of reflections is increased. To then determine the estimated effect on a single receiver then is only a matter of determining which rays intersect with the position of the receiver and summing them together (use Equation 3.6, substituting ray characteristics for sources).

What ray-tracing does not do well are two things. First, if the amount of diffusion at each surface depends on frequency, then without generating new rays at each intersection, the simulation may have to be repeated for each octave band. Second, ray-

tracing does not simulate edge diffraction effects. Sound waves will bend around corners, or partial barriers, but the ray-tracing model does not support this phenomenon. However, there are some methods for estimating diffraction around partial barriers [Pierce 1989] that may potentially be used to estimate the effects of diffraction independently of ray-tracing.

Although the later augmentations to overcome diffusion and edge-diffraction effects have not yet been implemented for this dissertation work, ray-tracing is used for modeling the reverberant field in Chapters 4 and 6. Chapter 4 in particular describes the data collection process by our mobile robot, and the creation of a reverberant field model from that data. A pseudocode description of our implementation is also available in Appendix B.6.

#### Other Methods for Estimating Reverberation

Besides the image-source method and ray-tracing, there are also a number of other methods for calculating the reverberant field. Although these approaches were not implemented during dissertation work, these alternatives each have their own advantages/disadvantages, and are included here for completeness:

- Beam-tracing [Funkhauser et al. 2004]

also uses rays, but creates beams out of adjacent rays. The advantage is that edges can “split” beams to model diffraction effects. Beam-tracing is similar in computational complexity to ray-tracing, but does not easily model surface scattering.

- Radiosity [Korany 2000]

Another computer graphics inspired method which predetermines the wall reflection values as parts of larger wall elements. It is computationally efficient and easily handles a moving receiver. However, while scattering is simple to predict, specular reflections are more difficult to incorporate. In addition, it does not handle edge diffraction.

- Solving the wave equation

There are also several numerical solutions to the wave equation for estimating the reverberant field. Volume element methods, such as the finite difference method in the time domain (FDTD) [Botteldooren 1995], discretize the air volume, and calculate the sound propagation as a function of neighboring units. Surface element methods, such as the boundary element method (BEM), discretize all boundaries and estimate their contributions to the sound pressure and particle velocity [Savioja 1999]. Numerical solutions provide very good details about diffraction, diffusion, reflectance, and surface scattering, but are extremely heavy computationally, so are generally practical only in either small or very simple environments.

The general purpose of being able to estimate the reverberant sound field is to integrate some information that was not previously available into the robotic controller. Unlike creating virtual spaces [Svensson 2002], or building concert halls [O'Keefe 1998], where accuracy is critical, there is already so much noise inherent to mobile robotics that obtaining the highest level of accuracy in acoustic simulations of the reverberant field is

unnecessary. What the availability of all these tools for estimating the reverberant field means is that the level of accuracy in estimating the field can be tailored to the acoustically-aware application. If diffraction effects are particularly strong in that target environment, then an alternative method to ray-tracing may be used. Hybrid combinations of two or more methods [Svensson 2002] are also easily fitted to this computational approach.

In general, the differences between each of the approaches for modeling reverberant effects are the complexity of the computational model and the specific effects included in the reverberant model. All of the approaches, however, from the image-source method to solving the wave equation, make use of the same information. They need the same information that the direct field required, plus knowledge of the geometric layout. Furthermore, these methods can now incorporate material properties of the surface if they are available. Figure 3.9 summarizes the set of information required for building a reverberant field estimate.

### 3.3.4 TRANSMISSION

The transmission of sound occurs when not all of the sound is reflected from an obstacle. In that case, some energy from the incident wave ( $E_I$ ) is reflected ( $E_R$ ), some is absorbed by the wall ( $E_A$ ), and some is transmitted through the wall ( $E_T$ ) and out the other side.

$$E_I - E_R - E_A = E_T \quad \text{Equation 3.8}$$

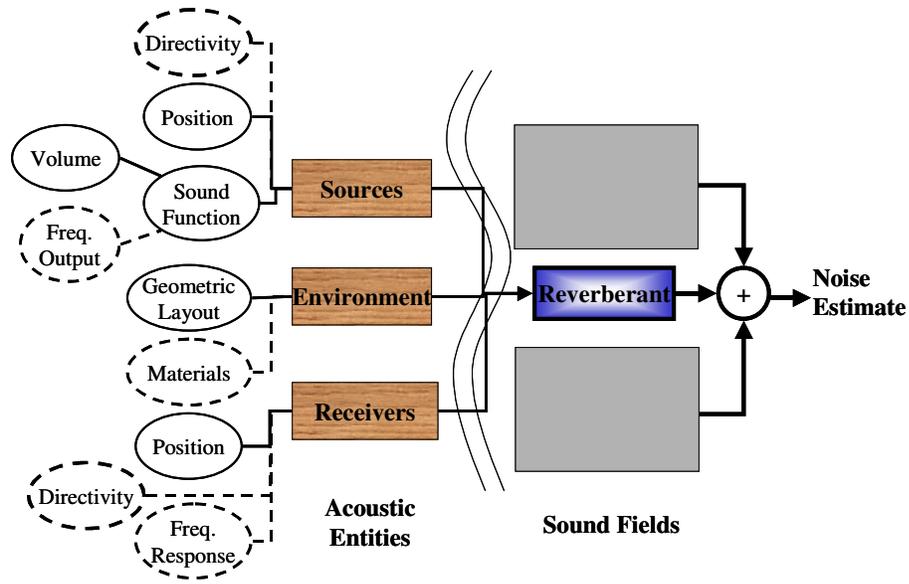


Figure 3.9. The information from each acoustic entity necessary for building a reverberant field estimate. Any non-trivial reverberant field requires a geometric layout in addition to the same information required by the direct field. Also, in the reverberant field equations, we see the use of material properties as optional

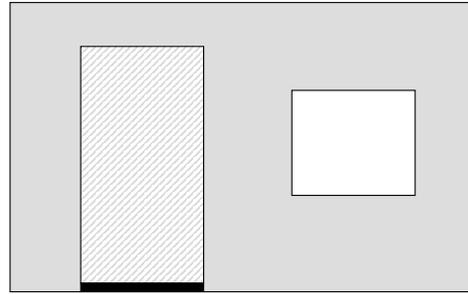
Unfortunately, even assuming a flat, relatively thin, panel in air with uniform density, the resulting transmitted energy is not easily estimated. It is not quite as simple as calculating either the direct or reverberant fields, because even if an accurate enough model of the environment (including geometrical layout and material properties) was available, then accurately estimating transmission would still depend on having accurate representations of the driving source functions. Due, in part, to the difficulty of acquiring accurate enough knowledge about the domain, our robots will not be estimating transmitted energy in the experiments described in this dissertation. However, if transmission plays an important role in the environment a robot is being deployed to, then there are still some techniques that can be used to make rough approximates of transmission for use by a mobile robot. These techniques, which will be described in the

following sections, are divided into two different sets, each estimating different types of transmitted energy. The first type is direct transmission, which calculates the transmission loss (i.e. the fraction of sound power transmitted to incident sound power) of a wave traveling directly through an obstacle. The second type is structure-borne transmission, which happens when energy absorbed drives vibrations that travel through the structure to reappear as sound in a new location.

### Direct Transmission

Direct transmission of sound is the transmission of sound through the obstacle, so that the angle of the waving leaving the obstacle is relative to the angle of incidence. Most typically, it is calculated with respect to walls in an environment, particularly outer walls that are designed to reduce noise interference from outside the structure. Unfortunately, modern building construction tends to make exact models of the structure difficult. Accurate transmission calculations would require knowing, not only what materials were used, but also where the studs, pipes, insulating materials, and especially air gaps between any of the materials are located. This information is just not available. What is available, however, are average transmission loss estimates for walls of similar make and construction. The same thing can usually be found for a variety of other materials that might also affect transmission, including windowpanes, doors, and even floor constructions. These then can be combined together to produce a single estimate of the transmission loss due to a given wall.

If, for the wall seen in Figure 3.10, we have the transmission coefficient ( $\tau_i$ ), and the surface area of each material exposed to the sound source ( $A_i$ ) then the combined



 **Oak Door**     **Glass Window**  
 **Air Gap**     **Painted Brick Wall**

Figure 3.10. An example outer wall of a house used in estimating transmission of sound. This wall has 4 separate materials blocking entrance to outside noise: (1) Oak, (2) Glass, (3) Air, (4) Painted Brick. Even though the oak and brick walls are good at preventing transmission of sound, much noise will be transmitted through the air gap and the glass window.

transmission loss due to the wall can be estimated from the following equations found in [Raichel 2000]:

$$\tau_{combined} = \frac{\sum_{i=1}^n A_i \tau_i}{\sum_{i=1}^n A_i} \quad \text{Equation 3.9}$$

$$TL_{combined} = 10 \log_{10} \left( \frac{1}{\tau_{combined}} \right)$$

Where:

$A_i$  = Area of surface  $i$ .

$\tau_i$  = transmission coefficient of surface  $i$ .

$TL_i$  = transmission loss due to the wall

This equation makes the assumption that the source is far enough way that the sound hitting the wall is roughly the same at every location on the wall. While this estimate may not be ideal for sound sources inside a house, it is a good enough approximation for estimating the noise interference due to traffic, industry, or even airports, which are a common source of outside noise for home environments.

If the transmission coefficient is not available for a particular material in the wall then it is also possible to estimate the transmission coefficient given other properties of the material used:

$$\tau = \left( \left[ 1 + \eta \left( \frac{\omega m \cos \theta}{2\rho c} \right) \left( \frac{\omega^2 B \sin^4 \theta}{c^4 m} \right) \right]^2 + \left[ \left( \frac{\omega m \cos \theta}{2\rho c} \right) \left( 1 - \frac{\omega^2 B \sin^4 \theta}{c^4 m} \right) \right]^2 \right)^{-1} \quad \text{Eq. 3.10}$$

Where:

$\rho$  = the density of air

$c$  = the speed of sound in air

$\theta$  = the angle of incidence on the panel

$m$  = panel mass density per unit area

$\omega$  = frequency

$B$  = panel bending stiffness

$\eta$  = a composite loss factor

In practice, estimating transmission loss with any degree of accuracy is very difficult. Even with the ability to estimate transmission coefficients and some idea of the

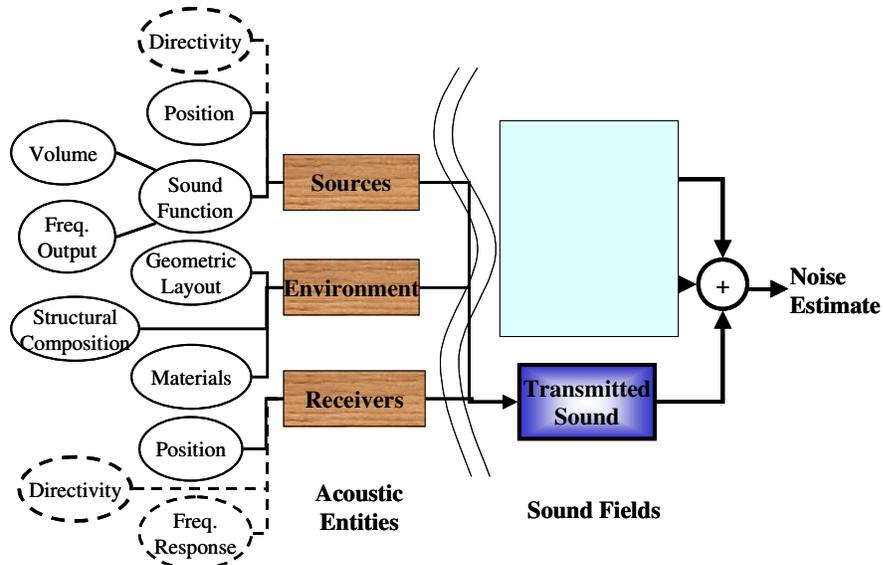


Figure 3.11. The information from each acoustic entity necessary for building a transmitted sound estimate. Estimating transmitted sound requires information about not only the sound source, but also the environment on both sides of the wall, and material properties.

structural composition of the wall, identifying all possible paths for sound to travel through the wall is unlikely, so the transmission loss equations can only provide a good initial estimate of how much sound is lost. To identify how much sound is hitting the wall in the first place, the robot needs to build direct and/or reverberant field estimates for other side of the wall. Figure 3.11 summarizes the large amount of information required to estimate transmitted sound.

Despite the apparent problems with acquiring so much information to build transmitted sound estimates, however, this area of the sound fields model may be of great importance to many applications. The reason is that cars, trains, airplanes, etc all impact everyday environments on a regular basis, transmitting significant quantities of noise through the walls from outdoors. Although a robot cannot predict exactly when a car will

pass by, it can use its knowledge of transmission loss to improve its performance by predict where sound levels will be higher or lower due to these external noise sources. This, however, is left for future work.

### Structure-Borne Sound

Another type of transmitted noise is structure-borne sound. Structure-borne sound is sound that reaches the receiver in at least part due to the vibration of a solid structure. Direct transmission through the wall is technically structure borne, because the wall must vibrate to generate the noise on the other side, but direct transmission is not usually what is meant by structure-borne sound. A good example of structure borne noise might be the plumbing in the house. As the pipes are all highly interconnected, and pressurized, a vibration at one end could easily generate noise at the other end of the house. Another good example are HVAC (heating, ventilation, and air conditioning) systems in office buildings. Besides the noise generated from the moving air, an HVAC in the basement can generate vibrations that travel through the building's frame to cause noise in seemingly arbitrary locations on other floors.

Of course, for a robot that is working in a small environment, structure-borne noise can often be modeled as a new source seemingly co-located with the wall. Transmission due to unobserved noise sources may also, sometimes, be modeled this way. If the robot needs to know, however, the relation between certain structurally-borne noise and an HVAC system (or any other known sound source) then there are some ways of estimating the structure-borne noise. Most commonly, identifying the transmission due to structure borne sound is done experimentally. If two noise sources are known to

be interconnected, then samples are taken by hand at different frequencies to determine the exact relation. A robot with appropriate a priori knowledge, such as the fact that all water noise in the house should be inter-related, could also do this same task.

There has also been some limited work in using Statistical Energy Analysis (SEA) to estimate structure-borne effects. Like direct transmission, however, the models for SEA have to be fairly reliable in order to demonstrate any real accuracy, but there are some available programs that have been successfully demonstrated for analyzing building interiors [Koizumi et al. 2002], so that might be something to check out in the long run.

### 3.4 CHAPTER SUMMARY

This chapter has focused on identifying the nature of acoustic awareness. More specifically:

- **What does it mean to be acoustically aware?**

Being acoustically aware means a coupling of action with knowledge about sound flow through the acoustic environment. There are, however, two levels of awareness, reactive vs. deliberative, that could be used for navigating a robot with respect to the auditory scene. Reactive awareness has been demonstrated in many of the applications in Chapter 2, as the robot listens, and then reacts to its current situation. To generalize the application to multiple environments and different types of tasks, we are focusing in this dissertation on the second type of awareness. A deliberative awareness of the auditory scene emphasizes the importance of

knowledge of sound flow in the environment for guiding robotic navigation.

- **What types of information can, or should, an acoustically-aware robot acquire?**

The soundscape can be divided into three primary acoustic entities: sound sources, paths, and receivers. The sound source is the source of noise propagating through the environment. The path entity incorporates knowledge of the environment to identify how the sound travels from the source to the listener. Finally, the receiver describes the listener, microphone or human ear, that has its own limitations on what it can hear, and from what angles.

- **How can the information be combined together to estimate sound flow?**

The theory of sound fields was identified in this chapter as a convenient framework in which to insert information about the three primary acoustic entities, so as to generate a model of sound flow. The framework itself is not new, and, in fact, has been steadily improved over the past two decades by commercial software companies including Odeon and Catt-Acoustic. Although their exact computational methods are proprietary, their published works [Naylor] suggest they make use of both the theory of sound fields and wave-equation calculations to provide their estimates. What is new to this work, however, is the use of any part of this framework for real-robot applications. The theory of sound fields

provides a necessary middle-step along the path to an acoustically-aware robot.

In the following chapters, we will build parts of this generic model of information and sound flow estimates into a real robotic application. Chapter 4 describes how to autonomously gather the necessary information for constructing models of both the direct and reverberant field, and demonstrates how lacking some information can still allow for useful models of sound flow through the environment. Chapter's 5-7 then describe robotic applications, demonstrating how to move the theoretical models from this chapter onto real robots and showing the advantages of being acoustically aware.

## CHAPTER 4

### ACQUIRING KNOWLEDGE ABOUT THE AUDITORY SCENE

In Chapter 3, we used the theory of sound fields to identify information useful to a mobile robot in understanding the flow of sound through the environment. With knowledge about the receivers, the sources, and the paths through the environment, a model can be created of the auditory scene to guide the robot in improving its performance. From where, however, can a robot reasonably expect to acquire this information? When not available a priori, the answer, by necessity, must be that the robot can determine this information using the sensors available to it. In this chapter, we focus on the problem of how, answering the second sub-question posed in Chapter 1. How can we combine data from multiple sensors to build effective representations of the acoustic environment?

The remainder of this chapter is organized as follows. The first section discusses existing work in localizing sound sources with static-mounted arrays and mapping the environment. These domains have been well studied as part of other problems for many years, and have produced well-established algorithms that can be used as a basis for further work. The next two sections build on this groundwork to extend the problem of sound localization to the more general auditory mapping domain. Multiple simultaneously operating sound sources are localized in 2-3 dimensions relative to the moving robot, and their volume and directivity are determined. Then, in the path domain, how to extract the geometric layout is discussed, and reverberant fields are created. Finally, from these sources of information we then build models of sound flow using the

equations in Chapter 3 and compare the results to sample-based representations of the environment.

#### 4.1 BUILDING BLOCKS

The development of an acoustically-aware robot has really been made possible by the scientific advances in three fields. The first field, architectural acoustics, contributed the mathematical framework described in Chapter 3 for combining information together into estimates of sound flow through the environment. The second two fields then contributed to the building of tools, and representations, from which the necessary information for the sound fields framework is extracted.

From the field of mobile robotics, the recent successes [Thrun 2002] in localizing the robot with respect to the environment and its past locations have contributed heavily to this work. Without relative position data a robot can be reactively aware to the auditory scene, sampling the environment and making decisions based primarily on local information. By incorporating knowledge of where the robot has been, the robot can now fuse its disparate collection of sampled data together into spatial representations of the environment, predicting where to move and where to avoid. Furthermore, with localization information, a robot can plan paths through the environment using those representations as a navigation guide.

From the field of digital signal processing, the last of the three scientific advances has been the work in localizing sound sources using arrays of microphones. Driven by interests in teleconferencing and military applications, researchers have developed reasonably robust algorithms for estimating angles and, sometimes, distances to the

sound sources in the environment. Although the developed solutions for acquiring distance and angle need some adjustments for robotic deployment (Section 4.2.1), the underlying algorithms are essential in acquiring a critical piece of knowledge about the auditory scene, the angle to the sound source.

Together, these three fields together will serve as the building blocks from which we can build additional representations. They will assist in gathering the information necessary for building an acoustically-aware robot.

#### **4.1.1 ROBOT MAPPING AND LOCALIZATION**

The problem of mapping the environment has been of great interest to the robotic community for many years, as maps are convenient tools for planning and sharing information with other robots or human observers. One of the simplest methods for building such a map makes use of robot localization, i.e. a robot knowing where it is located with respect to its previous positions, to spatially fuse sensor data into maps. The evidence grid [Elfes 1992], or occupancy grid, is the representation through which data is combined to predict the probability of something occurring in each grid cell in a discrete map of the environment. Traditionally the sensor data being combined together are the robot positions, and the laser or sonar readings to obstacles in the environment, resulting in obstacle maps useful for guiding robotic navigation (Figure 4.1, Bottom). In Section 4.4, we will use such an obstacle map representation for building reverberant field models. The evidence grid representation, however, is not limited to sensing obstacles. In Section 4.2, we will apply the same representation to sound source localization, estimating the likelihood of a sound source appearing within each grid cell.

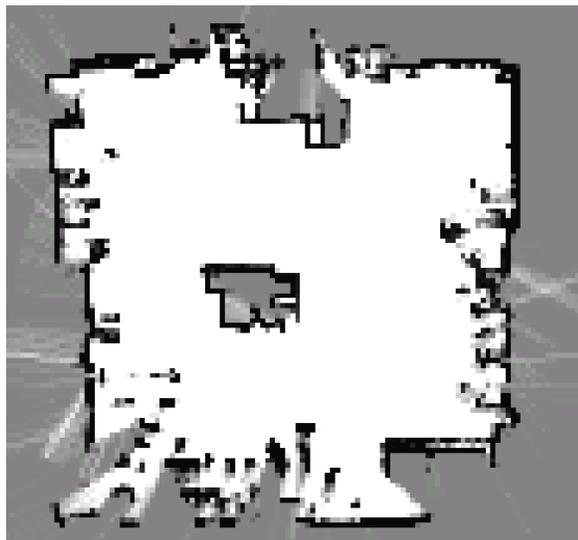


Figure 4.1. The B21r mobile robot (Top) and the obstacle map it created using the continuous localization algorithm (Bottom).

Unfortunately, reliably determining the robot's relative position to create such fused data maps can be a very difficult problem. The extended Kalman filter is an algorithmic method designed to do just that by fusing data together from a variety of sensors, including GPS, accelerometers, gyroscopes, odometric positions sensors, etc. Its accuracy, however, while substantially better than that of individual sensors, can still vary wildly with the precision, and cost, of the sensors being fused together. Simply placing a map building algorithm on top of this fused localization process may not be the best way to build an accurate map. For this very reason, the problem of localizing the robot has often been combined with mapping the environment. The combined problem is called the simultaneous localization and mapping problem (SLAM). This combined field has been actively researched for over a decade, and many good algorithms have been developed to exploit the advantages of mapping the environment while minimizing the disadvantages and uncertainty of robotic movement. A good source for specific details about many of these algorithms can be found in [Thrun et al. 2005].

After many years of development, robotic localization and mapping tools have recently become available to the wider mobile robot research community. Although such tools still have inherent problems in mapping dynamic or large scale environments [Thrun 2002], they are reasonably robust in structured indoor environments spanning a few rooms. For this dissertation, we made use of two such freely available tools to build two-dimensional maps of the environment and localize the robot using a SICK laser measurement system (LMS). The first such tool is based on the continuous localization algorithm [Schultz and Adams 1998], which extends the evidence grid representation to the simultaneous localization and mapping problem. This localization algorithm was

used on an iRobot B21r located at the Naval Research Laboratory, Center for Artificial Intelligence. Both the robot and its evidence grid map can be seen in Figure 4.1.

The second tool utilized for robot localization came bundled with the Player/Stage software for mobile robotic control [Gerkey et al. 2003]. The pmap software developed by Andrew Howard [Howard 2004] uses particle filters to build a discrete, probabilistic representation of the environment. This representation is then used in real-time by a mobile robot to localize itself using an adaptive monte-carlo localization driver equipped with Player/Stage. This combined solution was deployed on an ActivMedia Pioneer 2-dx robot in the Mobile Robot Lab at the Georgia Institute of Technology. The fully equipped robot and the resulting map can be seen in Figure 4.2.

#### **4.1.2 SOUND SOURCE LOCALIZATION**

The problem of sound source localization is one of the fundamental issues in modeling the acoustic environment: from where does the sound originate? Before we can predict the effects on the surrounding environment, even if that environment is assumed anechoic or otherwise acoustically simple, we must know from where the sound is being generated.

The fundamental property of sound that drives most localization algorithms is its finite, and relatively low (compared to light) speed as it propagates across the room. Given an array of microphones in an anechoic environment, the signal received at each microphone should be the same signal, only delayed by different amounts. Therefore, if we knew the delay between the arrival of the signal at each microphone we could also identify the position in environment from which the sound is originating. Assuming a

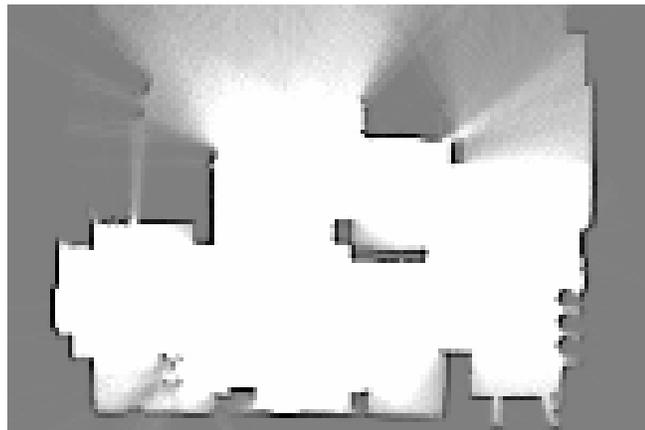


Figure 4.2. The Pioneer2-dxe robot (Top) and the map of the Mobile Robot Lab at Georgia Tech created using the PMAP software (Bottom).

constant speed of sound ( $c = 343$  m/s), the time ( $T$ ) required to travel the distance ( $d_{l,m}$ ) from a source at position  $L$  to a microphone at position  $m$  is the distance traveled divided by the speed of sound:

$$T(l, m) = d_{l,m} / c \quad \text{Equation 4.1}$$

Therefore, the delay between the signal arriving at the microphone located at  $m$  and the signal arriving at the microphone located at  $n$  is:

$$\text{delay}(m, n) = (d_{l,m} - d_{l,n}) / c \quad \text{Equation 4.2}$$

This delay is usually referred to as the time difference on arrival (TDOA), and measuring it can be difficult. If the microphones are in close proximity to each other, as is typical for most on-robot microphone arrays, then the delay between the signal's arrival at each microphone is very small. At 343-m/sec, microphones that are 0.3-m from each other (a likely maximum for a small robot array) experience maximum delays of less than 1-msec when the sound source is in line with both microphones. When the source is not in line, then this delay decays to 0-msec as the angle approaches perpendicular to the center of the array (Figure 4.3). With such small delays, accurate measurement in the time-domain is impossible. Instead, the signals recorded at each microphone are usually converted to the frequency domain using the Fourier transform, where signals can be compared at much finer delay increments than in the time domain.

The next problem in estimating the TDOA is comparing the signals. In any real environment, the signal of interest will be corrupted by some amount of noise, either from other ambient noise sources or reverberation from the environment. Therefore, in

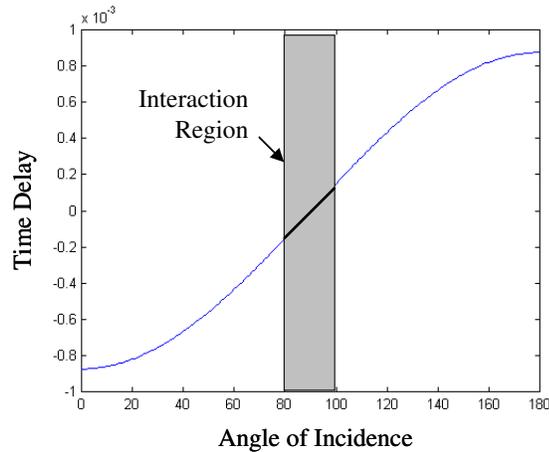


Figure 4.3. Measurable time delay between signals arriving at each microphone vs. the angle of incidence. This graph assumes 0.3-m between microphones in a binaural array, and a 1-m distance to the sound source. The gray region indicates the face-to-face interaction region for a binaural array mounted as ears on a humanoid.

actually, the microphones are measuring multiple signals arriving from all different directions. Which signal is the robot interested in localizing? If the type of signal to be measured, such as speech, is known, then a filter can often be applied to the incoming signal to significantly reduce the effects of ambient noise sources. For reverberant effects, however, or when the ambient noise sources mask the same frequencies as the source being localized, a filter cannot completely clean the signal. If greater knowledge about reverberant paths or knowledge of ambient noise sources is available, then noise cancellation techniques can go beyond filtering to clean the signal within the frequency bands of interest. Even with both filtering and noise cancellation, the signal will rarely be free of noise, so any method for comparing signals recorded at each microphone needs to take into account the existence of noise.

## Adaptations to Robotics

Overcoming these problems in microphone arrays small enough to fit on a robot have generally been restricted to three types of algorithms. The first type of algorithm uses the steered response of a beamformer for identifying the angle to one or more sources [DiBiase et al. 2001]. The advantage of using a beamformer, commonly used for combining acoustical signals in voice capture applications, is that it can be tailored using filters for the specific environment and placement of the microphone array. The basic algorithm uses the general TDOA principle to delay the incoming signal from each microphone some amount, effectively maximizing the energy for a specified angle. Localizing a source is then identifying the angle of incidence with the greatest energy. The drawback to beamforming systems can be their computational complexity. Especially once distance and environmental customization is included in the estimation, the amount of processing can be orders of magnitude greater than other localization methods.

The second type of algorithm uses high-resolution spectral estimation to locate sound sources. The MUSIC algorithm (MUltiple SIgnal Classification) is such an approach that has recently been applied to robotic platforms for speech localization [Argentieri and Danes, 2007]. Adapted from the field of high-resolution spectral analysis, these spectral estimation techniques are designed to handle multi-source localization problems. The difficulty with these approaches, however, is the amount of information required for high-resolution localization. An uninformed system requires many assumptions that decrease its effectiveness, especially under reverberant conditions [DiBiase et al. 2001].

The final approach, and the one most commonly used in robotic systems, is to calculate time-delay estimates using a generalized cross correlation algorithm [Martinson and Dellaert 2003; Blisard et al. 2007; Valin et. al. 2005]. This approach estimates the energy associated with a number of specified angle/distance pairs, and then maximizes the energy to localize a sound source. Although the specific generalized cross correlation approach has also been used with traditional beamforming, the real difference is in the computationally simpler weighting scheme. Instead of customizing the algorithm for a particular environment and/or signal type, an uninformed weighting localizes sources with a wide variety of sound functions. As this final approach is the one used for this entire dissertation, we will now describe this algorithm in greater depth.

#### Time Delay Estimates Using Generalized Cross Correlation

The solution most commonly employed by TDOA estimation algorithms is generalized cross correlation (GCC). For discrete-time signals, cross-correlation provides an estimate of similarity by bit-wise multiplying two signals together and summing the result. Two identical signals should produce an energy value equal to the sum of square of the signal, where two random signals would be significantly less. To find the TDOA estimate, the cross correlation algorithm is run many times, each time delaying the second signal by some amount. The delay that produces the highest cross-correlation energy is the best estimate of the TDOA between a microphone pair. Cross-correlation in the frequency domain works on a similar principle, maximizing the energy between microphone pairs, only now the cross-correlation is done in the frequency domain. Equation 4.3 demonstrates this equation for one pair of microphones.

$$F_l = \int_{\omega} W(\omega) M_a(\omega) \overline{M_b(\omega)} e^{-j\omega(\text{delay}(a,b))} d\omega \quad \text{Equation 4.3}$$

where ( $M_a$ ) is the Fourier transform of the signal received by microphone (a),  $\overline{M_b}$  is the complex conjugate of the Fourier transform of the signal received by microphone (b), ( $\omega$ ) is the frequency in [rad/s], and ( $W$ ) is a frequency dependent weighting function. Called the “phase transform” (PHAT) [Mungamuru and Aarabi 2004], this weighting scheme depends on the current magnitude at each frequency to evenly weight all frequencies present in the signal (Equation 4.4):

$$W(\omega) = \frac{1}{|M_a(\omega)| |M_b(\omega)|} \quad \text{Equation 4.4}$$

The position ( $l$ ) that corresponds to the highest cross correlation value ( $F_l$ ) is then the most likely position to contain the sound source.

Besides the PHAT weighting scheme, a weighting scheme that is also commonly employed in sound localization is maximum likelihood (ML) weights. This weighting scheme is most appropriate for tracking sound sources, such as human speech, which are not always present in the environment and cover a broad frequency spectrum. ML weights use knowledge of the noise (i.e. ambient sound not being tracked) affecting each microphone to attach greater weight to frequencies present in the tracked signal that are not present in the ambient sound. Equation 4.5 demonstrates the creation of ML weights from the noise spectra ( $N_a$  and  $N_b$ ) corrupting each microphone:

$$W(\omega) = \frac{|M_a(\omega)| |M_a(\omega)|}{|N_a(\omega)|^2 |M_b(\omega)|^2 + |N_b(\omega)|^2 |M_a(\omega)|^2} \quad \text{Equation 4.5}$$

The drawbacks to ML weights, however, often prevent their regular use. Tracking human speech works well, particularly in conjunction with a speech detection algorithm, because speech sounds are significantly different in frequency from common ambient noise sources like as HVAC systems. Other types of sound sources, unfortunately, are not tracked as well using ML weights. In particular, counter-weighting ambient noise is detrimental to tracking those ambient noise sources, so ML weights should not be used when the sound sources of interest cannot be turned off. Furthermore, sound sources with frequency signatures similar to the ambient noise will be minimized using this weighting scheme. This problem is especially difficult in indoor environments where air vents, fans, and to a lesser extent, plumbing noise, may be of interest to a robot.

Using either PHAT weighting or ML weights, Equation 4.2 still only applies to a single microphone pair. When the microphone array consists of more than two microphones, it can obviously be broken up into pairs of microphones, but how can the results be combined together? The simplest method for extending the GCC algorithm is to identify not the set of delays, but rather all possible locations in the environment from which sound might be originating. Knowing the speed of sound and the distance to the microphone array, we can predict the TDOA associated with a single microphone pair, and calculate the GCC energy for that location in the environment. With multiple microphone pairs (for  $n$  microphones in an array, there are  $n!/(2!(n-2)!)$  sets of two microphones), we use the same location method, but now identify the delay associated with that location for every microphone pair, calculate the GCC energy for each microphone pair, and sum the resulting energy for that location over all microphone

pairs. Equation 4.6 restates the original GCC equation in terms of the location, summing across all microphone pairs  $\{a,b\}$  in the array:

$$F(l) = \sum_{a=1}^N \sum_{b=1}^N \int_{\omega} W(\omega) M_a(\omega) \overline{M_b(\omega)} e^{-j\omega(T(l,a)-T(l,b))} d\omega \quad \forall a \neq b \quad \text{Equation 4.6}$$

This version of the generalized cross-correlation algorithm is often referred to as a spatial likelihood [Mungamuru and Aarabi 2004] as the resulting GCC energy directly corresponds to the likelihood of a sound source occurring in any given location. Figure 4.4 shows a contour plot of a spatial likelihood created for a  $6 \times 6 \text{m}^2$  grid centered about the microphone array. Appendix B.1 provides pseudocode for creating this grid representation of a spatial likelihood.

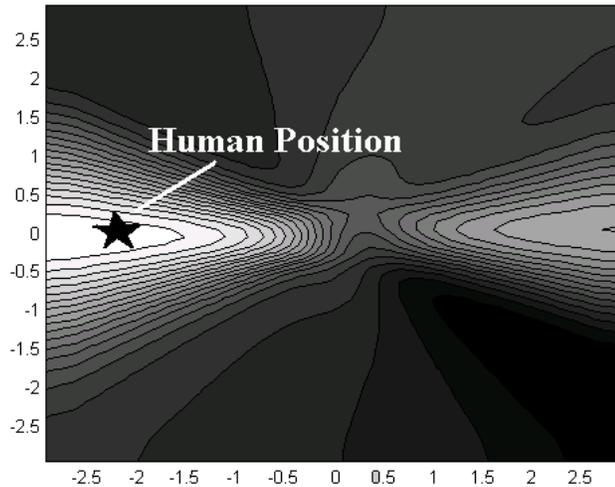


Figure 4.4. A contour plot of a spatial likelihood result for detecting human speech. Light areas are considered more likely. This result demonstrates the common problem of a strong angular performance, but poor distance estimates.

Figure 4.4 also demonstrates the drawbacks to the TDOA approach. In theory, given enough microphones in an array, it should be possible to exactly localize upon the source generating the noise. In practice, however, given the small distances between microphones in an on-robot array, as well as the levels of ambient noise and echoes from the environment, we have observed high amounts of error in the localization from one location. That error tends to be concentrated mostly along the axis stretching from the center of the array out through the sound source location, meaning that the cross correlation results are generally better at estimating angle to the sound source rather than distance.

#### Identifying Angle and Distance

In general, the problem of accuracy using on-robot microphone arrays is a problem of the domain and not the algorithm. Beamforming suffers from a similar problem in identifying distance to the sound source. With an on-robot array, there is just not enough difference between the signals arriving at each microphone to extract two-dimensional coordinates of the sound source from a single position. To overcome the missing distance problem, a common solution is to widely distribute the sensors about the environment. With a wide distribution of microphones, ideally placing the sound source within the convex hull created by the array, there have been a number of successfully deployed systems that could provide 3D coordinates for sound sources [Girod and Estrin 2001; Nakadai et al. 2006]. Work by Thrun demonstrates how not only the sources can be localized in this situation, but also the microphones themselves [Thrun 2005].

When a widely distributed network of microphones is available, a robot can certainly use this localization system for modeling sound propagation through the environment. A robot of large enough size could even be used to distribute the microphones in the first place if the situation is appropriate [Zhang and Sukhatme 2005], as could a team of robots, each with their own microphones [Martinson and Dellaert 2003; Parker et al. 2003]. However, such a system is not always available. In military or police situations, the robot is likely to be in hostile territory where people or robots cannot move in ahead of time to place microphones. Alternatively, maybe the funds are simply not available for placing, synchronizing, and utilizing large numbers of microphones distributed about the area. When the robot does not have the assistance of a microphone network to localize the sound source, there is another solution to acquiring more accurate coordinates. The robot can move itself to a new location and using its small microphone array with bearing dominated measurements to triangulate on the source from a different angle. In Section 4.2.1, we will demonstrate how this fusion of data can be done in real-time to extract sound the relative position of one or more sound sources in the environment.

## **4.2 REPRESENTATIONS FOR CHARACTERIZING SOUND SOURCES**

The minimal information necessary for estimating sound fields, or the barest of information required by the mathematical framework, is the position of the sound source and the average volume at which it is producing noise. From this information alone, we can estimate the effects of the direct field up to some arbitrary distance about the source. As we acquire more information about the sound source, these effects can be estimated

with greater detail and accuracy, ultimately contributing to estimates for both the direct and reverberant fields. If the information about the sound source is not known a priori, however, then what information can a robot collect to build the sound field estimates?

In this section, we will discuss the set of representations, or tools, that a robot can use to acquire the two-dimensional location of one or more sound sources in the room, and then determine their volume, directivity, and sound function. These representations will build on the mathematics presented in the previous chapter. For now, we are focusing on medium to long duration sources that can be expected to remain relatively static over the time the robot is collecting this information. In the long run, however, more information about the sound source function could also be collected by a mobile robot and incorporated into the sound fields framework.

#### **4.2.1 IDENTIFYING SOURCE LOCATION IN 2D**

For the purpose of modeling sound sources in the environment, we need sound source localization that includes more than just the angle from the center of the array. Specifically, assuming that every sound source can be represented as a point source, we need to identify a specific centroid coordinate in 2 or 3 dimensions. The reason for this simplifying assumption (point source) is that even though sound is usually generated from a vibrating surface, and the sound-fields framework can work with a more complicated model, sound source localization algorithms typically operate on the assumption that the sound is loudest from a given point. Greater information about the nature and size of the vibrating surface, such as might be needed accurately describing

the effects of larger sources, may require more a priori information about the source being localized or more sensory information (e.g. camera).

As mentioned at the end of Section 4.1.2, the way to acquire these 2D coordinates using an on-robot microphone array is through robotic movement. By moving the robot from place to place in the environment, we can use combine the results of an angular source localization algorithm to triangulate upon the two dimensional location of the source centroid.

Depending on which underlying sound localization algorithm is being used, there are different possible methods, concurrently developed, for triangulating on sources in the environment. Using a steered-beamformer, the first method identifies the angle to one or more sources, and then mathematically minimizes the squared error identify sound source locations [Sasaki et al. 2006]. The drawback to this approach, however, is that if the error is high in the original beamforming results, the algorithm may not localize the sound source at all, or have a high false-positive rate. In their paper, the authors developed a specialized 32-element microphone array with an ideal beam pattern for their scenario to counter this effect. Even then, they still could not localize sound sources in the vicinity of walls or other highly reverberant areas of the environment.

As an alternative to using beamforming, a second approach developed for this dissertation [Martinson and Schultz 2006] uses time delay estimates (particularly, spatial likelihoods) to localize sound sources with a moving robot. While the robot moves through the environment, it creates a spatial likelihood for each collected sample, and combines the disparate measurements together into an auditory evidence grid. Unlike the concurrently developed beamforming approach, the auditory evidence grid representation

does not require specialized hardware. Furthermore, although better microphone arrays and anechoic environments still demonstrate the best performance, the algorithm degrades gracefully in the presence of noise, allowing for a variety of environmental and hardware configurations. In the worst-case configuration, using the bare minimum microphone array (i.e. binaural) in a reverberant environment, sources can still be localized to some degree of accuracy.

The remainder of this section on localizing sound sources will focus on this second approach using auditory evidence grids. We will first describe the algorithm, including a review of evidence grids in general, and how to adapt them to auditory information. Next, we will investigate the performance of the algorithm under different hardware configurations, robotic movement strategies, and sound source types, with the goal of automating the process. With this experience, we then devise an automated process for extracting the sound source coordinates from the resulting grid and devise, and test, some autonomous robotic movement strategies for accurately estimating the location of sources in the environment.

Some of this work has appeared in other publications, prior to this dissertation. The initial description of how to construct an auditory evidence grid, along with the first phase of testing was originally reported in [Martinson and Schultz 2006]. Then in [Martinson and Schultz 2007] the robotic movement strategies and the second phase of testing for measuring accuracy were described.

## Auditory Evidence Grid - Algorithm

The underlying algorithm for auditory evidence grids is the same evidence grid algorithm used in creating obstacle maps of the environment (Section 4.1.1). Only instead of using measurements to obstacles, as returned by lasers or other similar sensors, we are using estimates of distance to sound sources, collected aurally by a microphone array. The name auditory evidence grid, therefore, follows the change in information being mapped, from obstacles and spatial layouts to sound sources and auditory layouts.

As with spatial evidence grids, the auditory evidence grid uses Bayesian updating to estimate the probability of something being located in a set of predetermined locations (i.e. grid cell centers). Since we will be feeding the algorithm spatial likelihood measurements collected by the microphone array, that something being estimated is the probability of a sound source. Initially, it is assumed that every grid cell has a 50% probability of containing a sound source. Then as each new spatial likelihood is created from a sensor measurement, those probabilities for each grid cell are adjusted. For the simplicity of adding measurements together, we use the log odds notation to update the evidence grid. Equation 4.7, from [Thrun 2002], demonstrates this additive process for each new measurement

$$\log\left(\frac{p(SS_{x,y} | z^t, s^t)}{1 - p(SS_{x,y} | z^t, s^t)}\right) = \log\left(\frac{p(SS_{x,y} | z_t, s_t)}{1 - p(SS_{x,y} | z_t, s_t)}\right) + \log\left(\frac{p(SS_{x,y} | z^{t-1}, s^{t-1})}{1 - p(SS_{x,y} | z^{t-1}, s^{t-1})}\right) \quad \text{Eq. 4.7}$$

In these equations,  $z_t$  and  $s_t$  are the sensor measurement and robot pose respectively recorded at time  $t$ ,  $z^t$  and  $s^t$  are the set of all sensor measurements and robot poses recorded up until time  $t$ , and  $SS_{xy}$  is a particular grid cell in the evidence grid. Therefore,  $p(SS_{x,y}|z^t,s^t)$  is the probability of grid cell  $SS_{xy}$  being occupied given all

evidence collected up until time  $t$ , and  $p(SS_{x,y}|z_t, s_t)$  is the *inverse sensor model*, or probability that a single grid cell contains the sound source based on a single measurement.

The *inverse sensor model* used in this work is based on the spatial likelihood measurements described in Section 4.1.2. Every time a sample is collected from the microphone array a spatial likelihood is created by estimating the generalized cross correlation (GCC) energy over a set of pre-determined locations. For creating auditory evidence grids, we generally restricted the set of pre-determined locations to a 3-m radius about the robot, so as to limit the computational requirements of calculating a spatial likelihood. By itself, however, the spatial likelihood needs some additional modification for use with the evidence grid representation. In particular, we need a likelihood varying ideally from 0-100%, but the range of energy values returned is nowhere near that range. The energy from most unlikely to most likely in a single measurement often varies by a factor of  $10^4$ . Furthermore, different environmental conditions, source types, etc. can then shift the entire grid by a similar factor either higher or lower. Therefore, to estimate  $p(SS_{x,y}|z_t, s_t)$ , we use an inverse sensor model that scales and shifts each spatial likelihood measurement, so that the result lies between two chosen probabilities [ $P_{low}$  and  $P_{high}$ ] where the lowest cross correlation value resulted in a probability of  $P_{low}$  and the highest in  $P_{high}$ . Equation 4.8 shows how this scaling and shifting process is accomplished:

$$\begin{aligned}
 K_1 &= (F_{\min}(t) - F_{\max}(t)P_{low} / P_{high}) / (1 - P_{low} / P_{high}) \\
 K_2 &= (F_{\max}(t) + K_1(P_{high} - 1)) / P_{high} \\
 p(SS_l | z_t) &= (F_l(t) - K_1) / (K_2 - K_1)
 \end{aligned}
 \tag{Equation 4.8}$$

Where  $F_{\min}(t)$  and  $F_{\max}(t)$  are the lowest and highest  $F_1$  values calculated for the measurement taken at time ( $t$ ). To then extract the resulting  $p(SS_{x,y}|z_t, s_t)$  from  $p(SS_l|z_t)$  the robot pose ( $s_t$ ) is used to convert from local coordinates ( $l$ ) to global coordinates ( $x,y$ ). Figure 4.5 demonstrates an auditory evidence grid resulting from this process, localizing two radios in the environment. Appendix B.2 describes this creation process in more detail, providing pseudocode for the implementation used throughout this dissertation.

The remainder of this section on localizing sound sources in 2D is focused on testing the auditory evidence grid, and then automating the collection of samples and localization of sound sources. For all of these tests using the auditory evidence grid, the spatial likelihood results were typically scaled between  $[0.2, 0.95]$ , but this could be

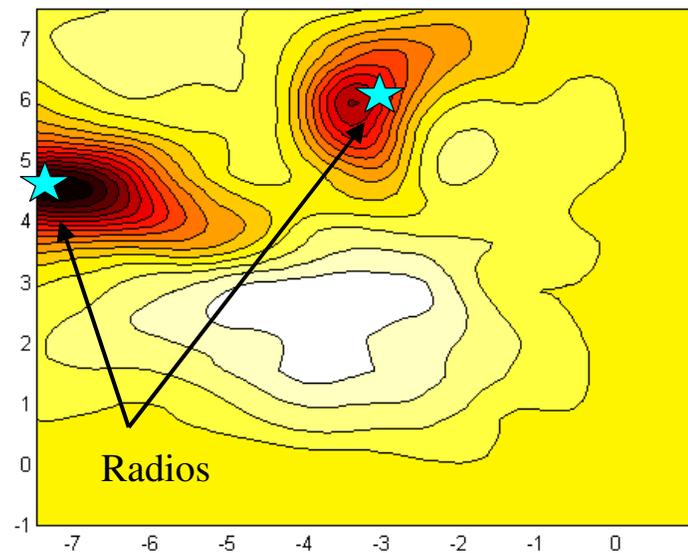


Figure 4.5. Contour plot of an auditory evidence grid localizing two radios. This contour plot combines 190 samples collected from a moving robot with four microphones. The darkest areas indicate the most likely sound source positions. To reduce the noise of a moving robot, a square sliding window (0.6-m in width) was used to produce smoother contours.

varied when tracking different types of sources. These scaling numbers were chosen empirically based on spatial likelihood reliability. As the spatial likelihoods would generally only point at one source at a time, areas not indicated with a high cross correlation result were not necessarily devoid of sources so setting the probability at 0 would unfairly penalize quieter sources. Similarly, spatial likelihoods could also make a mistake in the direction they pointed, and so 100% confidence was inappropriate in scaling the results.

### Testing Phase 1 – Investigating the Range of Auditory Evidence Grid Performance

Testing of the auditory evidence grid algorithm was performed in two phases, both taking place at the Navy Center for Research in Artificial Intelligence, located on the Naval Research Laboratory. In this first phase of testing, a human-operator manually tele-operated the robot in a loop in the vicinity of some set of sound sources to test the performance of the algorithm under different operating conditions.

- How many sources can be successfully localized?
- How does robotic movement affect the results?
- How many microphones need to be used?
- Does filtering the data assist in localization?

From the answers to these questions, we will next construct an algorithm for automating this process by extracting sound source coordinates and moving the robot for better sample collection.

This phase of testing, with some modification, was originally reported in [Martinson and Schultz 2006].

## Hardware Setup

The robot hardware used in this work was a B21R research robot manufactured by iRobot (seen in Figure 4.6). The robot is equipped with a SICK laser measurement system (LMS) mounted in the robot base, and two onboard computers for processing. Robot pose information is then provided by the continuous localization [Schultz and Adams 1998] algorithm, which uses a spatial evidence grid representation (different from auditory evidence grids) constructed from LMS range data and robot base odometry. The robot also has an additional SICK LMS mounted above the robot base and a full sonar ring that were not used in these experiments.

The equipment used for gathering the acoustic data was an array of (4) Audio-Technica AT831b lavalier microphones mounted at the top of the robot. These microphones were each connected to battery powered preamps mounted inside the robot body and then to an 8-Channel PCMCIA data acquisition board.

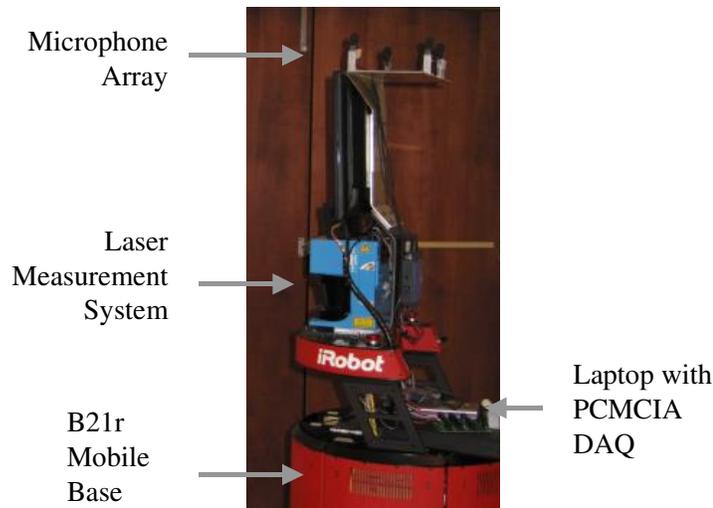


Figure 4.6. Fully equipped B21r mobile robot used for phase 1 testing.

## Results

To test the algorithm, the robot was run in 20 trials, varying two parameters: (1) the set of sources active in the environment, and (2) whether or not the robot was moving while gathering data. A total of 10 different configurations of sources were tested, where a source configuration is defined as a unique set of active sources in the environment. For the following trials, 9 sources were mapped by the robot: 2 human speakers (male and female), 1 tape recording of human speech, 2 radios playing different types of music, and 4 air vents in the laboratory. Figure 4.7 shows the relative positioning of each of the sources, along with the grid used for localizing the robot in the 12x12-m<sup>2</sup> laboratory. In

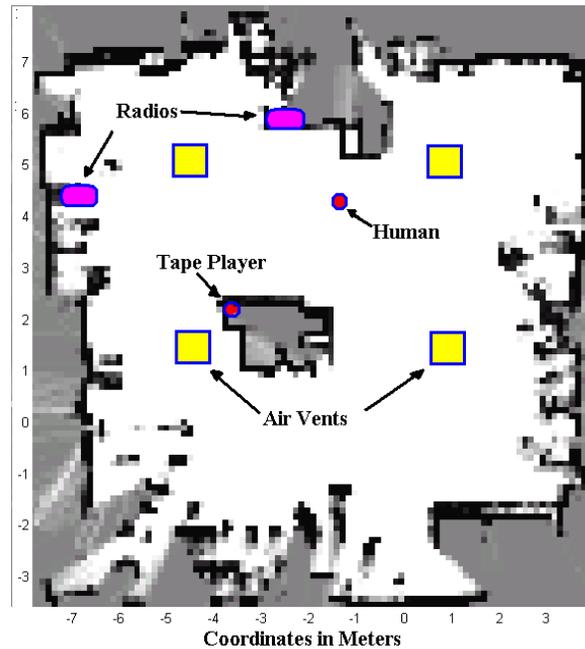


Figure 4.7. Spatial evidence grid used by the robot for localization with source positions shown relative to the obstacle positions in the room. Note that not all sources were active in every test.

general, the robot was not always exploring the entire area, but was instead restricted to a subsection in the vicinity of the sound sources of interest.

Robot movement during these tests was varied according to whether or not it was stationary while sampling audio data. In both cases, the robot was tele-operated in a large circle in the vicinity of the sound sources. In the first case, however, the robot would stop 6-7 times to gather samples of the auditory scene using its microphone array. In the second case, the robot would continue to gather microphone array measurements while it was moving. The reason for the different data collection techniques was to evaluate the effect of increased ego-noise on the robot, as movement increased the volume of wheel and motor noise generated present.

The results of all mapping experiments are shown in Table 4.1, where a successful test is defined as correctly placing a peak in the smoothed contour map within 1-m of the true source location for all active sources. The number of sources listed in this table does not include air vents. As the vents could not be fully disabled, they were on during all trials, but were too quiet to detect except when all other sources were disabled.

Table 4.1. The results of all phase 1 auditory evidence experiments.

# of Sources	# of Source Config.	Successes: Pausing while Collecting	Successes: Moving while Collecting
1	5	5	5 (4)
2	4	4	3
>2	2	1	1

In general, as demonstrated by the table results, the auditory evidence grid algorithm worked very well for mapping one or two sources. In only one test with two sources, did the robot fail to correctly map all of the active sources. There was an additional test using one active source, in which a phantom, or illusory, peak appeared strong enough in the evidence grid to be mistaken for a real source, but the active source was also found in the same evidence grid. Note that in both of these trials, the robot was moving while collecting data instead of stopping, so movement obviously did introduce some additional error, but the algorithm still succeeded in most cases to successfully map 1-2 sources.

Larger numbers of sources were not as successful, but this may have been due to the relative scarcity of samples. The trial that succeeded in localizing 3 sources had all three sources in a relatively small area, while the trial that failed involved a large area and multiple widely spaced sound sources (the air vents). In addition to these being relatively quiet sources, the robot did not sample equally in the vicinity of all sources due to the large area being sampled, thereby limiting the effectiveness of the localization results. A solution to this problem used in the following phase of testing is to investigate individual sources, sampling extensively in their vicinity to improve localization results, and ultimately gather sound source characteristics.

Besides the general sound source distribution problem, this first phase of testing was designed to test to reveal the effects of different design decisions on the quality of the map. What follows here is a discussion of those results:

- **Moving when gathering auditory data**

As was seen earlier, the evidence grid representation still works when the robot is moving while sampling, but more problems occurred in trials where the robot was moving than when not. There are two reasons for this decreased accuracy in evidence grid. The first reason is that, when moving, the robot pose estimation algorithm introduces more relative error into the representation. As the robot pose estimates are used to align overlapping spatial likelihood measurements, this results in wider, lower peaks in the resulting evidence grid. The second problem when moving comes from the louder robot ego-noise generated by the robots wheels and motors. If the robot is generating more noise when moving, than there will be more noise present in the environment that can partially or totally mask the active sound sources being investigated. Algorithmically, this results in degraded spatial likelihood results, and less certainty on the origin of the loudest sound. The effect of this on the resulting evidence grid is twofold: (1) poorer spatial likelihood accuracy results in more phantom peaks, making it harder to distinguish actual sources; and (2) rougher object contours will be evident in the final map.

- **Number of Microphones**

Many robots are now being equipped with a binaural microphone array (i.e. two microphones) to mimic human hearing, and there is no reason why spatial likelihoods cannot be computed using only 2 microphones. However, with a binaural setup, the accuracy of calculated spatial

likelihoods decreases in both distance and angle. So to test the effect of a binaural setup on auditory evidence grid, we reused the data from the same trials discussed earlier, and only used two microphones streams instead of all four to generate the spatial likelihoods. The resulting effects on the evidence grid from this binaural approach is actually very similar to those seen when moving while gathering audio data: (1) more phantom noise sources, or peaks in the evidence grid are generated, and (2) the object peaks have rougher contours. However, as demonstrated in Figure 4.8, the same sources were generally still evident for both 2 and 4 microphone configurations in most trials.

- **Map Resolution**

To detect sources in real-time the evidence grid and spatial likelihood grid cell size was a minimum of 0.3m. This is a relatively coarse resolution that may have affected the resulting accuracy. To determine exactly how the resulting map was affected, we recreated the trial maps at a higher resolution (0.1m) using the data collected earlier. The result of increasing resolution was that it shifted the center of the peak in the evidence grid towards a more accurate center. However, that center would have otherwise been included in a larger grid cell at a lower resolution, so it was not unexpected. Unfortunately, though, changing resolution did not appear to affect the creation of phantom peaks or rougher contours.

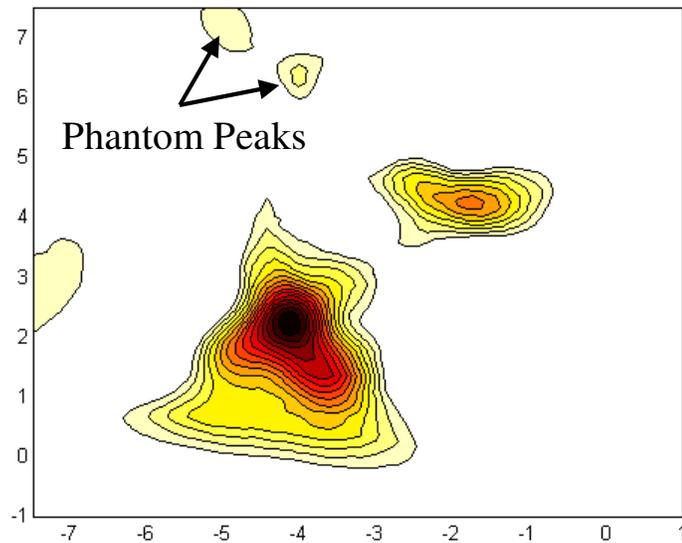
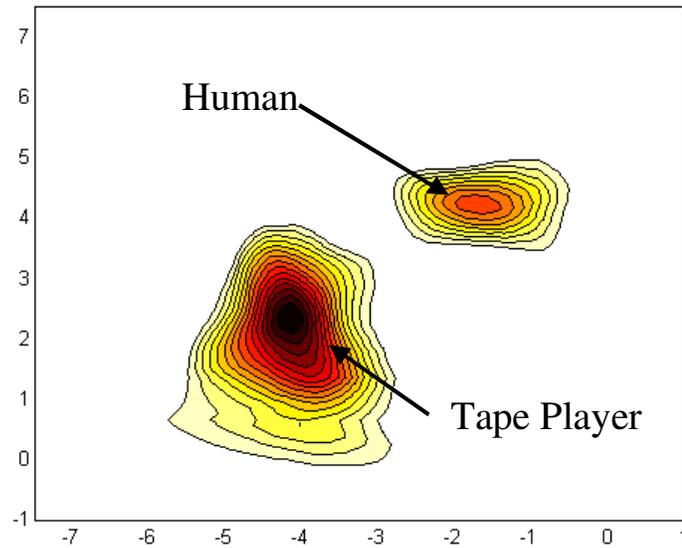


Figure 4.8. Auditory evidence grids localizing two speech sources (a stationary human speaker and a tape player ) from 463 data points collected at 6 positions. Both grids are created from the same 463 recorded samples (not all are speech) using either a 2 (Bottom) or 4 (Top) microphone array, and thresholded to display only points more than 50% likely. Note the rougher contours and phantom peaks found in the grid created by only 2 microphones.

- **Filtering the Sample Set**

If a priori knowledge exists about the set of sound sources being mapped, then another method for removing error from the map is to filter the sample set. One such filter tested was an rms threshold, like that employed in Linear Predictive Coding [Tremain 1982] for detecting speech over the telephone. The resulting maps for speech sources had smoother contours and better defined peaks. There is a tradeoff, however, in that fewer samples were used to create the maps in general, and that some source types (non-speech) were removed by this filter entirely.

### Iterative Clustering

After gathering enough data, the resulting auditory evidence grid representation estimates the combined likelihood of a source being located at any position. By itself, however, the auditory evidence grid does not localize sources in space. What we need for use with the mathematical framework discussed in Chapter 3 are two- or three-dimensional coordinates of the source, but, as seen in Figure 4.8, sound sources in the auditory evidence grid are merely peaks of varying heights and contours in the map. An algorithm for extracting coordinates from the grid is required.

By applying a threshold to the auditory evidence grid (Figure 4.8), we can see that sound sources appear as clusters in the map. Using a nearest-neighbor clustering algorithm [Duda et al. 2001] on these evidence grids, we can easily extract the two largest speech sources and estimate their centers. With the large number of samples used to build evidence grids in phase 1 nearest-neighbor clustering algorithm typically

produced errors of 0.3-m for clearly evident clusters such as seen in Figure 4.8 (Top). When using only two microphones, however, or moving the robot, phantom peaks often showed up in a thresholded grid. The first question we needed to address when using a nearest-neighboring clustering algorithm, was, therefore, how do we separate these from valid sound source positions?

The second question we needed to address when building the algorithm was discovered while watching the evidence grids form in real-time with more than three sources in the environment. As each new measurement was collected by the robot and added to the grid, all of the sources would appear for a time while the robot was in close proximity, but then, as the robot moved away from the source, one or more sources would be suppressed by new measurements. The reason for these suppressive effects over time is two-fold. First, it has to do with the nature of sound, as the sound volume and resulting GCC measurement will naturally fade both with distance from the source, and, of course, with variations in the source volume. Second, this suppressive effect is then further exacerbated by linearly scaling the GCC data. As mentioned in Section 4.1.2, each spatial likelihood measurement is most strongly associated with a single sound source. By scaling between two set values, however, each measurement that points at one source will effectively suppress the evidence grid in other areas not being pointed at, including areas containing another sources. Therefore, if too few measurements point at a source because it is too quiet or too far away, then the cumulative effect of the suppression may end up being greater than the cumulative positive effect.

### **Steps in the Iterative Clustering Algorithm**

*Step 1.* Scale and Threshold the map.

*Step 2.* Cluster points together using nearest-neighbor clustering.

*Step 3.* Localize the largest source.

*Step 4.* Identify spatial likelihoods pointing at the largest source.

*Step 5.* Create a new evidence without largest source.

*Step 6.* Repeat until all sources are found.

To overcome both the phantom peak problem and the suppressive effect over time, we developed an iterative approach to source localization, extracting one source at a time from the evidence grid. By taking advantage of the angular nature of spatial likelihoods, we can then match newly extracted sources with the spatial likelihoods that point at them. Firstly, this allows us to find more sources that might have been suppressed by those samples. Secondly, a minimum number of associated samples can be used as a convenient baseline for eliminating phantom peaks, which typically are created by one or two very strong echoes.

- **Step 1 - Scale and Threshold the Map**

To prepare the map for clustering, it is first scaled so that the most likely point is no more than 99% likely, and the least likely point is no less than 1% likely. A threshold of 75% is then applied to the map (or 1 in a log-likelihood grid) to eliminate points unlikely to contain a sound source. Lower thresholds were also tested, but often led to joint clusters when

sound sources were located too close together. Figure 4.9 demonstrate an evidence grid before (Top) and after (Bottom) thresholding.

- **Step 2 - Cluster points together using nearest-neighbor clustering**

A nearest-neighbor clustering algorithm is then used to collect all points together that are within 0.3-m of each other. Appendix B.2.1 provides algorithmic detail on accomplishing this clustering task.

- **Step 3 - Localize the largest source**

A weighted centroid of the largest cluster is calculated using the likelihood at each grid cell as the weight. Appendix B.2.1 describes how to calculate this weighted centroid as part of the clustering process. If the cluster is larger than 0.5-m<sup>2</sup> in area (determined empirically), then it is identified as a potential sound source and its centroid is used as the source position. If the area is too small, then no sources were successfully detected using this map. Figure 4.9 (Bottom) demonstrates the thresholding and clustering process.

- **Step 4 - Identify the set of samples that point at the largest source**

While every spatial likelihood measurement does contain information about multiple sources (and echoes), each spatial likelihood measurement at the time of sampling. Appendix B.1.1 provides pseudocode for estimating the most likely angle detected by a single spatial likelihood measurement. So for each measurement we can calculate the most likely angle to the “loudest” source by compressing the log-likelihoods along the angular axis at some increment  $\delta$ . Let  $F_\theta$  be the log-likelihood of the

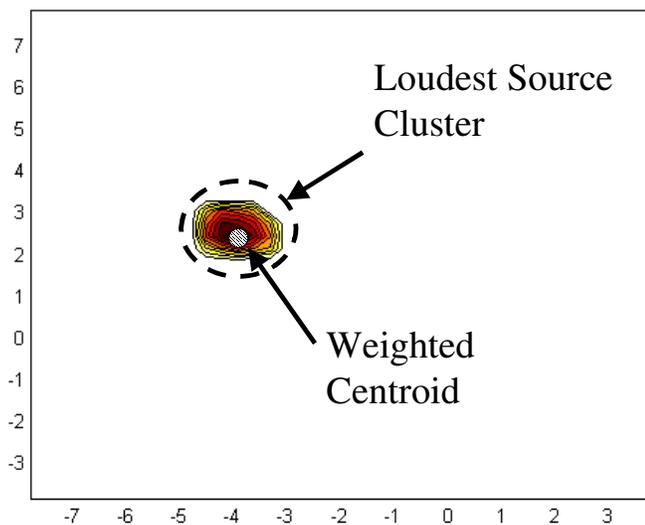
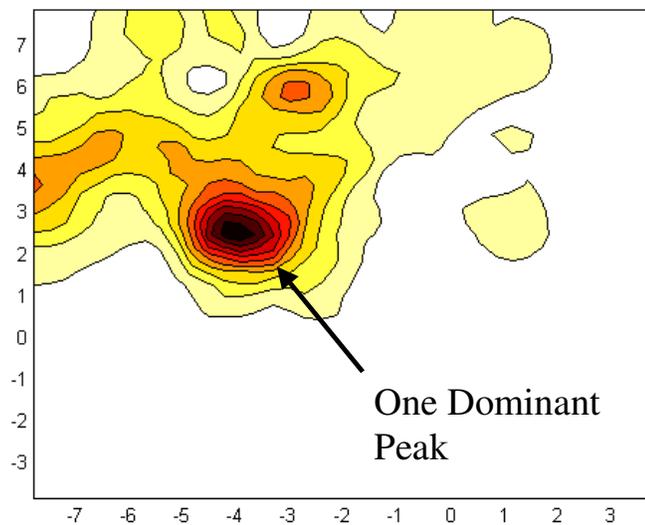


Figure 4.9. Stepping through the iterative clustering process, first round. (Top) Original auditory evidence grid with one dominant peak. (Bottom) Thresholded result with largest cluster circled.

detected source being located along angle  $\theta$ , and  $F_{\phi,r}$  be the log-likelihood of the sound source being located at cylindrical coordinates  $(\phi,r)$ . Then the most likely angle towards the detected source is the angle  $(\theta)$  with the highest log-likelihood:

$$|\theta_{best} + \theta_{robot} - \alpha_{source}| \leq threshold \quad \text{Equation 4.9}$$

Now, using this notion of most likely source angle, we can determine which spatial likelihood measurements actually point at sources found:

$$F_{\theta} = \sum_{r=0}^3 \sum_{\phi=\theta-\delta/2}^{\theta+\delta/2} \frac{F_{\phi,r}}{1 - F_{\phi,r}} \quad \text{Equation 4.10}$$

where  $\theta_{best}$  is the most likely angle as predicted by the spatial likelihood function in local coordinates,  $\theta_{robot}$  is the orientation of the robot in global coordinates, and  $\alpha_{source}$  is the angle from the robot to a detected source in global coordinates. If the difference between the angle to the source location (as predicted by the evidence grid) and the most likely angle (as predicted by a single spatial likelihood measurement) is less than some threshold, then that measurement is estimated to be pointing at the source.

- **Step 5 - Create a new auditory evidence grid without the largest source**

Using just those spatial likelihood measurements that are not estimated to be pointing at a previously localized source, create a new auditory evidence grid. Figure 4.10 demonstrates a new auditory evidence grid created from this reduced sample set. Appendix B.2 describes how, in

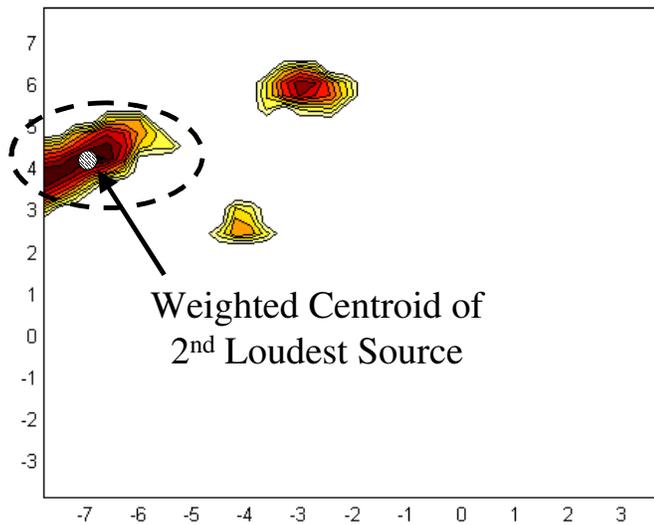
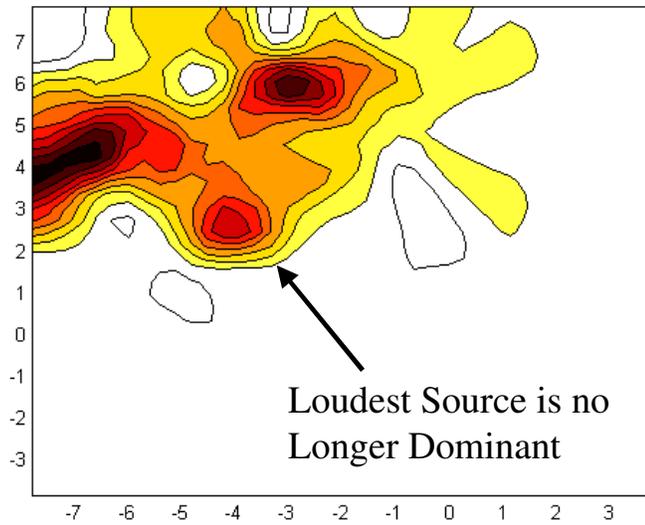


Figure 4.10. Stepping through the iterative clustering process, second round. (Top) Original auditory evidence grid with one dominant peak. (Bottom) Thresholded result with largest cluster circled.

pseudocode, source information can be incorporated into the auditory evidence grid creation algorithm.

- **Step 6 - Repeat steps 1-5 until all sources have been localized**

Knowing when to stop this iterative process is the difficult part. There are a number of properties that can be used to predict the end of this iterative process:

1. The largest remaining cluster belongs to a source that has already been detected and removed during this iterative process.
2. The largest cluster remaining has a very large variance since it was formed from samples dominated by reverberant sound. The variance is defined in Equation 4.11:

$$V = \sum_i W_i (x_i - \mu_i)^2 \quad \text{Equation 4.11}$$

3. where, for all locations ( $i$ ) included in the cluster, ( $W_i$ ) is the log odds probability predicted by the auditory evidence grid for that location, ( $x_i$ ) is the centroid for that cell in the auditory evidence grid, and ( $\mu_i$ ) is the centroid of the detected cluster.
4. The numbers of samples pointing at the largest remaining cluster is very small, since the peak was formed from samples pointing in arbitrary directions.

None of these properties are singularly perfect at identifying when no more sources are remaining in the environment to be localized. Together, however, they can usually detect the end of the iterative process.

Therefore, each of the previous steps should be repeated until either no source is localized during step 3, the largest remaining source is already known or has very large variance, or the number of remaining measurements used in step 5 is too small. For this work, a minimum of 20 samples was required for mapping. Figure 4.11 demonstrates a third source detected by iteration (Top), and a final evidence grid on which the iteration stops due to large variance (Bottom).

This iterative approach to localizing stationary sound sources worked well when applied to data collected from the first phase of testing. Except for the test localizing air vents, this iterative clustering approach could extract all of the sound sources from maps created using either of the sampling strategies. Furthermore, the accuracy appeared high for most extracted sound sources, generally within 0.3-m (note that the source position was only recorded to within 0.3-m accuracy). The one source that was not localized well was a human speaker in the presence of other noise sources. The iterative approach still found a separate cluster for a source within the vicinity of the speaker, but possibly due to movement by the source or the scarcity of samples primarily indicating the human target, at least one localized source was off by 1-m. The next sub-section will focus more on this question of accuracy.

Ultimately, what should be gathered from these results and the results of the next section is that, in the short-term, this iterative approach to localizing stationary sound sources will prove highly useful to an acoustically-aware robot. It allows a robot to localize stationary sound sources in the environment, filling a critical gap in the mathematical framework for sound propagation discussed in Chapter 3. As such, we will

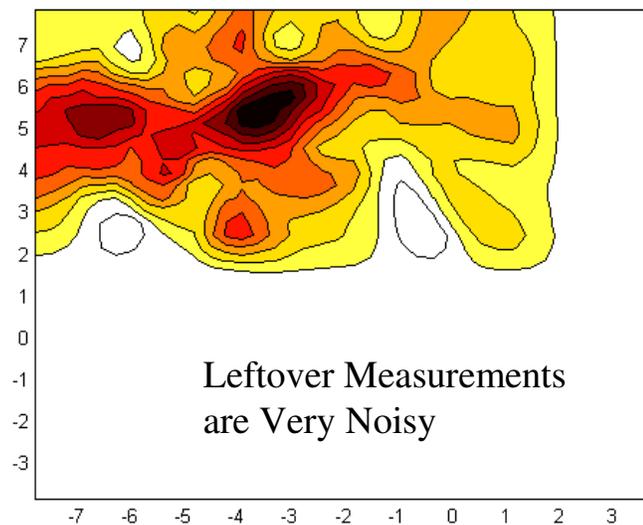
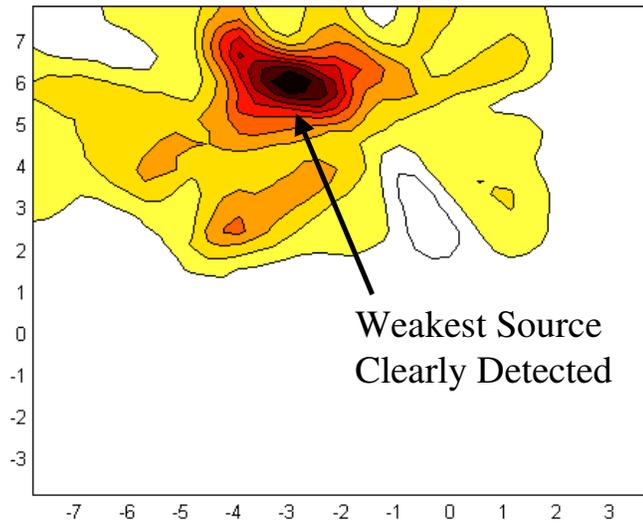


Figure 4.11. Stepping through the iterative clustering process, conclusion. (Top) Third evidence grid, focusing on single source. (Bottom) Final evidence grid on which algorithm stops, because it is very noisy and created from only 8 samples. Notice that the level of noise in the measurements makes separating the sources very difficult.

continue to use auditory evidence grids, and the accompanying iterative clustering approach, for the remainder of this dissertation. However, the results also reveal that there remains work to be done in this area. For one, the current approach has problems in localizing large numbers of sources in the environment. This could be due to a poor sensor model choice, so a better sensor model that incorporates models of sound propagation or environmental effects may actually the need for this iterative heuristic. Alternatively, the use of greedy exploration techniques designed to flush out an evidence grid [Thrun et al. 2005] in combination with short-duration auditory evidence grid creation may serve the same purpose.

A second problem that was highlighted by the tracking of human speakers was localizing sources in the presence of moving sound sources. Although it is explicitly stated that this technique is designed to localize stationary sound sources, moving sound sources are likely to be present in many of the environments. If the source is always moving around, then the source should not be localized, or interfere more than any other type of noise with the localization of other sound sources. If, however, the source remains still for some period of time, and then moves again, like humans or robots in the environment, then that source is likely to have a larger impact on source localization. Not only will it appear as a source in the evidence grid, but it could also mask other sources that a robot is trying to localize. Given that knowledge of moving sound sources is of general interest to a mobile robot anyways, future work in mapping sound sources in the environment needs to incorporate models of moving sound sources into the representation.

## Testing Phase 2 – Determining Accuracy

Where the first phase of testing focused on identifying requisite hardware and control strategies, the second phase of testing focused on determining the accuracy of the resulting estimates, when collected by an autonomous robot. While a human may sometimes be available for guiding a robot, the localization of sound sources should not require human assistance. Therefore, given an autonomous robot, what are some different control strategies for acquiring localization information, and how does the accuracy vary between strategies?

The two strategies tested here for autonomous localization of sound sources are a waypoint path and an area-coverage heuristic, each having a different goal in the localization of sound source. The purpose of the waypoint-path is to quickly cover a large area, identifying potential sound source locations in the environment. The purpose of the second algorithm, an area-coverage heuristic, is to spend more time in the vicinity of the source, verify that a source is present, and more accurately identify source properties. Combined, the hope is that the two robotic movement strategies will present a common strategy for localizing unknown sources in an arbitrary environment. The robot follows a waypoint path, effectively patrolling the environment, until something is detected, at which time it changes to an area-coverage heuristic focused on investigating the potential sound source identified during while patrolling the area.

The data used for this phase of testing was originally reported on in [Martinson and Schultz 2007]. Since the original publication, this section has been updated to reflect recent modifications to the algorithm.

### Experimental Setup

In this second stage of testing, the same B21r robot used in the first stage was again used in the AI Center Laboratory at the U.S. Naval Research Laboratory (NRL). The layout of obstacles was slightly different from the first stage, however, so as to allow the robot access to multiple sides of the sound source. Figure 4.12 illustrates this modified layout.

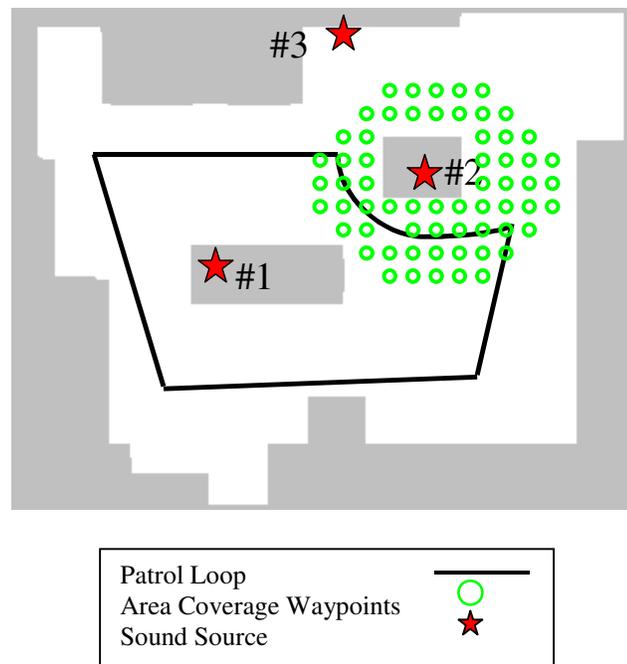


Figure 4.12. An overlay of the NRL environment, showing an example waypoint path, set of area coverage target points, and source locations.

During this second phase of testing, the B21r was used to localize each of three pc-speakers with unknown  $\{x,y,\theta\}$  5 times, for a total of 15 trials. Figure 4.12 illustrates their respective positions in the environment. During any one test, only one speaker was

playing and all speakers played the same nature sounds track (rain) at a 65dB volume. For each test, the robot first followed a waypoint path, quickly estimating the location of the sound source in the environment. Then it would dynamically choose where to center and execute its area-coverage heuristic based on the waypoint-path localization results.

### *Waypoint Path*

The first type of autonomous movement is the waypoint-path, described by a set of ordered waypoints in the environment for the robot to visit (see Figure 4.12). The purpose of this phase is to expose the robot to as much of its environment as possible so that it will be able to detect any significant ambient noise sources.

Provided with a waypoint path, the robot uses a path-planner (Trulla, [Hughes et al. 1992]) to guide it from its current position to each waypoint in turn while dynamically avoiding obstacles. Upon arriving within some threshold distance (0.4-m) of the desired waypoint, the robot selects the next waypoint in the specified order as a target, and the cycle repeats. To account for inconsistencies between the real world and the map, a timeout mechanism monitors the robot progress and forces it to move on to the next waypoint after 3 minutes. The task is finished when the robot has successfully visited or tried to visit all specified waypoints. After completing one loop through the environment, the robot then processes its auditory data using the auditory evidence grid and iterative clustering process to search for likely source position candidates.

The expected goal of this type of autonomous movement is the quick localization of possible sound sources. As such, the robot sampled while moving through the environment to reduce time, resulting in an increased localization error. After quickly

patrolling the environment (collecting an average of 40 samples per run) the robot was still able to identify an approximate location for the source in every trial (Table 4.2).

Table 4.2. Mean localization error when auditory evidence grids are used with data collected by a moving robot.

	Localization Error (m)
Source 1	1.0
Source 2	0.93
Source 3	1.5
Combined	1.1

Unfortunately, however, this quick patrol strategy produced a relatively high error over each of these trials, even though it was localizing only a single source. This error was due to the very limited number of samples recorded in the vicinity of each sound source. Usually, less than half of the samples were even within the 3-m range over which the spatial likelihoods were calculating, and even then the distance to the source varied significantly. Source 3 was particularly poorly detected for this latter reason, as the robot never came closer than 1.7-m to a source situated on a bookshelf next to the wall.

Given the high error demonstrated during these tests, it is clear that this strategy of using a moving robot to quickly localize the source is not going to produce accurate enough results for estimating sound flow. However, what this patrol strategy did do was correctly localize something in the vicinity of the source. Therefore, to overcome the error introduced by this scarcity of data, a second type of robotic movement is needed.

Either the robot needs to travel slower, and pause and sample, along the waypoint path, or, when the application has enough time, the robot can use the coordinates provided by the quick pass through the environment to thoroughly investigate the source and improve accuracy. This second type of investigatory movement is explored in the next section.

### *Area-Coverage*

The second type of autonomous control is a directed investigation of each source using an area-coverage heuristic. As it is impractical to investigate the entire lab space, this type of control requires an initial target around which to center the area-coverage activities. For this purpose, we used the results of the waypoint-path phase previously described. Therefore, once a coordinate was identified, the robot would move to that area and begin the area-coverage task. The goal of this control strategy was to acquire as many samples pertaining to this sound source as possible, so as to ideally improve on the localization result provided by following a waypoint-path.

Provided with a target set of sound source coordinates to investigate, a set of unobstructed locations is identified within a 3.5-m radius of the target using the obstacle map of the environment. A smaller radius could also be used when identifying unobstructed locations, but the goal of this phase was to identify the location as accurately as possible, so a large area was selected for investigation. These unobstructed locations become waypoints for the robot to visit, effectively performing an area coverage task in the vicinity of the suspected sound source. Unlike the waypoint task, however, visiting these waypoints does not need to be done in any particular order, and so the robot will always travel to the nearest waypoint. The circles in Figure 4-10 show a

set of waypoints to be used for investigating a single source. Also unlike the waypoint task, the robot will stop at each target to collect samples. Movement during sampling introduces additional error to the estimate (as discussed during phase 1 of testing), so stopping the robot should improve accuracy at the expense of time. Appendix C.3 describes the area-coverage heuristic in more detail.

After completing the investigation of a single source, the robot now has enough data to refine the position of the source using iterative clustering. The accuracy of the entire process, from waypoint path to investigated source, is given in Table 4.3 for each of the three sound sources:

Table 4.3. Mean localization and orientation error as produced by the discovery process.

	Localization Error (m)	Standard Deviation (m)
Source 1	0.32	0.26
Source 2	0.13	0.20
Source 3	0.19	0.22
Combined	0.21	0.23

Even though the initial investigation coordinates may have had a high error due to the quick nature of the waypoint-path estimation, the resulting estimates after performing an area-coverage task in the vicinity have relatively low error, and are ideal for building estimates of the direct or reverberant fields. Figure 4.13 demonstrates an example evidence grid created from a directed investigation of the source using area

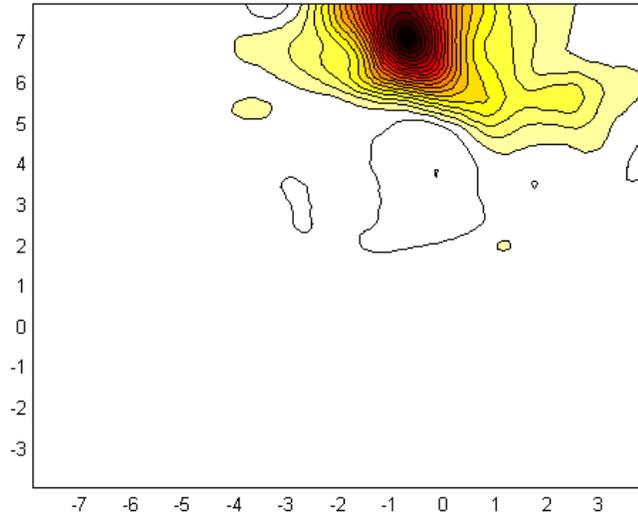


Figure 4.13. Auditory evidence grid created from 137 samples collected during a directed investigation of a source using an area coverage heuristic.

coverage. Notice the smooth, singular nature of the peak, which makes it easy to extract using nearest-neighbor clustering.

#### 4.2.2 SOUND SOURCE VOLUME AND DIRECTIVITY

After determining the position of the sound source, the next step is to determine the volume of (i.e. sound pressure level generated by) the sound source so that we can estimate the effects of the direct field. If there is little time or need for an accurate model of the source, then the simplest source model is an ideal omni-directional source. Using the average sound pressure level of a few samples taken at known distances from the source, a relative volume can be quickly established and the source inserted into our models of the auditory scene. This simplistic model of volume, however, is far from being correct. Most sources, usually due to their physical shape, are not omni-directional, meaning that the direct sound produced by that source varies in volume

depending upon the angle to the source. Therefore, without a model of directivity to estimate how the volume changes with direction, averaging a few samples collected at random locations around the source will not produce a good estimate of the source volume.

The challenge in building a model of source directivity and volume is the difference between the ideal method for constructing such a model and the actual nature of the data from which to construct it. In the ideal method for determining source directivity, the sound source would be located in an anechoic chamber where the magnitude of any reflections is negligible, and the sound could be measured at a constant distance from the source. With the robot, however, we are in a real environment where there is a substantial reverberant component to measured sound. Furthermore, due to the presence of obstacles in the environment, the collection of data gathered comes from an arbitrary set of distances and angles to the source. How do we overcome these differences? The solution, originally described in [Martinson and Schultz 2007], is to work backwards from the sound source propagation model discussed in Chapter 3.

#### Determining Directivity - Algorithm

When the robot records a sample of the auditory scene, it is actually measuring some energy from the direct field of each sound source in the room, plus some amount of reverberant energy and some amount of transmitted energy. Equation 3.2 described this sum in terms of field effects. Equation 4.12 re-writes this equation in terms of pressure:

$$p_s^2 = \sum_i (p_{direct_i,s}^2) + p_{reverb,s}^2 + p_{trans,s}^2 \quad \text{Equation 4.12}$$

Where  $p_s$  is the rms pressure of the sample ( $s$ ),  $p_{direct,i,s}$  is the rms pressure due to un-reflected sound from source  $i$ ,  $p_{reverb,s}$  is the rms pressure due to reflected sound waves, and  $p_{trans,s}$  is the rms pressure due to transmitted sound. The loudness of the direct sound for one particular source is the quantity we are the most interested in, so we will be limiting the set of samples used to those in close proximity to the source being modeled. As a simplifying assumption, we will, for now, assume that active sources are not located close together, therefore we can ignore the effects of their direct field on samples taken near another source. We will also assume that transmitted sound is negligible, leaving only the direct field component for a single source and the reverberant field:

$$p_s^2 = p_{direct,s}^2 + p_{reverb,s}^2 \quad \text{Equation 4.13}$$

To remove any more components from the equation is impractical. The direct field component is what we need for estimating directivity, while the reverberant component will almost always be too large to ignore when estimating the volume. As is, however, this equation still has too many unknowns. To identify the separate components, we need additional equations provided by 2 simplifying assumptions. The first such assumption is that the loudness due to reverberant sound will remain constant over the entire room. Since reverberant sound describes the contribution of reflected sound waves, and sound waves will reflect many times all over the room before either decaying to nothing or reaching a receiver, this is a good first approximation often used for estimating the reverberant field [Raichel 2000].

The second simplifying assumption involves the contribution of the direct field. As the direct field describes the volume of un-reflected sound emanating from the source,

conservation of energy ( $p_{rms}^2$ ) tells us that the energy of the direct field should decay linearly with square of the distance. So the farther away the robot is from the source, the greater the energy coming from reverberant sound and the less from direct sound. Equation 4.14 illustrates how to estimate the original volume of the source at a distance  $d_0$  given an actual distance  $d_s$ .

$$d_0^2 p_{direct,d_0}^2 = d_s^2 p_{direct,s}^2 \quad \text{Equation 4.14}$$

Therefore, we can also assume that after some distance the contribution due to the direct field is minimal, and then estimate  $SPL_{reverb}$  as the mean volume of the sampled data taken beyond  $d_r$  meters from the source. In this work, we used two meters as a good approximation, where the volume of the direct field will have dropped 6dB from the volume at 1-m from the source.

Now that we have estimated the contribution of the direct field, the final step is to combine all of the data collected from arbitrary distances and angles into a single model estimating volume for any specified distance and angle. For this purpose, we first use Equation 4.14 to calculate  $p_{direct,d_0}$  at the specified distance  $d_0$ , and then we apply a Gaussian smoothing function centered on the desired angle ( $\omega$ ). The final equation for the model of directivity is:

$$p_{direct}^2(d_0, \omega) = \frac{\sum_s e^{-(\theta_s - \omega)^2 / 2\sigma^2} d_s^2 p_{direct,s}^2}{d_0^2 \sum_s e^{-(\theta_s - \omega)^2 / 2\sigma^2}} \quad \text{Equation 4.15}$$

Where  $(d_s, \theta_s)$  is the position of the sample relative to the center of the source, and  $\sigma$  is the standard deviation of the applied Gaussian. Figure 4.14 demonstrates a

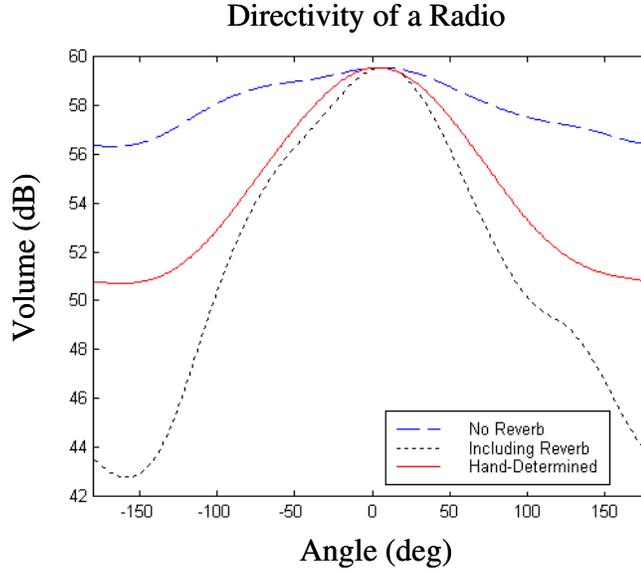


Figure 4.14. Comparison of robot-created directivity models using different reverberation assumptions, with a hand-measured directivity model. The three models assumed a maximum volume of 59.5-dB, as measured by the sound pressure level meter.

directivity model for a pc-speaker at a distance of 1-m. Appendix B.3 presents the pseudocode for creating this directivity model from sampled data.

Given this directivity model, the volume of the source is angle dependant, i.e. the volume is determined directly from the directivity equation using the known angle to the source. Furthermore, this same model can account for changes in volume of the source over time. If the robot returns to the area some time later, it can re-measure some small area (preferably away from the walls), and use that new volume ( $V_m$ ), measured at a known position ( $d_m$ ) and angle ( $\omega_m$ ), to add a constant multiplicative offset to the directivity model to reflect the change in the sound source:

$$p_{direct}^2(d_0, \omega) = p_{direct,0}^2(d_0, \omega) \frac{V_m}{p_{direct,0}^2(d_m, \omega_m)} \quad \text{Equation 4.16}$$

In the same manner, this representation of directivity can also handle differences between measuring equipment. Sensitivity differences between microphones is handled similarly. If two robots measure the same source using different microphone arrays, then their results can be compared by using a similar offset and the difference in estimated volume between the two microphones.

In Figure 4.14 that constant offset is used to compare three different directivity models of the same radio at a 1-m distance. The solid line is created from hand-measured data using a directional sound pressure level meter (Type II accuracy). The dotted line uses robot-measured data with Equation 4-15 and all of the stated assumptions. Finally, the dashed line uses the same robot measured data and equation, but assumes that the reverberant component is negligible (i.e.  $p_{reverb} = 0$ ). What this figure demonstrates is the effect of the reverberant field assumption on our resulting model. Each of the directivity models demonstrate the same cardioid centered at  $0^\circ$ , which should be the result for a radio speaker. However, using the hand-measured model as the ground-truth, assuming a negligible reverberant field underestimated the difference between peak volume and minimum volume. In contrast, assuming a constant reverberant field overestimated the difference in volume.

In practice, the actual difference between these two assumptions and the ground truth is likely to depend heavily on the type of environment. For example, a smaller environment with similar materials may better fit the constant reverberant field assumption. In the future, improving this model should probably incorporate more information about the nature of the reverberant field. As will be demonstrated in Chapter

5, however, the constant reverberant field assumption is good enough for many applications.

### Identifying Source Orientation

Given the variation in the resulting directivity models due to environmental effects, experimental testing of the source modeling process was focused on the correct identification of source orientation. If a robot can identify the direction of maximum volume (i.e. the source orientation), then avoiding or maximizing the effects of the sound source is possible, regardless of the noise in the rest of the directivity model. As such, testing this source orientation detection was divided into three stages. In the first stage, we tested the accuracy at which source orientation was estimated by using a mobile to investigate one source with known  $\{x,y\}$  and unknown  $\theta$ . In the second stage, we tested the autonomous localization and modeling process for a single source of unknown  $\{x,y,\theta\}$ , using the same data from the phase 2 testing in Section 4.2.1. Finally, in the third stage, we tested the ability of the robot to localize, and identify source orientation of multiple simultaneously operating sources of unknown  $\{x,y,\theta\}$ .

The first two stages use data collected by the B21r for localizing sources, while the third stage uses the Pioneer2-dxe robot in a different environment, so as demonstrate generality. During each of these three stages, we applied a 10<sup>th</sup> order highpass FIR filter (300-Hz cutoff frequency) to every sample before analyzing the data. Since the ambient noise sources being measured had significant high frequency components, the filter had little effect on the auditory evidence grid creation. What the filter did do, however, is reduce the impact of robot motor noise on determining directivity. Since the robot's

motor was in close proximity to the microphone array, it could overpower the weak volumes measured farther away from the source.

### *Stage 1 - Known $\{x,y\}$ , Unknown $\theta$*

In this first stage of testing, a single source of known centroid position was rotated through 7 different angles in 45° increments. Provided with the ground truth source location, the B21r was used to investigate the source once for each different angle using just the area-coverage algorithm with a 3.5-m range. One angle was not tested due to the source pointing at a solid wall where the robot could not investigate. The sound source used in this stage was a pc-speaker playing nature sounds (rain) measured as being 65 dB at 1-m from the source (including both direct and reverberant sound).

Over 7 trials, the mean error for estimated source orientation was 0.2-rad of ground truth with a maximum error of 0.5-rad. Given that the source itself is a pc-speaker with a wide frontal lobe, this approximation should be adequate to guide the robot away from the loudest areas surrounding the source.

### *Stage 2 – Unknown $\{x,y,\theta\}$*

In the second stage testing, the B21r was used to localize each of three pc-speakers with unknown  $\{x,y,\theta\}$  5 times, for a total of 15 tests. During any one test, only one speaker was playing. All speakers played the same nature sounds track (rain) at a 65-dB volume. For each test, the robot first moved along the same patrol route, localizing the active source. Then it would dynamically choose where to center its investigation using area-coverage. After sampling the area, the sound source orientation was estimated

along with the original location. This same data was used reported earlier for estimating source location coordinates using an auditory evidence grid. The layout of the room and the source positions can be seen in Figure 4.12. Table 4.4 shows the mean error in source orientation, for each source:

Table 4.4. Mean error in identifying the direction of maximum volume, as produced by an area coverage task.

	Mean Orientation Error (rad)	Standard Dev. Orientation Error (rad)
Source 1	0.22	0.23
Source 2	0.18	0.08
Source 3	0.32	0.23
Combined	0.24	0.17

These results demonstrate the reliability of the discovery process in accurately finding and modeling sources. Sources 1 and 2 were located in areas where the robot could completely encircle the source, and therefore gather data from all directions. Source 3, however, was against a wall, so the robot was limited to gathering data in the 180° foreground. Due to this limited area, as well as the proximity to the wall and its echoic effects, the orientation error is highest for this third source.

### *Stage 3 – Multiple Sources*

The final stage of robotic testing demonstrated the ability of the robot to detect multiple simultaneously operating sources and identify their characteristics. Two

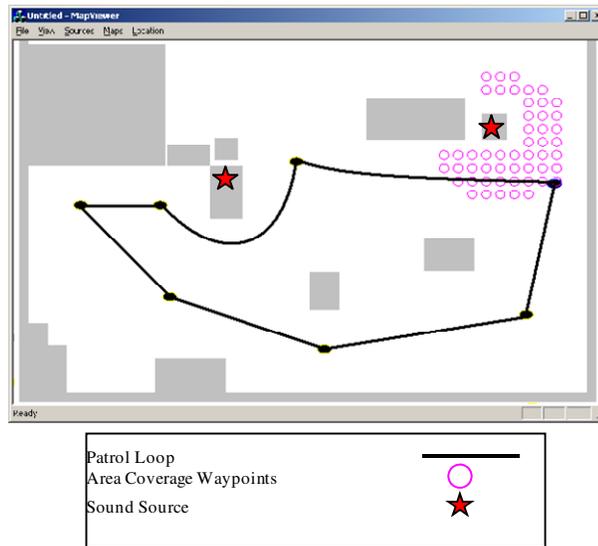


Figure 4.15. Hand coded obstacle map used by the pioneer for navigation in an environment with two sources. The black line shows the waypoint path followed during a patrol, while the circles illustrate a set of targets reached by the robot to complete an area coverage task for a single detected source.

sources, an air filter (0.5-m x 0.3-m x 0.3-m) and a two-speaker radio generating static noise, were placed 5.8-m from each other. Figure 4.15 shows their relative placement. The pioneer2-dxe robot (Figure 4.2) was then used to localize and model each source. Following the initial patrol phase, the robot identified two potential clusters, corresponding to each of the two sources. Both initial clusters were within 1-m of the actual source location. Upon further investigation, the robot improved the localization accuracy for the air filter to within 0.2-m, and to 0.4-m for the radio. The orientation accuracy for each source was 0.64-radians and 0.4-radians respectively.

Table 4.5. Localization and orientation accuracy of the two source discovery process

	Localization Accuracy (m)	Orientaion Accuracy (rad)
Filter	0.2	0.64
Radio	0.4	0.4

### 4.2.3 USING SOURCE INFORMATION WITH THE SOUND FIELDS FRAMEWORK

So far in this section, we identified three different properties of sound sources that could be used in conjunction with the sound fields framework to estimate the flow of sound through an environment: source location, average volume, and directivity.

The position of the sound source was estimated using an auditory evidence grid in conjunction with an iterative nearest neighbor clustering algorithm. By itself, however, the source location does not contribute much to the sound flow estimation problem. In Figure 4.16 (Top), the shape of the direct field is plotted for some distance around a detected sound source of unknown volume. The displayed map assumes that the receiver can reach any location around the source (i.e. the geometric layout is unknown), and that the source is omni-directional. These estimates of sound flow without knowledge of the volume can certainly be used to guide a robot to or away from a sound source, but they are not very realistic. Without volume, there is little way to compare sources (unless they are assumed equally loud), or estimate how much noise a robot might be exposed to as it moves through the environment.

Volume, however, is not necessarily difficult to estimate for a source that does not vary over time. As discussed in Section 4.2.2, a simple estimate for volume is to collect

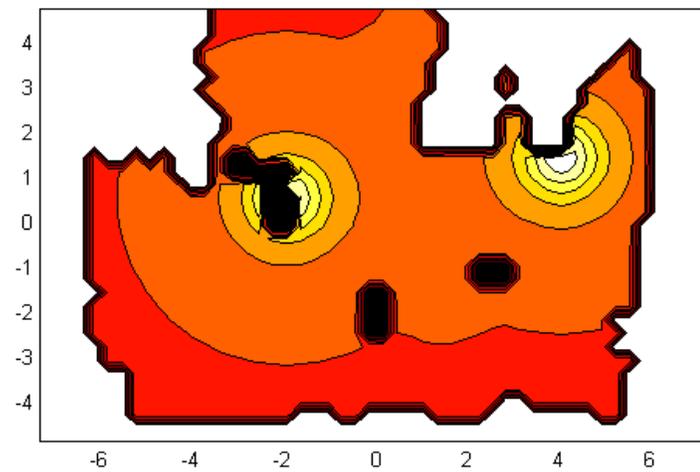
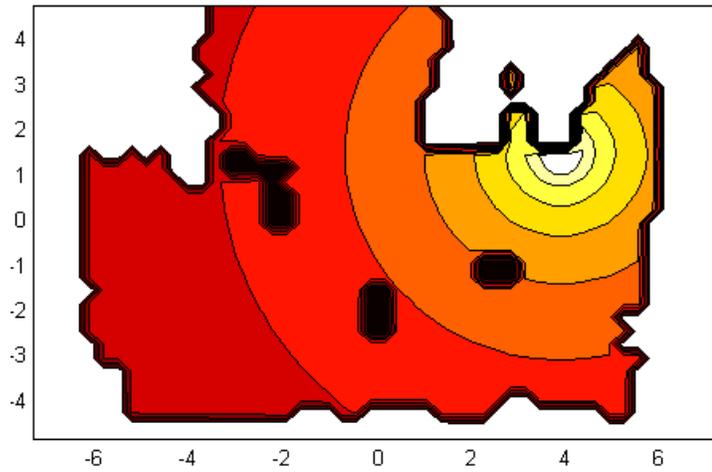


Figure 4.16. Direct field estimates created from a single source of arbitrary volume (Top) and two sources of known volume (Bottom). The obstacles in these field estimates are there for scale purposes, and were not used to predicting occlusions or reverberant effects.

some number of samples in the vicinity of the source and average the result to estimate volume. Using this simple method for estimating volume, and still assuming omnidirectional sources, we can estimate the combined volume of two detected sound sources over an arbitrary sized environment (Figure 4.16, Bottom). Still estimating only the direct field, the robot can predict that the lowest sound between the sources is not actually in the middle, but rather closer to the quieter fan source on the left.

To increase the accuracy of the sound fields estimate even further, the last piece of information discussed, which the robot can gather, is the directionality of the sound source. Although we can estimate the volume very simply for each source, the perceived volume should actually depend on the angle from the receiver to the sound source. Given time to investigate a sound source and collect enough samples from a variety of angles, a robot can build a model of directionality, predicting the volume detected at each angle to the source. Figure 4.17-19 demonstrates how knowing the directivity of each of the sound sources can be used to build directional fields for each source, and then combined together into a representation of overall sound flow due to direct sound. Appendix B.5 describes the pseudocode implementation of the algorithm used to construct each of these direct field models.

In general, what this information about sound sources allows a robot to do is make some predictions about the shape of the acoustic environment in the area where it may need to travel. For instance, the robot can use this modeling ability to predict the regions of loudest environmental noise, so as to either avoid or move into areas of loudest sound. These two applications will be demonstrated in Chapters 5 (Improving the Signal-

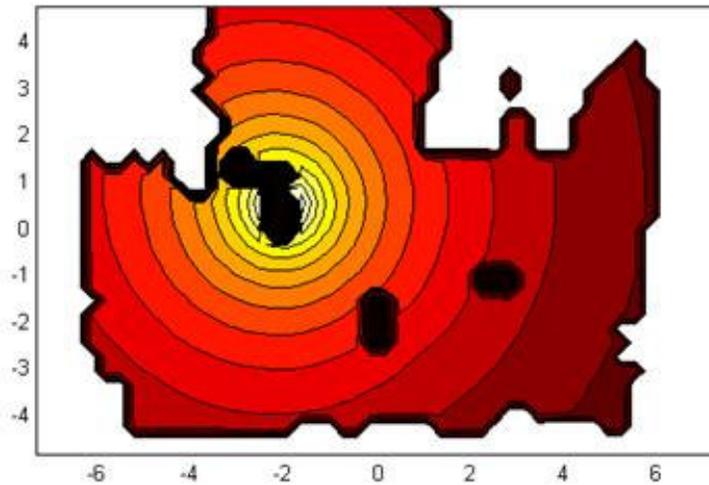
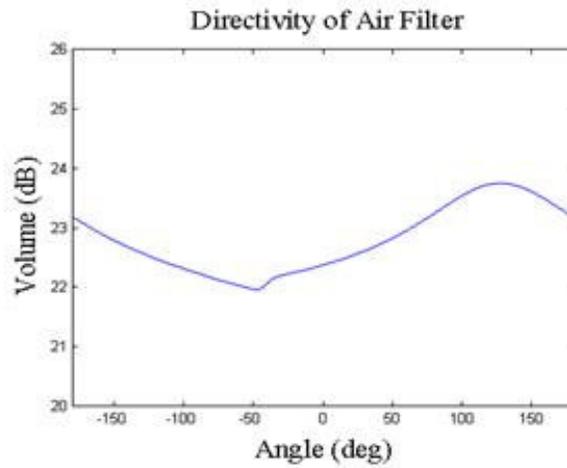


Figure 4.17. Process of creating sound propagation models from sampled area coverage data, part 1. (Top) Directivity results for the mostly omni-directional air filter. (Bottom) Direct field for the air filter, showing the spherical spreading common to mostly omni-directional sources

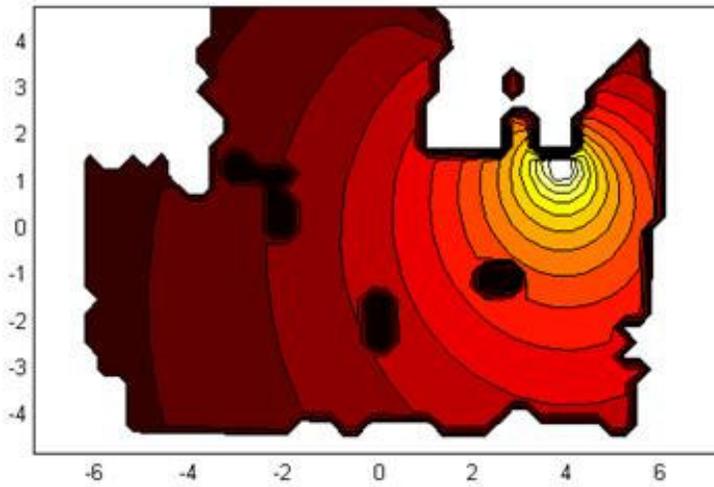
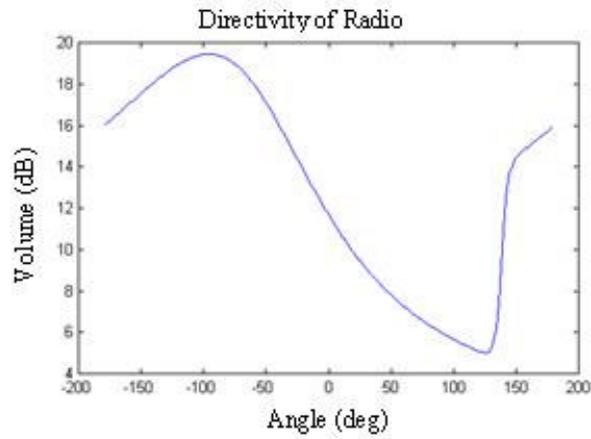


Figure 4.18. Process of creating sound propagation models from sampled area coverage data, part 2. (Top) Directivity results for the radio. (Bottom) Direct field for the radio, showing a distinctly louder region in front of the radio.

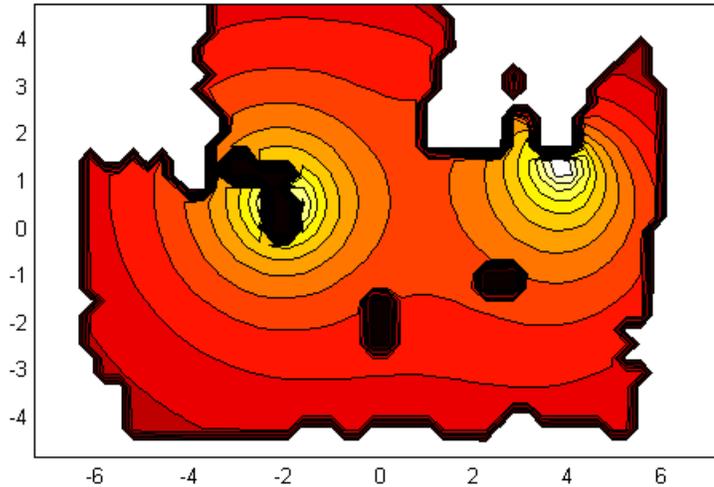


Figure 4.19. Process of creating sound propagation models from sampled area coverage data, part 3 - combined direct field for both sources

To-Noise Ratio) and 6 (Stealthy Approach). Another possible use of this information is predicting which sound source will be loudest in a given area. By knowing what source dominates each region of the environment, a robot can detect and track changes to the auditory scene over time (Acoustic Monitoring – Chapter 5). This information that the robot can gather about active sound sources in the environment serves as the basis for many applications. Although receivers and environments are important for refining accuracy, knowledge about the sound sources is critical for virtually all applications reacting to or making use of sound levels in the environment.

#### 4.2.4 IDENTIFYING AND REPRESENTING SOUND FUNCTIONS

The final piece of information about the sound source that we will be using in this dissertation is the sound function. In general, the sound function refers to the sound being generated by a sound source at any given time, including the volume of the sound

source. For example, the sound function of a radio playing a single song is the recording of that music, modified by the volume at which it is being played, the properties of the amplifiers/speakers playing the song, and the time at which it is started. Therefore, for any non-repeating source, knowing the sound function exactly would require knowing an infinite stream of data. If the exact output of the sound source is not needed, however, then the sound function can be reduced down to a more compact representation depending upon the application.

In this dissertation, we will use the source sound function for classification. Assuming that our robot has collected some number of samples from the environment, our goal is that the robot should be able to use its representation of the sound function to determine which source was loudest in each of the collected samples. Furthermore, we would like to use the representation of the sound function to separate out any new sources from known sources in the environment, and determine whether or not the sound source is on or off. In short, we would like to use the sound source function representation for classifying samples as belonging to, or most importantly, not belonging to any particular source. The representation that has been used successfully by a number of others for this purpose in audio classification is mel-frequency cepstral coefficients or MFCCs [Slaney 1994; Quatiri 2002].

### The MFCC Implementation

MFCCs are a classification feature set based on the mel-scale filter bank, a filter bank designed to group frequencies in a manner similar to human perception. In this filter bank, low frequencies are traditionally grouped equal spaced bands, while higher

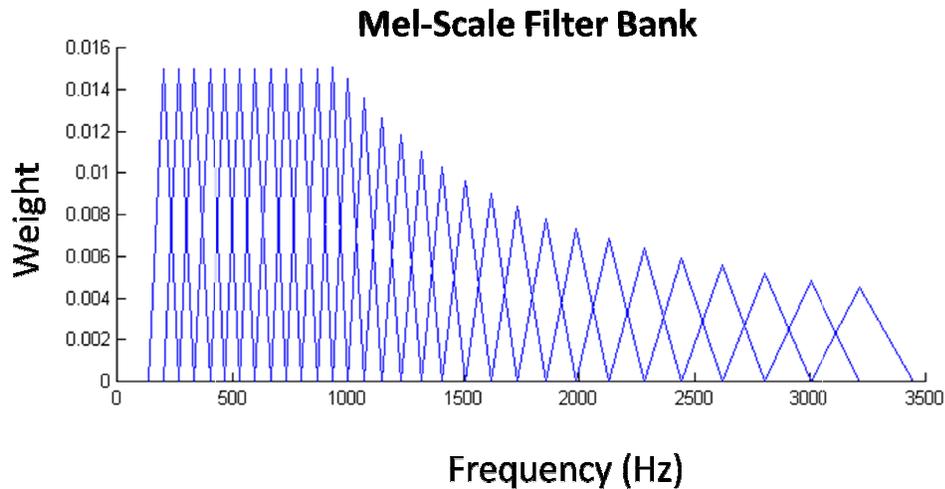


Figure 4.20. Weight vs frequency plot of a mel-scale filter bank.

frequencies are grouped in bands that increase logarithmically in size with the frequency. Figure 4.20 demonstrates the triangular weighting in the frequency band characteristic of a mel-scale filter bank. MFCCs are then calculated by taking the discrete cosine transform of the energy of these mel-scale frequency bands. A good description of how to calculate MFCCs can be found in [Quatiri 2002].

Although the mathematics behind MFCCs is generally the same across all implementations, the specific details of frame size, window size, pre-filtering algorithm, etc. vary from implementation to implementation [Zheng et al. 2001]. In this work, the code for calculating the coefficients was derived from the Auditory Toolbox implementation [Slaney 1994], which can be downloaded from Malcolm Slaney's website<sup>6</sup>.

---

<sup>6</sup> <http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/>, Accessed Aug 25, 2007

- Frame Size – 10-msec
- Window Size – 31-msec (length of 256 at 8192-Hz)
- Pre-Filtering – a hamming window

What these numbers mean is that a single 250-msec sample collected by the robot is run through an iterative algorithm which, for every 10-msec (the frame size):

- Collects the next 31-msec (the window size)
- Applies a Hamming window to the samples
- Calculates the first 8 MFCCs for these samples.

Therefore, for a single 250-msec sample, we get 21 sets of 8 coefficients. We then discard the first coefficient of every set, because it is generally unreliable for classification [Zheng et al. 2001]. Our classification feature set is then the mean and variance of these 7 MFCC's over that sample, for a total of 14 features. Appendix B.4 describes the creation of this classification feature set in more detail.

Typically, to achieve the best classification performance, a longer sample than 250-msec is desired, with 1-sec generally considered long enough to capture the wide range of variation in many typical sources. Our goal, however, was to use MFCC-based classification with the samples already collected by the robot when localizing sound sources in the environment. With a moving platform, even 500-msec samples can introduce a large amount of position error. Therefore, to improve the accuracy of the feature set, each MFCC feature vector is actually calculated over two successive 250-msec samples.

## Classification Algorithm

In this dissertation, we make use of nearest-neighbor classification [Duda et al. 2001] to identify sound functions with our MFCC feature set. This classification scheme has successfully been used with MFCCs and other auditory feature vector sets [Ravindran 2006] for distinguishing between a range of sound types including speech, animal noise, and music.

Before a robot can use nearest-neighbor classification to identify sound types, it first needs to build a set of class vectors. A class vector is essentially the mean feature vector for a given sound type that the robot is trying to classify. For instance, if the robot is trying to identify samples that contain noise generated by a particular air filter, it needs to know the average feature vector of samples generated by this air filter or class of air filters. This, in turn, requires that the robot have a set of samples known to predominantly contain noise from the air filter. Luckily for the robot, however, this information is already available.

In the previous sections, we described an area-coverage algorithm used to refine the source localization results and identify directivity and volume of the sound source. This area coverage algorithm involved collecting a large number of samples from many different angles and distances to the sound source. This same collection of sampled data can also be used to build the class vector for the same investigated source. Although some of the samples may be contaminated by other sources in the environment, most of the samples should have been collected within the region most strongly influenced by the target source, and, therefore, should be dominated by that source.

Using the samples collected from the investigation of the sound sources, the robot now has a class vector for every source it investigated. Now, whenever the robot records a new sample, that sample can be classified as belonging to any of the detected sources using the mahalanobis distance [Duda et al. 2001] from the new samples' feature vector to each of the class vectors. The class vector that is closest in distance is therefore closest in sound function to the recorded sample.

Assuming that each of the sound sources has a different sound function, this simple classification strategy works well for distinguishing between known classes. These classes could have been investigated by the robot, or provided *a priori* from hand-sampled data for particularly important sound types like speech. If the task of the robot, however, is to distinguish between known and unknown sounds, then this nearest-neighbor strategy has a problem. When the robot records a sample from a previously unknown source, that sample will still be matched with the nearest-neighbor class vector, which will still correspond to an existing source. To overcome this problem of being matched to existing classes, we inserted a set of 20 random classes into the feature vector set. Now, instead of matching to a known source, there is a good chance that a feature vector belonging to an unknown class will be closer in distance to one of the these random classes.

### Classification Results

To test the efficacy of the classification algorithm, we set up a small experiment involving two sources. An air filter with a significant directional component was placed to the right side of the room, generating noise at 50-dB. A second source, a small

fountain, was then placed roughly 3-m away at the top end of the room, generating water noise at a measured 54-dB. Figure 4.21 demonstrates the relative placement of these two sources within the environment.

To build a class vector for each of the two sources, a microphone was manually moved about each of the sources to simulate the collection of samples by a moving mobile robot performing an area-coverage algorithm over the surrounding 2-m. Both sources were enabled while the samples were being collected. These samples then served as the basis for a fountain class vector, and a filter class vector. An additional 20 random classes were also generated using the minimum and maximum range of the samples collected by the robot.

After the source functions were approximated with a class vector, additional sampling was performed at each of 10 sample locations distributed about the environment. Roughly the same number of samples were recorded at each location. Figure 4.21 shows the ratio of samples classified as belonging to the fountain vs. the filter at each sample location. Since the fountain was a significantly louder source, more than half of the sample locations detected more samples from the fountain than the filter.

Perhaps the most interesting part of the resulting ratios, however, was when a comparison was made with the predicted direct-field volume of each sound source. Given that the volume, directionality, and position of each source was constant and readily available, it was a simple matter to estimate the predicted volume of each source at each sample location in the environment. Comparing this ratio to the classification ratio revealed that, at all sample positions where the expected difference in

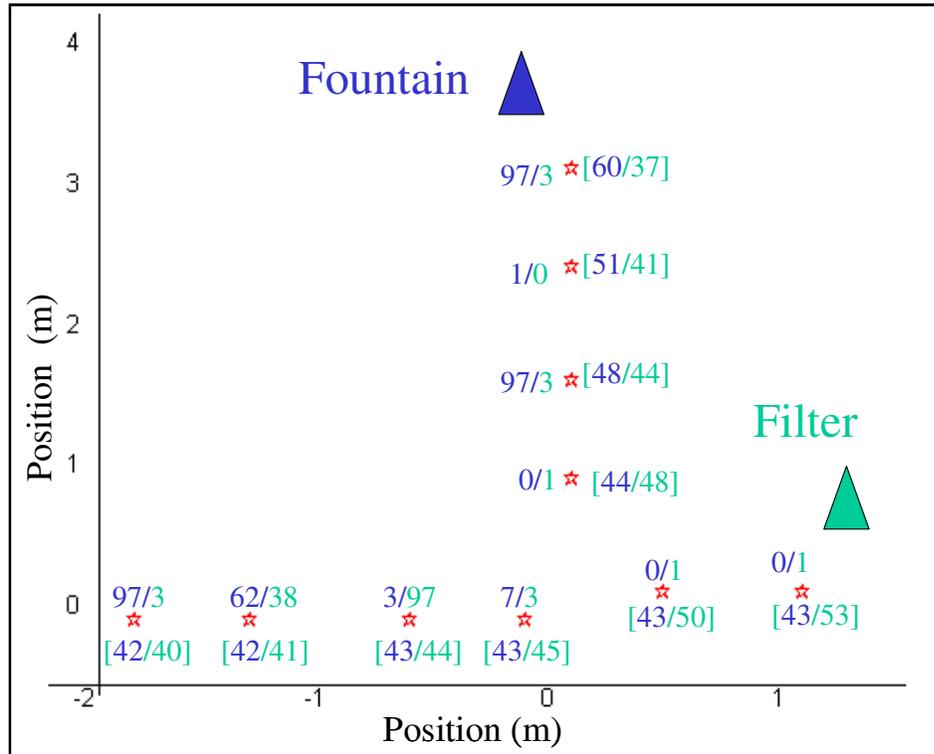


Figure 4.21. Classification results vs. predicted direct field volumes for two sources, a filter (green triangle) and a fountain (blue triangle), at regular intervals (red stars) around the room. On the top and left are the MFCC classification results, with the fountain in blue (nominator) and the filter in green (denominator). On the bottom and right (in brackets) are the predicted volumes of the direct field. At all sample positions where the expected difference in volume is greater than 2-dB, the louder source dominates the classification results. For volume differences less than 2-dB, the classification result is more variable.

volume is greater than 2-dB, the louder source dominated the classification results. For volume differences less than 2-dB, the classification result is more variable.

In general, this test suggests that the predicted volume of the direct-field seems to be related to the classification results. To verify these results, an additional round of testing was performed in the presence of three sources. A third source, a radio playing music in a range of roughly 56-62 dB, was added to the environment at position [-2.5,0]. This time, with the added variability of the music source, there were a lot more sample points with unpredictable classification ratios. Still, however, those sample points that were closest to each of the sources, overwhelmingly favored the nearest source.

These results suggest that MFCCs are a good choice for representing the sound function approximation of constant volume sound sources. When faced with a new sound source, a robot is capable of not only localizing the sound source, but also autonomously building a classification feature vector. Furthermore, using its knowledge of the surrounding acoustic environment, the robot can now also predict where to move to ascertain any changes to the sound function. After determining the loudest locations in the environment for this source, the robot can move to one of those locations, sample, and re-classify the results to potentially determine changes in volume, and/or sound function using MFCCs. This latter functionality will be demonstrated in Chapter 5.

#### **4.2.5 CHARACTERIZING SOUND SOURCES – SUMMARY**

In Chapter 3, a general sound source model was presented as being useful to an acoustically-aware robot (Section 3.2.1). For each sound source, the model contains knowledge about the position of the sound source in the room, its directivity, and its

sound function. What was not discussed in Chapter 3, however, was where the robot can acquire this information about each sound source. Section 4.2 has now addressed that concern for each part of the model, using a combination of algorithmic data fusion and robotic exploration.

The first part of the model, the location of the sound source, is determined using auditory evidence grids. By collecting samples over a wide variety of locations in the environment and fusing them together in an auditory evidence grid, a robot can identify the presence and general location of new sound sources in the room. Then the robot can improve on its localization accuracy for an individual sound source by investigating the area most likely to contain a source, collecting a large number of samples in the vicinity.

The second part of the model, directivity, is determined using the same investigative sampling technique that served to improve the localization accuracy of the auditory evidence grid. After collecting a large number of samples in the vicinity of the sound source, a robot can use its knowledge of spherical spreading to predict the volume of sound generated by the sound source for each angle. This information, combined with the position of the sound source, can then be integrated into the sound fields framework for estimating sound propagation through the environment. The resulting maps will be very useful in Chapters 5 and 6 for detecting changes to the environment, and avoiding or moving to loud areas of noise.

The third part of the sound source model that the robot can model is the sound function. Using the data collected during investigative sampling of the sound source, a robot can build a representation of the sound function using mel-frequency cepstral coefficients (MFCCs) that allow the robot to classify samples as being generated by a

particular sound source. This model of the source sound function will be very useful in Chapter 5, where a robot is trying to identify the presence of new sound sources, or changes in the environment. The model, however, currently lacks temporal information. As will be discussed in Chapter 6, the inclusion of such temporal information into the sound fields framework could be very useful to an acoustically-aware robot, and remains an important area of future work in developing the sound function aspects of the source model.

In summary, it is these tools for filling in information in the sound source model that will form the basis for all of the applied acoustical awareness work in Chapters 5-7. Knowing position and directivity, a robot can estimate how loud different parts of the room will be. Knowing the sound function, a robot can estimate what it will hear at each location. Either set of information allows a robot to make estimates about different parts of the room in which it is not currently located, providing navigational behaviors with the knowledge necessary to improve and acoustically aware application. The remainder of this chapter will now focus on improving those results even further by adding to the pool of knowledge that a robot can gather on its own.

### **4.3 PATH INFORMATION**

Once we have identified characteristics of sound sources in the environment, the next step is to utilize information about the path to enhance the accuracy of the direct field and build reverberant field estimates in place of the simplified, constant-field assumption used in the previous section. As discussed in Section 3.1.2, the attributes of the path entity that we are interested in acquiring for this purpose are the geometric

layout of the environment, the material properties of the surfaces, and any structural information for describing transmitted sound. Unfortunately, for the latter two categories there is little that a robot can do to acquire this information if it is not provided a priori.

Limited audio probing work (see Chapter 2) has had some small successes in identifying material properties, as have some computer vision efforts [Ragheb and Hancock 2003], but none of these work in the general case for characterizing an environment. The same is true of knowing the support structure. While DARPA is developing a sensor using RADAR to penetrate concrete and identify floorplans, material compounds, and enemy combatants[Miles 2006], the device is unlikely to be ready for robotic deployment in the near future.

Identifying something of the geometric layout of the environment, however, is well within the capabilities of a mobile robot. Section 4.1.1 described the creation of spatial evidence grids, or obstacle maps, by which a mobile robot can localize itself relative to obstacles in the area. These same spatial evidence grid representations can be used to describe the geometric layout of the surroundings.

#### **4.3.1 BUILDING REVERBERANT FIELD ESTIMATES FROM SPATIAL EVIDENCE GRIDS**

Figure 4.22 (Top) displays a spatial evidence grid used by the Pioneer2-dxe robot for localization in the Mobile Robot Lab environment. This evidence grid, built by the pmap utility [Howard 2004] using the measurements from a SICK Laser-Measurement-System mounted to the Pioneer robot, consists of a collection of estimates attached to specific locations, or grid cells, in the environment, where each estimate specifies the likelihood of an obstacle occupying this cell. These estimates range in value from 0-

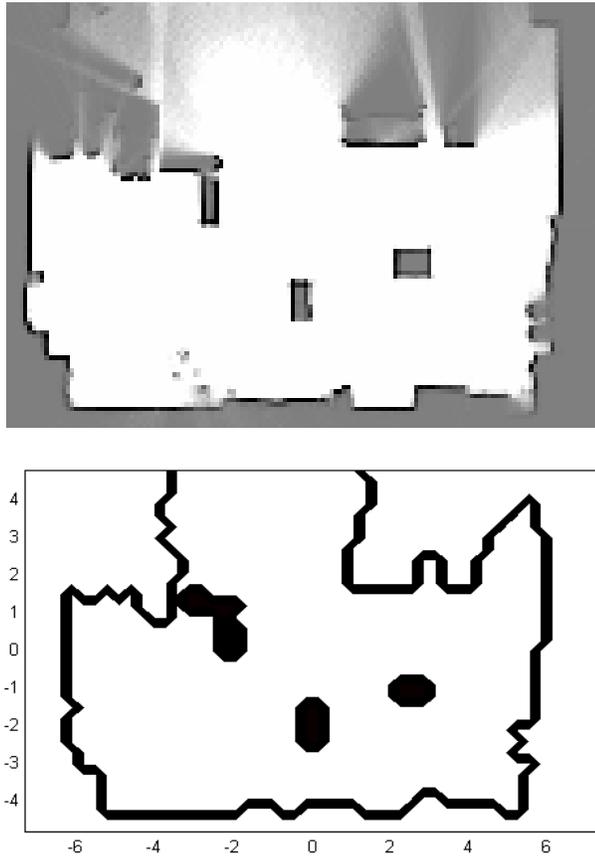


Figure 4.22. Comparison of a spatial evidence grid (Top) collected by the robot for localization purposes to a thresholded evidence grid (Bottom) used for reverberant field estimates

100%, and are not always consistent from neighbor to neighbor. To use this spatial evidence grid as a geometric layout in the mathematical framework, a few assumptions are needed about exactly where the obstacles are in the representation:

- **Which cells are occupied?**

An estimate of 0-100% cannot be easily used with reverberant field estimation models. A threshold is needed to determine, based on the evidence grid, whether or not any given cell is occupied or not. 75% is a typical threshold for determining occupancy. Figure 4.22 (Bottom) demonstrates the geometrical layout created by applying a threshold.

- **Where is the obstacle?**

If a grid cell was marked as containing an obstacle, then the obstacle boundary could be located anywhere within the area described by the grid cell. Unfortunately, spatial evidence grid cell sizes often range from 0.01-0.1m<sup>2</sup> in area, providing a large area in which the surface might occur. To simplify the representation for reverberant field estimates, each cell marked as containing an obstacle will be assumed to be completely filled. Therefore, the obstacle boundary is located at the edge of the grid cell.

- **What is the angle of the obstacle surface?**

Real surfaces in the environment are likely to occur at a variety of angles to the horizontal. By assuming that the obstacle fills the grid cell completely, however, all surfaces will now be restricted to either the vertical or horizontal edges of the grid cell. Given the large size of the average grid cells in spatial evidence grids used for localization, this

vertical or horizontal restriction will likely produce minimal error relative to other obstacle positioning errors. However, as smaller area cells become available, this assumption may need to be replaced by some interpolation algorithm across neighboring cell or obstacle surfaces.

After applying these assumptions to the spatial evidence grid and creating a geometric layout for the room, it is relatively simple to apply the ray-tracing algorithm discussed in Chapter 3 for estimating both the direct field and the reverberant field. Works by Savioja [Savioja 1999], and Elorza [Elorza 2005] provide implementation level detail of the ray-tracing algorithm and discussions of accuracy in 3D. As our spatial evidence grids only model obstacles in two-dimensions, our ray-tracing models differ slightly from these described works by limiting all rays to a single height. Furthermore, without knowledge of the specific material composition of the room, we assume that all surfaces are completely reflective (0% energy absorption). Appendix B.6 describes in pseudocode our implementation of ray-tracing.

In Figure 4.23, the ray-tracing algorithm is used to predict direct and reverberant field estimates for a filter generating wind noise at 54-dBA. Figure 4.23 (Top) describes the direct field in which a gradual drop-off from the maximum sound level can be seen as the energy dissipates. “Acoustic shadows” can also be seen in this direct field, where the sound from the direct field is blocked due to the presence of obstacles. Figure 4.23 (Middle) then describes a reverberant field estimate, where sound levels are still highest in the vicinity of the sound source due to the close proximity of the walls below and on the right. Note that the highest volume in the reverberant field is actually 3 dB quieter than the region near the source in the direct field. Figure 4.23 (Bottom) finally combines

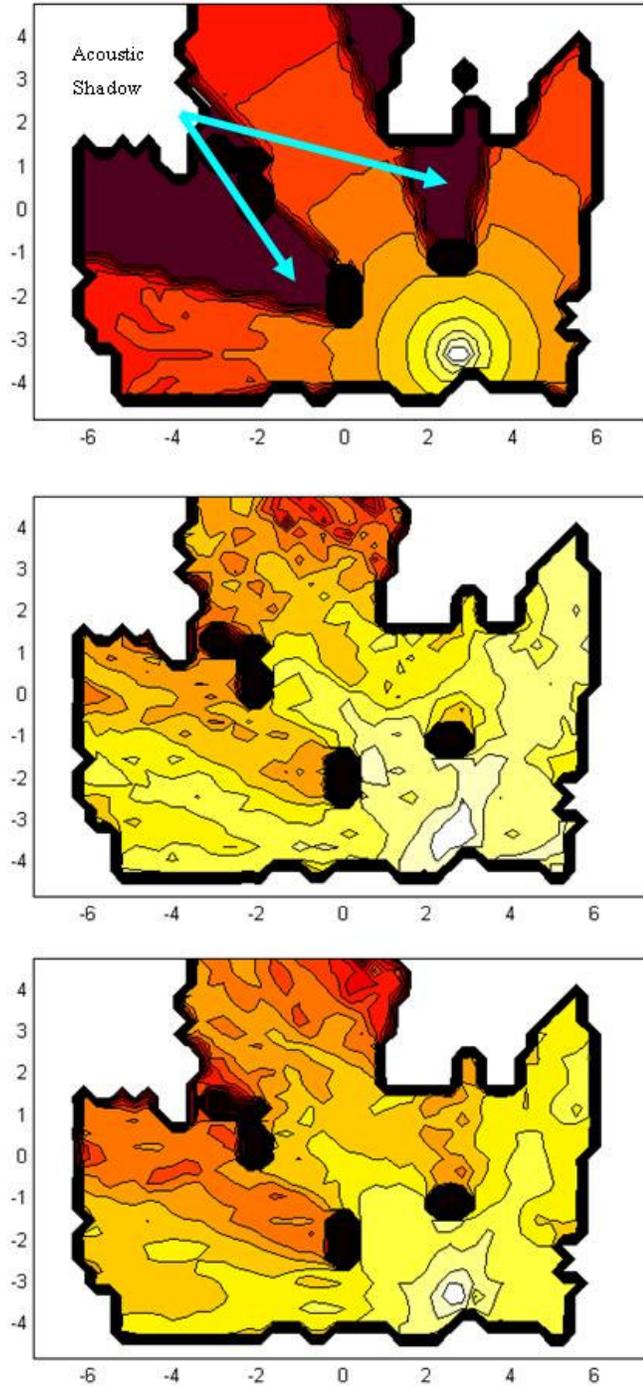


Figure 4.23. Maps of sound propagation created using a 2D robot-created evidence grid of the obstacles in the environment. (Top) Direct field of an ideal point source with known volume and position, note the acoustic shadows due to the 2D assumption. (Middle) Reverberant field created from the same source. (Bottom) Combined field, demonstrating characteristics of both the direct field (strong acoustic shadows) and the reverberant field (strong response near walls).

the two fields, maintaining the key features of each. The gradual drop-off in energy from the direct field dominates the combined field in the region near to the source, but is not as strong as the distance from the source increases and the reverberant field contributes more of the energy.

#### **4.3.2 IMPROVING THE ESTIMATE QUALITY**

A great advantage in using spatial evidence grids for finding the geometric layout is that they are created easily using software and hardware found in common mobile robotic applications. As such, spatial evidence grids, unlike more complicated representations of the environment, can be quickly adapted for acoustically-aware applications. Unfortunately, despite the ease with which evidence grids can be acquired and adapted to the problem of sound propagation, there are a number of inherent drawbacks in using them for identifying geometric layout: spatial evidence grids are very noisy, they do not accurately represent surfaces, and they are two-dimensional. What follows here is a discussion of how each of these problems affect the final representation, and how the error might be reduced in future implementations.

##### Noisy, Incomplete Spatial Evidence Grids

The first problem with the geometric layouts created from spatial evidence grids is that they are very noisy. While a robot travels around the room creating an evidence grid for localization, it does not always gather enough data about the environment to fill in the map completely. This can be seen in the displayed spatial evidence grid (Figure 4.23, Top), where some regions that should be empty are grayed, rather than white. Additionally, the walls have a number of holes in them, and some parts of the room are

completely unknown. Our mapping software (*pmap*, with Player/Stage) suffered particularly in this case, as its purpose was localization rather than mapping. Other algorithms for building maps and localizing the robot that have been rigorously initialized can possibly do a better job at creating the resulting map, but will still suffer from the same general problem. If the robot does not enter an area, or something blocks the view of the robots' sensors, then a map cannot be created of the missing area. Without the missing information, however, our sound propagation models will suffer in accuracy.

The solution to this problem requires gathering more data. Putting more sensors onto the robot may have some effect, as the additional data will be gathered from a different viewpoint. More importantly, however, the robot needs to explore the environment thoroughly, actively investigating unknown areas and trying to find better viewpoints for inaccessible regions by incorporating the need for an accurate map into its navigational control strategy [Thrun et al. 2005].

### Surface Estimation Errors

The second problem with these particular maps is the accuracy of the representation. As mentioned earlier, if each grid cell covers an area of  $0.1\text{m}^2$ , then the actual surface of the object could be located anywhere within a similar sized area. While our earlier assumptions of completely occupied grid cells and reflections along the cell boundary allow even large-grained spatial evidence grids to be used for constructing an estimate of the reverberant field, knowing exactly where the ray reflects, and at what angle, are important characteristics for accurate sound field estimation.

There are, however, alternative approaches for surface estimation that already exist in the robotic mapping community, although none have yet been tested with sound propagation algorithms. The use of particle filters to estimate the surface directly, rather than the presence of obstacles, should produce a closer match to the real surface. Unfortunately, the resulting surfaces are very noisy, and will produce significant refraction effects in sound propagation. Smoothing the surface may reduce these effects, but may also limit the accuracy gained by using a particle filter approach. Another possibility is to build predictions about the shape of the surface being mapped into the surface estimation algorithm. For instance, if most of the sensor readings will be from a small number of flat surfaces (common with indoor environments), then the algorithm can estimate the number of flat surfaces, and predict which readings belong to which surface[Thrun 2002]. Details about small objects on the surfaces will most likely be lost, but the resulting surface is flatter. If the surface is actually flat, then incorporating such knowledge into the algorithm may produce a better quality map for the purpose of modeling sound propagation.

### Two-Dimensional Descriptions of a 3D Environment

The third drawback with these spatial evidence grid representations is that they are 2-dimensional. This is due to the use of just one laser mounted on a flat surface (i.e. the top of the robot), which will produce models at only a single height. Although modeling will still work in 2-dimensions, the ability to accurately predict soundscape features such as echoic locations and acoustic shadows is certain to diminish without 3-dimensional data. This is in part due to missing reflections from above and below, and in

part to inaccurate representations of obstacles. Of particular interest are small obstacles, such as small boxes, which should have a minimal impact on sound levels, but are tall enough to be detected by a small robot. These obstacles result can result, when using ray-tracing, in significant acoustic shadows (Figure 4.23, Top) that are not nearly as pronounced in the real environment, due to sound propagating over the top of the obstacle.

Perhaps the most obvious solution to this problem is to use a 3D mapping algorithm on the robotic platform. If the robot is equipped with more sensors for viewing upwards, rather than along a single horizontal plane, the same algorithms described for improving surface estimation accuracy (particle filters, including surface models, etc.) can also be applied to the 3-dimensional modeling problem. There exists extensive work in mapping with upwards pointing lasers [Kaess et al. 2003], cameras, and combinations of both [Biber et al. 2004; Thrun et al. 2005]. At this point, however, none of these models have been tested with sound propagation due to the required hardware and/or software constraints. Furthermore, improved accuracy is not guaranteed by the adaptation of a 3D mapping algorithm. As with 2D evidence grids, they are still subject to the data collection problem and suffer from similar surface estimation errors that may incorrectly predict reverberation effects. In general, the application of robot generated 3D models to sound propagation needs more study.

Given this variety of problems, 3D models of the environment are not a guaranteed way for a robot to improve accuracy. Another possibility for overcoming this problem of small obstacles in 2D is to separate the direct field from the ray-tracing algorithm. The direct field does not actually require ray-tracing for estimation, as the

direct field energy can be estimated as decreasing linearly with the square of the distance from the source. The reverberant field can then still be estimated using ray-tracing. Seen in Figure 4.24, the resulting combined field for the filter from this method does not demonstrate as pronounced of an acoustic shadow as the pure ray-tracing solution, which used the geometric layout in calculating both the direct and reverberant fields. An acoustic shadow is still visible, due to reverberant effects, but the difference in volume is much less. Of course, this method is not entirely accurate either, as some obstacles present in the geometric layout may have been tall or massive enough to produce large acoustic shadows, and even the small obstacles likely have some impact. Still, this option allows the designer of a robot controller more freedom for tuning their system to a particular type of environment.

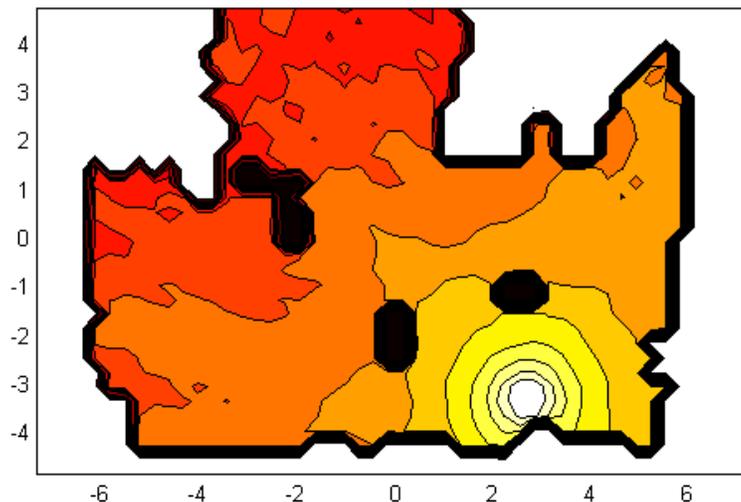


Figure 4.24. Map of the auditory scene combining a simplified direct field model with the reverberant field.

### 4.3.3 PATH INFORMATION - SUMMARY

At the conclusion of this section on path information, it should be understood that the use of robotic mapping for sound propagation has much work remaining. This work in modeling reverberation from 2-dimensional maps has primarily served to provide insight into the difficulties associated with the mapping the reverberant field, rather than try to build a working system for use with an acoustically-aware robot. It is a significant contribution to the field, because no others have used either environment maps created from robotic data or sound source knowledge generated by an autonomous mobile robot to build estimates of sound propagation through the environment. This dissertation demonstrates the use of both sets of information, and applies them to both the reverberant and direct sound fields. Figure 4.25 demonstrates a combined direct and reverberant field estimate for the two sources localized and investigated at the end of Section 4.2.

Future work in this problem of path estimation will attempt to validate the use of this robot gathered information with sound propagation models. One such validation method is to compare the resulting estimates with measured data at random locations in the environment. Although this is the typical validation method in architectural acoustics, a high level of accuracy does not always transfer to improvements in robotic applications. Therefore, in Chapter 6 we propose work in applying ray-tracing estimates from robotic data to the stealthy approach problem to improve accuracy and generality. It is in the context of this real robot scenario that we intend to demonstrate the usefulness of this information, and resolve some of the remaining issues about accuracy, such as: how accurate does the sound fields estimate need to be to improve performance? And, when is there a need for a better quality robot-generated map?

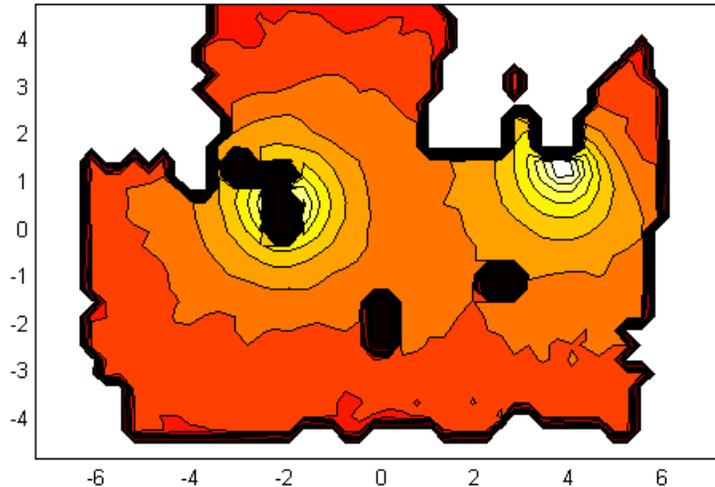


Figure 4.25. Estimated sound levels for the combined direct and reverberant fields, created from purely robot-collected information. The robot collected the obstacle map, localized sources, and identified directivity and volume. For this sound propagation estimate, the geometric layout is only included in the reverberant field estimates.

#### 4.4 BUILDING MAPS WITHOUT MODELS

The greatest advantage of estimating the auditory scene using models is that all of the information can be described *a priori*. Once a sound source has been discovered, the robot can use it over and over again when building a model of the auditory scene, so long as it detects that it is still operating and has not significantly changed. Identifying such changes may require some work on the robots' part to occasionally check up on the source, but major characteristics of medium duration sources, such as directionality and location, are unlikely to change very often. Such sources are instead more likely to be turned on/off and change volume, things easily determined by a mobile robot. Section 5.1 describes a scenario in which this is accomplished.

The greatest drawback to estimating the auditory scene using models is that the robot may not have, and/or may not be able to acquire, all of the critical information about the environment (or the information is inaccurate). For example, the application may be highly sensitive to the level of reverberant sound. In that case, an adequate model may require 3D models of the environment and material specifications to accurately model the sound propagation. Such information is not always easily determined *a priori* and, although work in both 3D modeling and material characteristics identification [Krotkov 1995] do exist, this info can be very difficult to accurately determine using a robotic platform. As another example, perhaps the robot only has a single microphone for listening to the ambient noise. The robot could try to use a gradient descent strategy to move towards and localize the sound source, but the resulting accuracy is questionable as the robot may become stuck in a local minima. In either of these cases, an alternative to estimating sound flow from models is to measure it directly using the robot.

As originally reported in [Martinson and Arkin 2004], a noise map for robotic use can be created from some arbitrary number of samples through interpolation. Provided with some set of samples collected by one or more microphones, and the relative positions at which each sample was collected, we can use some form of function approximation to estimate the sound field directly from the samples without collecting source localization, directivity, etc. In the original work, K-means interpolation was reported as it was fast and produced a quick approximation over areas that had not necessarily been sampled. Cubic interpolation has since been implemented as well, as it produces smoother contours in densely sampled regions at the expense of increased

computational complexity. The creation of a sampled data noise map using any available interpolation method is discussed in more detail in Appendix B.7.

Regardless of the type of function approximation used to estimate the sound field, the advantage of the interpolated noise map is that, while it still requires robot localization, the robot only needs a single microphone to get a rough approximation of the entire sound field. Array information can be incorporated as well, but it is only needed as separate microphone inputs. The drawbacks to this interpolation method, however, are twofold. First, the maps are heavily influenced by robot ego-noise, which may or may not be constant across the sampled region [Martinson and Arkin 2004]. Second, the interpolation method cannot reliably estimate anything beyond the sampled region, limiting its effectiveness for guiding robotic navigation. Given these sizeable drawbacks, we primarily used the interpolation method as an alternative, validating the sound fields estimates in areas that been heavily sampled.

#### **4.4.1 COMPARISON WITH MODELS**

Ideally, the sound fields framework described in Chapter 3 would lead to a faithful representation of the auditory scene. In practice, however, it might vary significantly from the ground truth if there is missing information, such as other sources, transmission effects, or 2D vs 3D reverberation models. The interpolation method for constructing noise maps provides a convenient alternative, at least in the sampled region. In Figure 4.26, the source directivity indicates that the fan source in the MRL environment should be loudest to the right, but should still remain fairly loud to the left, at least in some spots. Both representations demonstrate these common features,

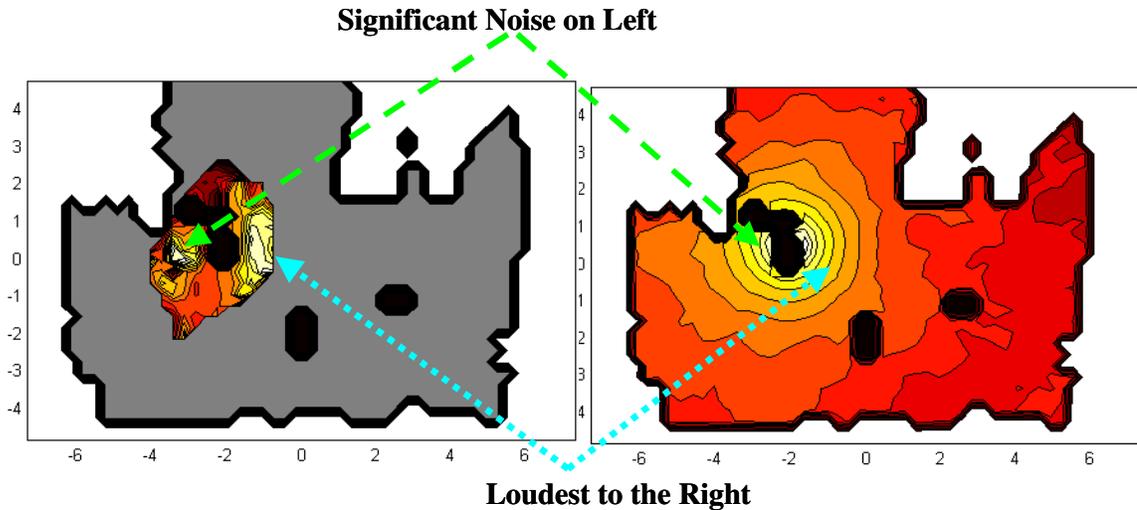


Figure 4.26. Comparison of the interpolated noise map (left) to the sound fields model (right) for the fan source. In both maps, the directivity of the source is indicated as being loudest to the right, but still fairly loud in the rear. Using interpolation, however, generates a noise map that is only valid for the small area that was sampled, while the sound fields map can make predictions for the entire room.

suggesting that the sound fields' estimation is a reasonably good representation of the soundscape. Looking at another example, the radio source (Figure 4.27) from the MRL environment, the different representations indicate common source orientations, or regions of maximum volume. In general, the interpolation method should be reasonably good at predicting local phenomenon such as the direction of maximum volume or other hot spots due to reverberation, so if the two resulting maps do not demonstrate similar characteristics, then it may indicate to a potential problem to a robotic platform.

An example of missing information can be seen in Figure 4.28. The obstacle map of the NRL AI Laboratory is very cluttered, and really needs 3-dimensional data for a faithful reconstruction. In this case, the source, located on a bookshelf next to the wall, is correctly localized and a directivity model is created. But the final model, utilizing the

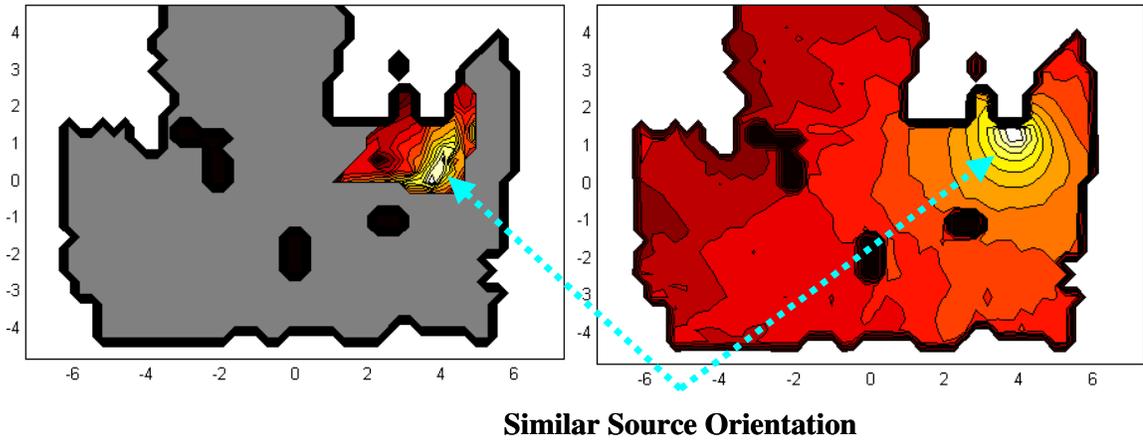


Figure 4.27. Comparison of the interpolated noise map (left) to the sound fields model (right) for the radio source. Both maps indicate the loudest region as directly to the front of the sound source, rather than to a side, or omni-directional.

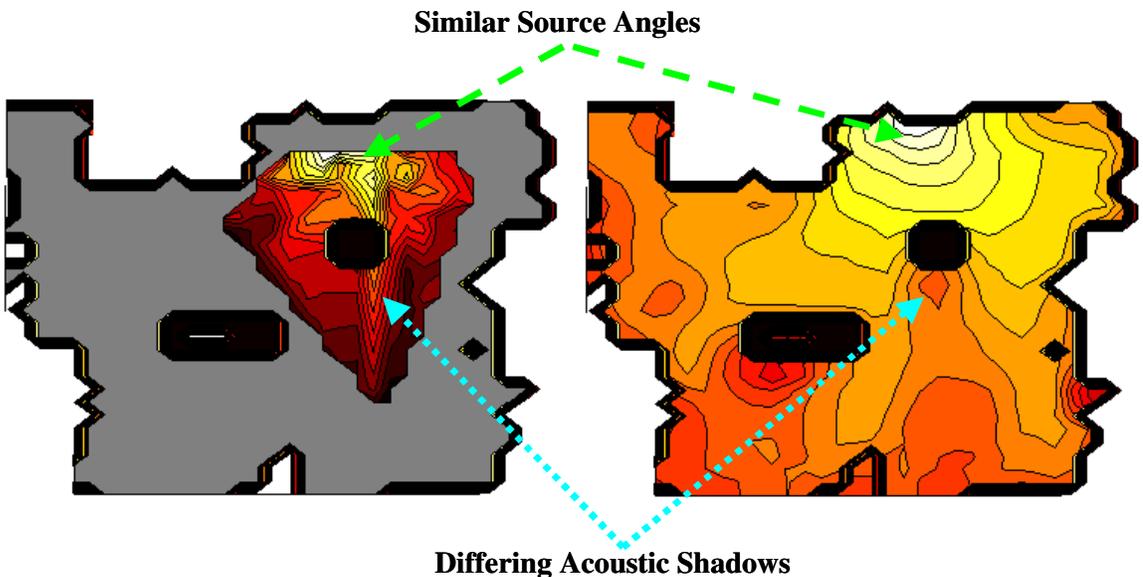


Figure 4.28. Demonstration of the effects of poor reverberation models in the NRL AI Center. The interpolated map shows that the sound can still be quite loud behind the obstacle, while the sound fields map predicts significant decrease in volume.

obstacle map for direct and reverberant fields predicts an acoustic shadow behind the box that does not appear in the interpolated noise map. Although the spike in the interpolated noise map is possibly just an artifact due to the other people in the room, there is still no predicted shadow behind the box. A robot designed to spot such a deviance should probably reinvestigate the area.

Naturally, spotting such discrepancies between these maps is a very hard problem. How significant does the difference need to be before the robot should reinvestigate? Even after identifying a problem, predicting a cause would be even more difficult. For example, the source appearing to point in different directions could indicate a faulty directivity model, a missing source, poor reverberation estimates, etc. Given the difficulty of the comparison problem, this dissertation only used the interpolated noise maps for debugging purposes. If the human designer of the robotic system could see a noise map, then they had more information available for identifying problems, missing information, and faulty sound fields construction. In the long term, however, the combined use of both types of maps could lead to more robust systems. By identifying the discrepancies between maps, a robot can make use of its limited time to better direct its investigatory efforts.

## **4.5 CHAPTER SUMMARY**

Acoustical awareness, as defined in Chapter 1, is the “coupling of action with knowledge about the acoustic environment”. In Chapter 3, we then explored this problem in more depth, differentiating between types of awareness, and describing a mathematical framework for making predictions about the soundscape at different

locations in the environment. This ability will, over the next 3 chapters, prove to be very useful to a robot performing some acoustic application, as it provides a baseline for constructing plans to maximize the performance of an acoustic application.

By itself, however, the sound fields estimation framework does not make a robot acoustically-aware. In order to make any predictions about the auditory scene, the robot must have some knowledge of the primary acoustic entities: the sound sources, the receivers, and the environment itself. From where, though, does the robot acquire this information? Certainly some of this information could sometimes be made available *a priori*. What happens when the information is not available? Or maybe the information that is available is not enough? Or, what does the robot do when the auditory scene changes? A robot that is completely dependent on a human for its information about the primary acoustic entities is limited in terms of scope and performance. Chapter 4, therefore, focused on resolving this data collection problem that was at the heart of the second sub-question of the thesis, answering how and with what representations can we combine disparate sensory data together for the purpose of enabling acoustical awareness. The emphasis, in particular, was in representations, or tools, that enable the autonomous collection by a mobile robot of the information needed for the mathematical framework presented in Chapter 3:

- **Source Location in 2D**

Traditional sound source localization algorithms for microphone arrays were extended to incorporate robotic movement for the purpose of localizing one or more simultaneously operating sound sources in two-dimensions.

- **Directivity and Volume Models of Sound Sources**

Using an area-coverage algorithm, a robot can investigate regions identified as likely to contain a sound source, building models of directivity and overall volume for the source.

- **Sound Function Classification**

From samples collected during robotic investigation of a sound source, a model of the sound function can be constructed from mel-frequency cepstral coefficients, allowing the robot to classify samples as belonging to a particular source.

- **Geometric Layout**

Borrowing from existing work in creating spatial evidence grids, the geometric layout of the room can be approximated by a mobile robot for use with ray-tracing algorithms to estimate the direct and/or reverberant fields.

In addition to acquiring information about specific characteristics of entities in the auditory scene, this chapter also identified a method by which a robot can model the combined fields simultaneously:

- **Interpolated Noise Maps**

The sound fields framework provides a convenient tool for modeling information about known acoustic entities. By creating an interpolated noise map directly from the sample, the robot now has a tool with which it

can check the resulting estimate over small areas, and, through comparison, attempt to recognize the presence of missing knowledge.

Each of these representations provides another tool to an acoustically-aware robot, enabling it to find sources, model those sources and/or the environment, and double check the resulting sound fields estimate. What has also, hopefully, been demonstrated over the course of this chapter, however, is the flexibility of the sound fields framework for enabling acoustical awareness. If information about the shape of the room is available *a priori*, then, of course, the robot does not need to re-acquire that knowledge through robotic mapping. If the robot does not need accurate source directivity models for its application, then the robot does not have to spend the time to acquire those models through area coverage heuristics. The sound fields framework (described in Chapter 3) allows for a wide variety of circumstances, applications, and *a priori* knowledge under which a robot can still successfully navigate with respect to the soundscape, can still improve its performance at acoustic tasks, and can still be acoustically-aware. This chapter then demonstrates a set of tools for acquiring more information autonomously in order to enhance that awareness, if the robot has the need, the time, and the resources to acquire it.

## CHAPTER 5

### THE AUTONOMOUS MOBILE SECURITY ROBOT

In Chapter 3, we used the theory of sound fields to identify information useful to a mobile robot in understanding the flow of sound through the environment. With knowledge about the receivers, the sources, and the paths through the environment, a model of the auditory scene can be created to guide the robot in improving its performance. Then in Chapter 4, we identified how, and from where, a robot can reasonably expect to acquire this knowledge or information. When available, a priori information can be utilized, but, in lieu of missing information, the robot itself can also gather enough data with reasonable accuracy to create direct and reverberant field models of the environment. What has not yet been addressed by either of these chapters, however, is the utility of these results. Acoustics, and robotic applications that use acoustics in any fashion, are wide areas of research. A robot can be a listener, or it could be a sound source. In this chapter, we focus on the problem of how acoustically-aware control can be applied to a robot-listener to improve classification of sound sources, source localization, and the general signal-to-noise ratio. This will answer, in part, the third, and final, sub-question posed in Chapter 1: How does acoustical awareness change with control over the source vs. the receiver? Chapter 6 will then discuss control over the sound source (a vocalization application). The application domain in which we will explore this first robot-control problem is a robot security guard.

The remainder of this chapter will focus on two robot applications designed for an acoustically-aware robot security guard. The first application uses knowledge of sound

flow through the environment to identify changed or new sound sources along a robot's patrol route. By using prior, or self-discovered, knowledge of the environment being patrolled, acoustical awareness allows the robot to predict what should and should not be heard at different locations throughout the environment.

The second robot security application focuses on moving the robot with respect to the acoustical environment. In addition to correctly classifying data heard along a patrol route, an acoustically-aware robot can also augment its path acoustically to better its chances of detecting an acoustic event. By making use of noise maps of the environment, a robot can strive to avoid areas of loud ambient noise, increasing its signal-to-noise ratio while listening to the environment.

## **5.1 RELATED WORK IN SECURITY ROBOTICS**

Security robotics is one of the relatively few, but growing number of application areas for mobile robotics that have seen significant commercial investment to date. Companies ranging from ActivMedia Robotics<sup>7</sup>, Cybermotion<sup>8</sup>, Denning Mobile Robotics, and iRobot<sup>9</sup> in the United States, to SECOM<sup>10</sup> in Japan, and YAAN Technology Electronics<sup>11</sup> in China have all developed semi-autonomous mobile robots for commercial and/or military security. The reasons for this explosion in commercial interest are twofold. First, awareness of security loopholes at many public and private institutions has been heightened by recent terrorist activities in the U.S. and Europe, as

---

<sup>7</sup> <http://www.mobilerobots.com/PatrolBot.html>, Accessed 5/16/2007

<sup>8</sup> <http://www.cybermotion.com/>, Accessed 5/16/2007

<sup>9</sup> <http://www.irobot.com/sp.cfm?pageid=138>, Accessed 5/16/2007

<sup>10</sup> <http://www.secom.co.jp/isl/e/mission/index.html>, Accessed 5/16/2007

<sup>11</sup> <http://www.chinayaan.com/en/news.htm#>, Accessed 5/16/2007

well as the wars in Afghanistan and Iraq, creating a demand for qualified security personnel that has been difficult to meet. Given the increased demand, automated solutions ranging from fixed surveillance networks to robotic systems that can reduce the number of required security personnel have been of great interest to the security community.

The second reason for the increased commercial interest is the nature of the job itself. Most of a security guard's time is spent waiting for something to happen, watching for a potential threat to the security of the installation. When such an event occurs, few dispute that a human is required (at least for now) to make judgments about the best course of action, and act to protect people and/or property. What a security guard does while waiting for such an event, however, such as patrolling the environment and checking locks/doors [Carrol et al. 2002], is repetitive, usually uneventful, and an ideal candidate for automated assistance. In the near future, a robot could undertake many of these duties autonomously, searching and/or waiting for a security event to happen while the security guard is busy at a remote location. Then when something does happen, the robot alerts a human guard who can take control of the robot to investigate the incident and make decisions about appropriate future actions.

Given the great desire for robotic assistance in the field of security, it should not be surprising that many different areas of research within robotics are working on applications that involve enhancements to security guard robots. These fields include tele-autonomous control [Chien et al. 2005; Liu et al. 2005], multi-robot protection [Guo et al. 2004], event recognition [Treptow et al. 2005; Luo et al. 2006], and stealthy or covert path planning [Birgersson et al. 2003; Marzouqi and Jarvis 2005; Kennedy et al.

2007]. In each of these cases, being acoustically aware can help enhance performance. The stealthy or covert path-planning task, which will be discussed in more depth in Chapter 6, can use knowledge of the auditory scene in order to hide the robot from an observer. All other applications need to know the auditory scene in order to separate significant sounds from uninteresting known ambient noise sources. In general, having knowledge of the auditory scene, and being able to incorporate that knowledge into a security guard application is important. In order to use this knowledge for performance enhancement in security operations, however, the big question that remains is how well can the robot monitor the auditory scene?

## **5.2 MONITORING THE AUDITORY SCENE**

The job of a security robot is a difficult one. Even when problems with vision, movement, localization, etc. are all removed, focusing on just its auditory role, the problem is very tough. Let us imagine for a moment, a typical auditory scene confronting the night-watch robot for some local manufacturing company. Just in the offices away from the factory floor, the robot will detect HVAC (heating, ventilation, air-conditioning) systems blowing air into the rooms, sometimes changing in intensity as the building temperature fluctuates. Office computers, some of which have been left on for the night, are humming at random intervals, even occasionally moved by the building occupants during the day to make space for other activities. Fountains, radios, even the lights also emit a constant ambient noise that a robot has to ignore while searching for unusual auditory events signaling a security problem. Once the robot moves onto the factory floor, the problem gets even worse as heavy machinery operates at all hours of the

night, dominating the auditory scene in their vicinity and raising the reverberation levels in the room to levels rendering even speech near unintelligible. Through all of this noise, wherever the robot is located, it is expected to identify not only impact noises like glass breaking, but also unusual auditory activity associated with malfunctioning equipment, fires, burglars, etc. It is a daunting task, requiring large amounts of knowledge about the auditory scene to complete. Fortunately at night, the environment should be somewhat more predictable with few or no people present.

In this section, we will focus on a subset of the general acoustic monitoring problem, tracking changes in medium-to-long duration sources present in the auditory scene. Given a known state of the environment, how can the robot determine if the auditory scene has changed in some way? And if it has changed, where should the robot focus further investigation? Being able to answer each of these questions will allow, in the future, a robot to better ignore ambient noise effects, while searching for significant aural ambiguities that constitute a risk to security.

Presented in this section is a set of algorithms that the robot can use to answer each of the three following questions after completing a single patrol through the environment:

- Did the environment contain a new source?
- If a new source was present, then where was it located?
- Were there any changes to sources known to be active in the environment?

Each of these algorithms is constructed using the representations developed in Chapter 4, particularly auditory evidence grids and mel-frequency cepstral coefficients (MFCCs). Each algorithm is also designed to work with no prior knowledge, but can

also incorporate such knowledge from earlier passes through the environment. The success rates for each algorithm are then evaluated in a series of tests, demonstrating that autonomous monitoring of the auditory scene is a realistic task for an acoustically-aware mobile robot.

### 5.2.1 EXPERIMENTAL SETUP

The robot hardware that was used for this task is the Pioneer-2dx robot equipped with a SICK LMS200 for localization and obstacle avoidance. Four ATR35S\*2 omni-directional condenser lavalier microphones are mounted to the back of the robot, sampled on demand by a laptop computer (located under the microphones) equipped with Measurement Computing's PC-CARD DAS16/16 data acquisition card. This same robot configuration was used previously for performing patrol and area-coverage tasks in the Mobile Robot Laboratory. The fully equipped robot can be seen in the laboratory in Figure 5-1. This robot uses the underlying Player/Stage [Gerkey et al. February 2006] controller for robot localization, obstacle avoidance, and path planning algorithms. More details about both the hardware and software setup can be found in Appendix A.

The environment used for this phase of testing is a 10x10-m<sup>2</sup> section of the Mobile Robot Laboratory. The sources that could be detected by the robot were as follows:

- *Filter* – a home air filtration unit for a medium sized room (Figure 5.1). It generates fan noise at different volumes, depending upon the speed of the unit. For testing purposes the filter was set either on low, or on high.

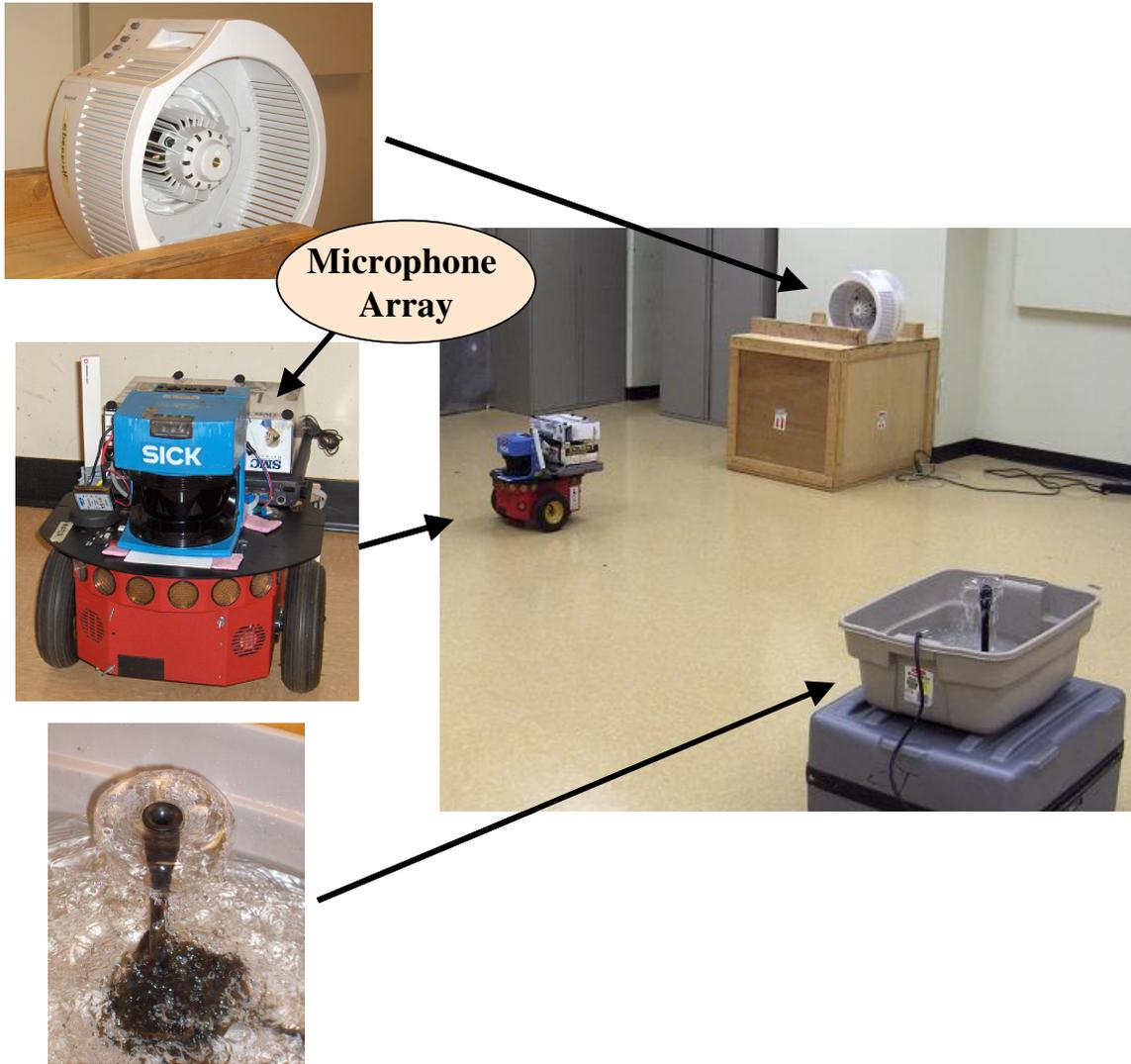


Figure 5.1. Pictures of the sound sources (left) dominating the auditory scene in the Mobile Robot Laboratory (right) for the acoustic monitoring task. The microphone array used to measure the auditory scene is shown mounted to the back of the mobile robot.

- *Fountain* – a small garden fountain (Figure 5.1) situated in the middle of the room. Besides being turned on/off, it could also be turned down to a lower setting generating less noise.
- *Radio* – a radio was placed in one of 10 randomly selected locations throughout the environment for a robot to detect while patrolling the environment. The radio was playing miscellaneous songs from a “Best of Journey” compact disc.
- *Robot* – In most trials, the most commonly detected sound function was actually the robot, due to the close proximity of the microphones to the robot motors/wheels and the absence of other loud sounds through most areas of the patrol route.

Before any of the trials were completed, the sound functions (MFCC classification vector) were determined for the filter, the fountain, and the robot. In the case of the filter and the fountain, the robot discovered and investigated each source prior to beginning the classification trials while only the source being investigated was enabled. The sources were determined to be largely omni-directional, and of similar volumes (60-dB for the filter, and 61-dB for the fountain). The sound function of the pioneer2-dxe ego-noise was determined separately by moving the robot through an empty environment while sampling and averaging the resulting MFCC vectors from those samples. Details on building classification vectors from sampled data can be found in Chapter 4, and again in Appendix B.4.

The obstacle map of this environment, with source positions indicated, can be seen in Figure 5.2. The 10x10-m<sup>2</sup> testing area seen in this figure fully encompassed one obstacle, and was intruded upon by two other obstacles, requiring the robot to possibly

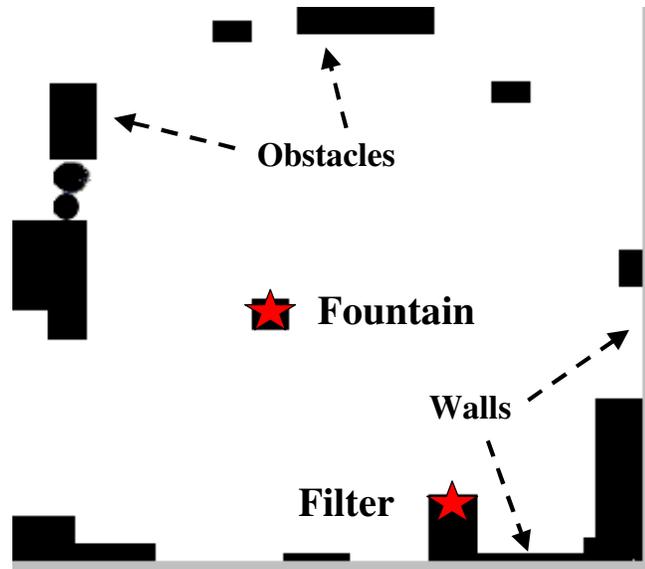


Figure 5.2. The obstacle layout used for the acoustic monitoring task. Within this environment, there were two sources whose positioned never changed.

navigate around them to reach a waypoint. The regions closest to the walls on the bottom and left of this map were not included in the testing area due to difficulties in reaching those areas using the Player/Stage Vector Field Histogram controller [Gerkey et al. February 2006] for moving while avoiding obstacles.

### 5.2.2 PATROLLING THE ENVIRONMENT

Before the robot could determine if the environment had changed, it first needed to complete a patrol route through environment during which it collected sampled of the auditory scene. The task of patrolling the environment was accomplished using a waypoint mission. The robot follows the waypoints in a loop through the environment, sampling as it moves, and ending up back at the beginning of the route. To make the task as general as possible, we tried to avoid hand-selecting a route through the environment,

instead opting for an automated selection of waypoints by the robot. The automated process used the known obstacle map of the environment (Figure 5.2), and the sensing range of the microphone array (3-m), to guarantee that the robot passed within sensing range of every reachable location in the environment. The steps used to build this waypoint path are as follows:

- **Step 1 - Use the obstacle map to identify areas reachable by the robot.**

First eliminate all sections of the map within 0.5-m of an obstacle as unreachable, and then use Dijkstra's shortest path algorithm [Cormen et al. 1990] to identify whether or not there is a path to each cell in the obstacle map given the remaining cells. Appendix C-1 gives more details on identifying reachable areas of an obstacle map.

- **Step 2 – Divide the reachable area of the map into discrete grid-cells.**

The goal of the waypoint selection algorithm is to guarantee that the robot gets close enough to all locations in the environment to detect a sound source. As discussed in Chapter 4, the maximum sensing range of the spatial likelihood algorithm is 3-m, therefore, the goal is to select a set of waypoints that get the robot to move within 3-m of all locations that might contain a sound source. We will guarantee this by placing a set of waypoint targets that would accomplish this goal, regardless of the path chosen to reach each waypoint. This is accomplished by, first, identifying the smallest square area that includes all reachable areas of the map, and then sub-dividing the square into smaller square regions with a diagonal length less than the maximum sensing range. Since the robot might end

up at a corner of a cell, rather than the middle, setting the diagonal to max sensor range means that the robot will be able to detect changes from all parts of the square. Figure 5.3 shows an example of this discretization.

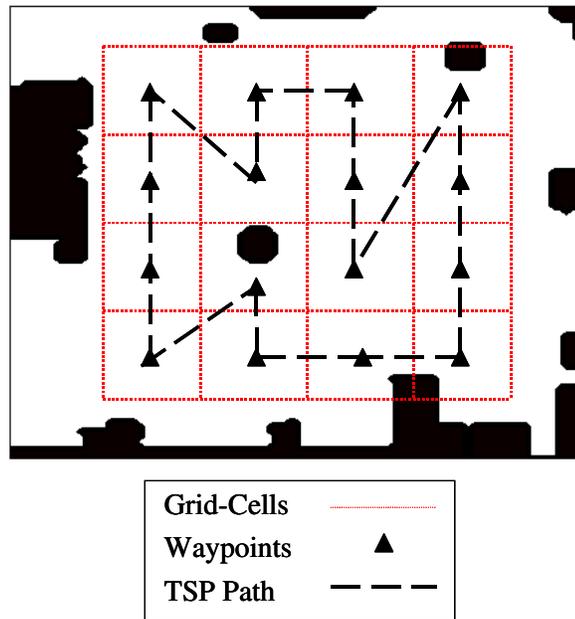


Figure 5.3. Discretized obstacle map through which a patrol route has been identified. The resulting route ensures that the robot passes close enough to all possible sound source locations to detect any sound sources.

- **Step 3 - Pick a target within each grid cell.**

Ideally, the robot should move to the center of each cell, so as to minimize its distance to all reachable areas of the map. If, however, an obstacle blocks the center, then choose the next closest reachable location to the center. Figure 5.3 demonstrates a set of chosen waypoints. Appendix

C.2.1 presents pseudocode for this process of selecting waypoints for patrolling the environment.

- **Step 4 - Find the shortest continuous patrol route through all targets.**

For this final task in identifying the patrol route, we use a traveling salesman heuristic. First, start with an arbitrary ordering of targets within the environment. Second, greedily swap nodes in the target order that will reduce the path length. Finally, repeat the greedy node swapping until the path length cannot be reduced any further. Appendix C.2.2 provides more detail on ordering the set of waypoints to reduce overall path length. Figure 5.3 demonstrates the results of the entire patrol route selection process on our chosen environment.

The result of this algorithm is a continuous patrol route through the environment that ensures the robot travels within sensing range of all reachable locations of the environment. Although this algorithm is general enough to be used within many environments that a robot might be located in, there are likely some non-rectangular environments where the chosen patrol route would be significantly less than optimal. In such a case, the resulting route could still be used, or a different patrol route could be automatically chosen, or even hand-selected, without significantly affecting the performance of the following acoustic monitoring task. It is only important that the selected path gets the robot close enough to likely sound source locations to be able to detect them.

### 5.2.3 TESTING AND EVALUATION

A total of 67 patrols were completed in this environment. The trials varied by the types and locations of active sources in the environment. Table 5.1 shows a complete list of all trials completed by the robot, ordered by the types of active sources. For each source configuration, the robot followed the automatically generated waypoint path from start to finish, passing close enough to all areas of the environment to detect new sources or changes to existing sources. As the robot followed the path, it sampled the auditory scene using its microphone array, and stored the results to a database for future analysis.

Table 5.1. List of trials completed by the robot for the acoustic monitoring scenario. All used the same patrol route, but varied in the types and numbers of active sources in the environment.

		Which Sources are Active				
Trial Name	# of Trials	Quiet Filter	Filter	Quiet Fountain	Fountain	Random Radio Location
No Noise	4	N	N	N	N	N
R	10	N	N	N	N	Y
FT	5	N	Y	N	N	N
Q-FL	4	Y	N	N	N	N
FT	5	N	N	N	Y	N
Q-FT	4	N	N	Y	N	N
FT+R	10	N	N	N	Y	Y
FT+FL	5	N	Y	N	Y	N
Q-FT+FL	4	N	Y	Y	N	N
FT+Q-FL	4	Y	N	N	Y	N
Q-FT+Q-FL	2	Y	N	Y	N	N
FT+FL+R	10	N	Y	N	Y	Y

Once the robot had completed a single patrol through the environment, it then stopped to analyze the collected data using the algorithms described in the following sections for detecting and localizing new sound sources, and detecting changes to existing sound sources. Overall, the goal of the robot is to identify what has changed from its prior knowledge of the configuration of the environment. If, for example, the robot believed that nothing was active in the environment (a standard assumption for a unexplored auditory scene), then the robot should be able to detect when any of the sources are active. In comparison, if the robot believed that only the filter was active, then the robot should be able to detect that the fountain had been enabled, or the filter sound had changed, or radio had been added to the environment. Any changes to sources in the auditory scene should be detectable from the sampled data collected during a single patrol loop through the environment.

For each of these tests, it is assumed that if something did change, it would still be active after the robot has completed a full patrol of the environment. This allows the robot to complete the patrol, and then check data from the entire patrol for changes, before going back to investigate further using the investigation process described in Section 4.2. Note that this final stage of investigation after identifying what specifically has changed in the environment, however, was not performed as part of these tests, as the focus was on detecting the change in the first place.

#### **5.2.4 DETECTING NEW SOURCES IN THE ENVIRONMENT**

The first problem confronting an acoustically-aware security robot monitoring the auditory scene is the detection of new sound sources in the environment. Given a list of

sources that the robot believed were active at the beginning of the patrol route, can it determine whether or not any sources not on its list are currently active? This includes both sound sources that the robot has not seen before, as well as sound sources that the robot simply believed were disabled. By being able to detect new sound sources in the environment, a robot can, when confronted with a new sound source, request the assistance of a human security guard to determine what changed, or if an old sound source, simply update its model of the environment to include the changes to the auditory scene.

The tool that an acoustically-aware robot can use best for this task is classification of mel-frequency cepstral coefficients (MFCCs). At the beginning of each trial, the robot knew what set of sound sources it expected to hear in the environment and what the MFCC class vector was for each known source. Furthermore, the robot also knew what it itself sounded like, having created an MFCC class vector to describe its own ego-noise. If there is a new source present in the environment, then an acoustically-aware robot should be able to determine that by using comparisons to source sound functions (i.e., class vectors).

Over the course of patrolling the environment, however, the robot may pass through many noisy areas. Some of these areas may belong to known, non-threatening sources, while others may belong to something new, which the robot needs to identify and investigate further. Unfortunately, if there are several different sources in the environment, the number of samples indicating a new source may be relatively small compared to other previously investigated sources. Furthermore, the mere presence of such samples is not always indicative of a new source, since such samples may merely

contain an extreme example of noise from a previously investigated source. To overcome this problem, we need to look at the data in smaller portions.

According to the investigations reported in Chapter 4, MFCC classification results are highly dependent on volume. The higher the volume of the new source relative to other sources in the environment at the measurement point, the more likely it is that the sample will be classified as belonging to something new. Therefore, a new source, rather than being indicated by a large number of overall samples, should instead be indicated by regions with a large number of unexpected or unclassified samples. For this reason, searching for new sources needs to be done at short regular intervals (1-m) along the patrol route. The algorithm for identifying the presence of a new sound source is completed in the following steps:

- **Step 1 – Identify Regular Intervals**

Using the data collected from an entire patrol route, identify a series of locations 1-m apart along the robot patrol path. These selected points serve as local, regular intervals at which new sources are searched for.

- **Step 2 – Classify Samples**

Classify all samples collected by the robot along the robot patrol path. Identify, in particular, which samples belong to known classes vs. unknown classes.

- **Step 3 – Find Percentage of Unknown Samples**

For each target point, identify the percentage of samples within a 2-m radius belonging to unknown classes. This sample percentage indicates the likelihood of a new source occurring at this location in the

environment. The 2-m radius is the same radius used for identifying directivity of a sound source, beyond which it was estimated (see Section 4.2.2) that the reverberant field dominated the samples.

- **Step 4 – Find the Maximum Likelihood (ML)**

The maximum likelihood along the patrol route is the likelihood, overall, that a new source exists in the environment. If any given location in the environment was recorded as containing more than 20% “new” source samples, then the patrol route was classified as containing a new source. The choice of 20% was determined experimentally for this environment from a 20-trial subset of the trials listed in Table 5.1.

This maximum likelihood approach to identifying whether or not a new source existed in the environment had a 92% success rate classifying sessions with no new sources, and an 83% success rate at classifying sessions with one new source present. These results were averaged over all source configurations, including those with 0, 1, and 2 known sources present in the environment. Overall, the average success rate for classifying environments as containing or not containing sources using the maximum MFCC concentration approach was 86%.

Table 5.2 presents the mean maximum likelihood of a new source being present, averaged across the trials by the type of source being detected by the robot. Note that the total number of trials listed in this table is larger than 67, as some trials could be used more than once with different belief states. For example, a trial where both the fountain and the filter were active, could be treated as a trial with no new sound source, a trial with a new fountain source, or a trial with a new filter source. In testing this algorithm, we did

Table 5.2. The relative performance of using the proposed maximum likelihood approach for detecting each type of source in the environment as a new source.

	No New Source	Radio	Filter	Fountain	Quiet Filter Or Quiet Fountain
# of Trials	24	30	10	10	16
Mean Likelihood of Detecting a New Source	8%	56%	65%	65%	49%
Identification Success Rate	92%	87%	100%	90%	75%

not examine any belief states where more than one new source was present in the environment.

As expected, the identification success rate shown in Table 5.2 suggest that the sources which are detected the best by this algorithm are the constant noise, loud sources. If the source has been turned down in volume, then it can be hidden by robot noise, which averages 52-dB as recorded by the on-robot microphone array. Similarly, the detection of the radio, which was not at constant volume, and placed at multiple locations throughout the environment, may have also suffered from loud robot ego-noise. It would mask quieter parts of the music or make classification of distant sources (2-3 m away) more difficult.

### 5.2.5 LOCALIZING NEW SOUND SOURCES IN THE ENVIRONMENT

The second question that a robot monitoring the auditory scene can ask is where is the new sound source? This can be used in conjunction with the previous step identifying that a new sound source is present in the environment, and/or other sensors distributed

through the environment could provide the same information. The goal, regardless of the specific sensors/algorithm used to identify that a new sound source exists in the environment, is to identify the most likely place for a new sound source to be located, so that a robot can focus future investigation on that area. In this fashion, a robot is using its accumulated acoustic knowledge from previous tasks to guide affect its decision-making processes.

The tool that will be used for this task is auditory evidence grids, and in particular, the iterative clustering algorithm described in Section 4.2.1 for finding sound sources by creating additional auditory evidence grids using subsets of the collected data. There are two differences, however, between this scenario and the previous work in auditory evidence grids. The first such difference is that the robot has available a priori knowledge about the sound sources and the sound functions that should be present in the environment. We already used the sound function knowledge to good effect, identifying whether or not a sound source is present. Knowledge of source locations can also be used to reduce the iterative process. If the robot knows that there is supposed to be a fountain in the room, and it is looking for something other than a fountain, then it does not need to localize the fountain in the first map, and can exclude samples that point at it from the very beginning.

The second difference between this scenario and the earlier work is that data collected here are relatively sparse, leading to a higher number of phantom peaks in the resulting evidence grid that need to be filtered out. The reason for this greater number of phantom peaks is the increased influence of a single measurement on the auditory evidence grid. Usually, a single loud reverberant sound produces a spatial likelihood that

has little influence on the overall result, so the largest cluster is typically also the correct position. With smaller numbers of samples, however, a few reverberant samples pointing in the same direction may build a significant phantom peak. Therefore, instead of only taking the largest cluster from the resulting evidence grid and calling it the new source location, we will instead select all clusters that have a minimum footprint in the auditory evidence grid of at least  $0.5\text{-m}^2$  and then use the characteristics of these clusters to separate the real sound source location from the phantom peaks.

### Picking the Most Likely Location

As discussed in Chapter 4, there is no single criterion for correctly identifying the new source location. Instead, criteria such as cluster variance and the percentage of samples pointing at the cluster centroid give clues to the likelihood of the cluster containing a sound source. The following sequence of steps illustrates the process for extracting each potential source location, and the properties we use to identify the most likely position of the source:

- **Step 1 – Build the Auditory Evidence Grid**

Identify the set of samples that do not point at a known sound source (see Iterative Clustering in Section 4.2.1) in the environment and build an evidence grid from those samples.

- **Step 2 – Extract all potential source locations**

Apply the clustering algorithm (Section 4.2.1) to the auditory evidence grid, and extract the centroids of all clusters that cover an area of more than  $0.5\text{-m}^2$ .

- **Step 3 – Build new Auditory Evidence Grids for each potential location**

For each centroid extracted in the previous step, build a new auditory evidence grid using only those samples within a 3-m radius (the estimated sensing range) of the centroid. As before, only include those samples not pointing at a known source.

- **Step 4 – Extract cluster properties from the new grids**

From each of the new grids, identify: (1) the variance of the largest cluster, (2) the percentage of samples pointing at the largest cluster, and (3) the distance the largest cluster from the center of the retargeted grid.

This sequence of steps identifies a series of locations, and step 4 presents the three criteria to be used in gauging the likelihood of the new source existing at each location. For each of these criteria, a range of values was determined over which the apparent cluster was of questionable reliability, difficult to identify as being a real source or not. Each range was determined empirically using a 20 trial subset of the trials listed in Table 5.1. Note that it is very likely that some of these ranges, most noticeably the cluster variance, would need adjusting for different microphone/sampling hardware. The following list describes each of the criteria, and the range [Worst, Best] over which a potential cluster is of unknown likelihood:

- **Cluster variance - [3,1]**

The best cluster variance was less than 1. For steady state sources, such as the filter or the radio static used in Chapter 4, variance was usually less than 2 for the best cluster. For unsteady state source, however, such as the

fountain or the radio-music, variance was commonly as large as 3. Anything larger was usually a phantom peak.

- **Sample Percentage** - [20%, 60%]

If less than 20% of the samples not pointing at a known source point at the detected cluster, then it is most likely reverberation. A real source should dominate the remaining samples in the area.

- **New Cluster Distance** - [1.5-m, 0.5-m]

If there are enough samples, then the source is usually localized to within 0.5-m of its true location. A moving robot, however, as revealed during the investigation testing in Chapter 4, can add a significant amount of noise to the process, especially when the source is far away. If, however, the cluster is re-centered more than 1.5-m (the maximum localization error from Table 4.2) away after adding more samples, it is unlikely that the cluster correctly identifies a new source location and is more likely tracking a phantom peak due to reverberation.

For each of these ranged criteria, we then constructed a linear mapping in the form of:

$$F_i(C_{i,s}) = \begin{cases} 0, & C_{i,s} \text{ worse than } W_i \\ \frac{C_{i,s} - W_i}{B_i - W_i}, & C_{i,s} \text{ better than } W_i, \text{ worse than } B_i \\ 1, & C_{i,s} \text{ better than } B_i \end{cases} \quad \text{Equation 5.1}$$

where  $i$  refers to a particular criterion (cluster variance, sample percentage, new cluster distance),  $F_i$  is the probability of the cluster being a real location given a particular criterion,  $C_{i,s}$  is the value of criterion  $i$  for cluster  $s$ ,  $W_i$  is the worst acceptable value for

that criterion, and  $B_i$  is the upper bound of the range for the  $i^{th}$  criterion (i.e.  $F_i(B_i)=1$ ). The likelihood of a cluster  $s$  being the real location is then:

$$L_s = \prod_i F_i(C_{i,s}) \quad \text{Equation 5-2}$$

The cluster  $s$  with the maximum likelihood ( $L$ ) is the most likely location in the environment for a new source to be present. It is at this location that a robot should center further investigations.

### New Source Localization Results

Before discussing the general results, let us focus first on localizing radio sources in the environment. Since the radios were moved around to different locations, they provide the best comparative performance across environment types. Table 5.3 presents specific results for different radio positions in the environment.

For now, just looking at the performance of the likelihood model on these 30 trials, the location of the source was predicted within 1.5-m of the true value in 70% of the trials. An additional three trials were within 3-m. Although 3-m is a large area, spatial likelihoods can detect sources up to 3-m away, and investigations (see Section 4.2.1) of a potential sound source location by a mobile robot cover regions of 2-m about the suspected source location. Therefore, a robot investigating the suspected source location is very likely to discover the true location, and ultimately localize the source, when the result is within 3-m of the real position. Across all trials, this likelihood model has a 70% chance of placing the robot within 1.5-m of the radio sound source, and an 80% chance of placing it within 3-m of the radio sound source in the environment.

Table 5.3. Illustrates successes in detecting and localizing radios playing music, compared across different environment types. (M) means that the robot did not localize the source within 1.5m, but that future investigations with a range of 3-m should correctly localize the source.

Radio Position	No Active Sources		Fountain Active		Fountain and Filter Active	
	Found?	Distance (m)	Found?	Distance (m)	Found?	Distance (m)
[ 5.2, 0.9]	Y	0.6	M	1.7	N	--
[-0.9, 2.4]	Y	0.6	Y	0.3	Y	0.1
[2.1, 3.4]	Y	0.3	Y	0.2	Y	0.6
[1.2, 1.8]	N	--	Y	0.3	M	2.5
[4.6, 2.4]	Y	0.4	N	4.9	Y	1.1
[4.3, -2.4]	Y	0.3	Y	0.4	N	7.6
[0.9, 3]	Y	0.3	N	4.9	Y	0.4
[2.7, -3]	Y	0.1	Y	0.5	N	5.8
[4.0, -0.6]	Y	0.5	Y	1.5	M	2.5
[-2.4, 2.7]	Y	0.3	Y	0.2	Y	0.8
Success Rate	90%		80%		70%	
Mean Detected Source Error (m)	0.4		0.6		1.1	
Mean Overall Error (m)	0.4		1.5		2.4	

As expected, the performance of the likelihood model decreases with the number of active sources in the environment. With an increased number of sources there is a larger amount of interference from the environment, and therefore, localization should be harder. For trials with no sources active in the environment, the average error is 0.4-m. With one source, the average distance error is 1.5-m, and with two sources, 2.4-m. Presumably, even larger numbers of sources would have an even larger error, which may need to be offset by pausing the robot while sampling, or simply taking longer to patrol the environment.

In addition to the likelihood model used for localizing sources in the environment, we also described in the previous section an algorithm for predicting whether or not a new source even existed in the environment using MFCC's. The idea is that the robot would first estimate whether or not a new sound source was present in the environment, and then use auditory evidence grids to localize the new sound source. That earlier source detection algorithm had an overall accuracy of 92%, but tended towards more false negatives than false positives, making for an accuracy of 87% in correctly classifying environments as containing a new sound source. Unfortunately, the trials where the source detection algorithm failed were not usually the same as the trials where the source localization algorithm failed. And efforts so far to combine the different algorithms have proven unsuccessful. In Table 5.4, the source detection, source localization, and combined accuracy are compared across the types of sources being localized.

Table 5.4. Performance of both the detection (Section 5.2.4) and localization algorithms for environments with at least one new sound source present. Results are compared across the type of source being localized. Note that mean distance only includes those trials where the robot identified at least one location as likely to contain a new sound source.

New Source Type	# of Trials	Mean Error	Accuracy		
			New Source Detection	Source Localization	Combined
Filter	10	0.9-m	100%	100%	100%
Fountain	10	0.7-m	90%	100%	90%
Quiet Filter or Fountain	16	1.1-m	75%	75%	69%
Radio	30	1.4-m	87%	80%	70%
Total	66	1.0-m	86%	85%	76%

From this table, we can see that, similar to the noise detection algorithm, the source localization algorithm works best with constant medium to loud volume sources. This is only natural, as they provide more data from which to be localized. 95% of the filter and fountain sources were not only detected in the environment, but also localized correctly from a single patrol's data.

Radios, which are loud, but not as constant in volume, were also localized relatively well (80%) as they were generally loud enough to be detected over other sounds in the environment. Radios in earlier trials (see Chapter 4), which remained stationary, were generally localized better than in this trial. This time, however, there were more sources located in the environment, and the radio was moved around the room to stress the localization performance. When combined with the new source detection algorithm (Section 5.2.4), however, the combined accuracy suffered significantly. Although each algorithm had an accuracy of 80% or more, there was only a single trial that both algorithms failed, resulting in a combined accuracy of 70%.

Localizing quiet sources resulted in the largest number of failures. Of these failures, only one actually placed the sound source in the wrong location. The others all failed to identify any clusters in the environment besides the known sound sources, hence the reason why the mean localization error is not too high. When the source being localized in the environment is not much louder than the robot, or the reverberant field in general, these are the types of results that we would expect to see. Even though auditory evidence grids are designed to mitigate the masking effects of robot ego-noise, the signal still needs to be loud enough to be heard over other ambient noise in the environment.

## 5.2.6 USING MFCC'S TO DETECT ENVIRONMENTAL CHANGES

The third problem facing a robot trying to maintain a list of active sources in the environment is change to existing sources. While the addition of a new sound source to the environment, and its localization, has already been discussed in the previous two sections, sources that the robot already knows about the environment can change as well. For instance, a sound source can be turned off, in which case it should be removed from the list of active sources. Alternatively, a sound source can simply change its volume or its sound function. Besides wanting to know these changes in order to improve the accuracy of predicted noise models, changes such as these could indicate possible security concerns of their own. A computer with an altered sound function could indicate that someone is illicitly accessing the data on that computer, or suggest that a hardware failure is imminent and that the computer needs to be turned off before data is irreparably lost. Whether there is anything actually wrong, or the source has simply changed, a robot can detect this, updates its models of the environment, and alert a human supervisor to the changes.

The key to detecting any types of changes to the environment is a maintained belief state about the auditory scene. Without any prior knowledge of the auditory scene, even the new source detection and localization algorithms would be impossible, as everything would be new to the robot. In this section, we demonstrate that by adding to this belief state only a little bit more information about the volumes and directivity of those sound sources, the robot can also identify changes to sound sources believed to be active in the environment. Determining exactly what has changed may be rather difficult, but detecting that a change has occurred, and therefore directing the robot to re-

investigate the environment can be done reliably using the information an acoustically-aware robot can gather about the auditory scene.

### Predicting MFCC Classification Results

In Chapter 4, we presented the results of a simple scenario in which the predicted volume difference between sound sources was compared to the measured difference between MFCC classification results. In general, a predicted difference of greater than 2-dB always indicated the correct source by a large margin. For a predicted difference of less than 2-dB, however, the result was more ambiguous. Usually, the louder source was had a higher percentage of samples classified as belonging to its sound function, but there were exceptions. Even though the result is not certain, these results do suggest that the relative volume of a sound source may be directly related to the number of samples classified as belonging to that same source. Therefore, noise maps may be useful in determining change to the environment.

In the following sequence of steps, we use a noise map to do just that, predict the relative number of samples that should be detected as belonging to each class. This information is determined first locally for particular positions along the robot's path through the environment, and then normalized across all of the positions the robot visited while traveling through the environment:

- **Step 1 – Build noise maps**

Build a noise map of the direct field for each of the known sources in the environment.

- **Step 2 – Identify volume differences between sources**

Calculate the difference in volume ( $\Delta V_{i,j}$ ) between all sources ( $i,j$ ) for location  $(x,y)$ .

$$\Delta V_{i,j}(x, y) = V_j(x, y) - V_i(x, y) \quad \text{Equation 5.3}$$

- **Step 3 – Estimate the probability of being detected.**

For each sampled location, estimate the probability of this source ( $i$ ) being heard over every other source ( $j$ ) in the environment ( $C_{i,j}$ )

$$C_{i,j}(x, y) = \begin{cases} 0, & \Delta V_{i,j}(x, y) > 3dB \\ 0.5 - \frac{\Delta V_{i,j}(x, y)}{2 * (3dB)}, & -3dB < \Delta V_{i,j}(x, y) \leq 3dB \\ 1, & \Delta V_{i,j}(x, y) \leq -3dB \end{cases} \quad \text{Eq. 5.4}$$

This simple linear scale relates the chances of being heard over another source to the volume difference between sources. If the difference in sources is greater than 3-dB, then it is assumed that the quieter source will not be heard at all over the louder source. If, however, the difference is less than 3-dB, then the chance of being detected varies linearly with the difference in volume.

- **Step 4 – Combine probabilities across all sources.**

Assuming that the probability of each source being heard over another source ( $C_{i,j}$ ) is independent, the probability of a single source being detected across all other sources at a given location is estimated by using multiplication:

$$W_i(x, y) = \prod_j C_{i,j}(x, y) \quad \text{Equation 5.5}$$

- **Step 5 – Combine probabilities across all locations**

To take the local information from each sample position, and build a global likelihood estimate, we will sum the results and normalize. First, for each source, sum up the chances of being detected across all sampled locations in the environment  $(x,y)$

$$T_i = \sum_x \sum_y W_i(x, y) \quad \text{Equation 5.6}$$

- **Step 6 – Normalize.**

Normalize the results across all sources (and all positions) to estimate the percentage of samples that belong to each active sound function in the environment if nothing has changed:

$$P_i = \frac{T_i}{\sum_j T_j} \quad \text{Equation 5.7}$$

This sequence of steps is designed to estimate what the robot should have measured if the environment was unchanged. It is based on the hypothesis that volume differences between sound sources, as predicted by the sound propagation framework, are linearly related to the probability of classifying a recorded sample as belonging to a particular source. The louder the sound source at a particular location, the more likely that the resulting MFCC classification vector should be closest to that sound source's function than any other in the environment. The louder the sound source is overall, the larger the percentage of samples that should be classified as belonging to that sound source across the entire route traveled by the robot. The choice of linear relationship is due to the fact that the pressure of the sound wave, the unit measured by the microphone, decays linearly with the distance from the sound source.

### Determining Change

After finishing the patrol route, classifying all of the data, and making a prediction of what the robot should have heard if nothing had changed, we have two values for each known source in the environment: (1) the predicted percentage of samples ( $P_i$ ) classified as belonging to source  $i$ , and (2) the measured percentage of samples ( $M_i$ ) classified as belonging to source  $i$ . To compare the predicted and measured results, we use a form of Bayesian updating [Thrun 2002].

$$L_i = \left| \frac{P_i - M_i}{P_i} \right|$$
$$LC = \prod_i \frac{L_i}{1 - L_i}$$

Equation 5.8

Where  $L_i$  is the likelihood of any given source's distribution having changed, range restricted to [0.01, 0.9], and  $LC$  is the resulting combined likelihood of one or more sources having changed in the environment. Note that the choice of range restrictions was made to prevent weak sources from dominating the equation. As weak sources can change significantly, often not appearing in the data at all, their associated likelihood of change can quickly approach 100%. While we need to include these effects in the calculation, it is important to make certain that a source that was predicted to occur in only 5% of the samples does not dominate the result.

### Results – Did the Environment Change?

In these trials, we are looking only at situations where either the sources are the same, a source (filter or fountain) has been turned off, or a source has been turned down

in volume. Given the prediction and comparison measures from the previous trials, how well could the robot identify that the environment had changed?

There were a total of 37 trials that could be used for this work, 4 of which contained no active source, 18 of which contained only 1 active source, and 15 of which contained 2 active sources. For evaluation purposes, some of these trials were used with multiple belief states to test for different types of changes. Table 5.5 lists the specific trials tested, and the different belief states that could be used with each.

- **No Active Source**

Used with three different belief states: (1) filter active, (2) fountain active, and (3) filter and fountain active. In all cases, the result should be that no source is active.

Table 5.5. Summary of belief states used for each patrol run through the environment.

Actual State	# of Trials	Belief States		
		Filter	Fountain	Filter + Fountain
No Sources	4	X	X	X
Filter Active	5	X		X
Quiet Filter	4	X		X
Fountain Active	5		X	X
Quiet Fountain	4		X	X
Filter and Fountain	5	X	X	X
Quiet Filter + Fountain	4			X
Filter + Quiet Fountain	4			X
Quiet Filter + Quiet Fountain	2			X

- **Filter Active**

Used with two different belief states: (1) filter active, and (2) filter and fountain active. In the first case, there should be no detected change to the environment. In the second case, it should be detected that the fountain has changed.

- **Quiet Filter Active**

Used with two different belief states: (1) filter active, and (2) filter and fountain active. In the first case, it should be detected that the filter has changed. In the second case, both sources have changed, but the fountain should have changed the most.

- **Fountain Active**

Used with two different belief states: (1) fountain active, and (2) filter and fountain active. In the first case, there should be no detected change to the environment. In the second case, it should be detected that the filter has changed.

- **Quiet Fountain Active**

Used with two different belief states: (1) fountain active, and (2) filter and fountain active. In the first case, it should be detected that the fountain has changed. In the second case, both sources have changed, but the filter should have changed the most.

- **Filter + Fountain**

Used with one belief state, filter and fountain active. No changes should be detected.

- **Quiet Filter + Fountain**

Used with one belief state, filter and fountain active. The filter should be detected as having changed.

- **Filter + Quiet Fountain**

Used with one belief state, filter and fountain active. The fountain should be detected as having changed.

- **Quiet Filter + Quiet Fountain**

Used with one belief state, filter and fountain active. Which source has changed the most is uncertain.

In Table 5.6, the likelihood of a change being present in the environment (LC), as determined by Equation 5.8, is compared to the number of actual changes in the environment. In theory, assuming that the MFCC classification results can be related to the predicted direct field volumes (as suggested by Section 4.2.4), and the proposed algorithm for predicting the MFCC classification results (Equations 5.3-5.7) is correct,

Table 5.6. Results of the source change detection algorithm, compared across different numbers of changes in the environment.

	All Trials			Discarding Outliers		
	# of Trials	Mean Likelihood of Change	Standard Deviation	# of Trials	Mean Likelihood of Change	Standard Deviation
No Change	15	0.31	0.37	11	0.11	0.12
1 Change	28	0.64	0.33	24	0.61	0.33
2 Changes	15	0.97	0.05	13	0.97	0.05

then the likelihood of change value should increase with the number of changes in the environment.

According to this table the mean likelihood of change does increase significantly with each additional source having changed in the environment. Unfortunately, the standard deviation is particularly high, meaning that, while there still appears to be a trend, it is difficult to identify changes due to the wide range of variation.

Taking a closer look at the scenarios, however, allows us to identify and discard some outliers. In particular, 8 trials had results where the percentage of samples classified as being dominated by robot ego-noise was significantly less than some other source (<1.5 times another source). Given the close proximity of the robot motors and wheels to the microphones mounted on its back, ambient noise sources that are only 10-15 dB louder than the robot, and a route through the environment that gets the robot away from active sources for significant periods of time, the robot should be one of the most detected sources in the environment. When it is not, this suggests that either another environmental source (besides the tested sources) is interfering with data collection, or that the robot's own sound function has changed. The latter case is particularly suspect, because different types of movement generate wheel noise for which no sound function model was available. Furthermore, additional fans on the robot could turn on and off to discard excess heat, changing the sound function in the process.

By discarding these 8 faulty trials from the results (2 Filter, 2 Filter + Fountain, 1 Quiet Fountain, 1 Quiet Filter, and 2 Filter + Quiet Fountain), we can see a large drop in both the mean and standard deviation of the unchanged environment condition. Now there is a significant enough separation between categories of change to identify, with

some certainty, whether or not something in the environment is different from what the robot believes.

### Discussion – Identifying What Changed

The previous results indicate that there is a significant separation between no-change, 1-change, and 2-change, therefore a robot should be able to identify that the auditory scene has changed somehow. That alone is enough to alert a security guard to the altered situation, but is not very significant if the change is relatively minor. For instance, did the filter (think HVAC) simply turn itself off? If so, then the robot should ideally just update its current belief state to reflect any changes to the environment and proceed on to the next task.

Unfortunately, the determination of exactly what changed is not completely straightforward. Below we discuss three different approaches, two of which require further information to determine what changed:

- **Using Likelihood of Change**

The obvious strategy for determining source change would be to use the likelihoods calculated in the previous section ( $L_i$ , Eq 5-8), taking the source with the highest percentage change as the most likely source. For this series of testing, this strategy is actually accurate 78% of the time for environments with only 1 changed (quieted or disabled) source. Environments with changes to 2 sources (10 trials, Filter+Fountain belief state), however, only successfully predict which source was disabled in 50% of the trials. Unfortunately, this approach to determining what

changed is naïve. The likelihoods calculated earlier do not reflect the delta change of percentage, but rather they estimate the delta volume change. Therefore, when a source decreases by some delta percent, the other sources in the environment absorb that change, increasing their own relative percentage of the samples, and vice versa. For example, let us assume that the collected samples for a normal environment contain 10% filter, 40% fountain, and 50% robot. Now let us assume that the fountain decreases in volume substantially. A plausible new distribution might be 30% filter, 15% fountain, and 55% robot. Using our predictive model, the delta change for the fountain is 63%, but the delta change for the filter is 200%. Which one changed? In truth, our likelihood model is measuring, not the likelihood of the source changing, but rather, the likelihood of the distribution having changed.

- **Informed of environmental volume change**

If we know how the volume of the environment changed (i.e. did it go up or down), then the tools presented earlier may still be useful for determining which source changed. For instance, if the environment is known to have become quieter, then we can look at how each source has changed, separating sources that went up in measured classification percentage from those that went down. Only sources that become quieter should be considered for further investigation.

- **With Known Source Models**

The third approach to determining what changed does not require the previous likelihood estimates. Instead, it requires knowing something about the source functions for active sound sources in the environment. In this scenario, let us assume that our active sources in the environment are only capable of being turned on/off. Now, auditory evidence grids can be used to search for disabled sources in the environment. By building an evidence grid centered on each source location, and then using the criteria specified previously for localizing “new” sources, a robot can identify inactive sources as sources where there is a 0% chance of there being a sound source at the proper location. Looking at just those trials with one active source, including those where the active source was quiet, patrol data correctly indicated the disabled source in 87% of the trials (14 out of 16). In the two trials where this method failed, it actually suggested that both sources had been disabled, when in fact the second source had only been quieted. This confusion could be resolved by further investigation.

In general, the problem of identifying exactly what changed in the environment is still an open question, as it requires more knowledge of how the environment has changed. Did the environment get louder/softer? Was one of the ambient noise sources turned off? Theoretically, the robot should be able to answer these questions using sound pressure level measurements of the reverberant field, or MFCC classification results that fail to detect a missing source. In practice, however, the accuracy required for either of these operations is not yet available on the robot. Localization error, combined with

changes in robotic movement, causes the average volume of a patrol route for the same environment to vary by several decibels. Similarly, classification using MFCCs can sometimes classify small numbers of samples as belonging to a disabled source. Therefore, using noise maps, a robot can identify that something has changed, but without further knowledge or effort by the robot, it cannot yet reliably identify what changed.

### **5.2.7 SUMMARY OF ACOUSTIC MONITORING PERFORMANCE**

The focus of this work in monitoring the environment was to identify the likelihood of specific changes that may have happened to the environment. After completing a patrol route, the robot can use its knowledge of what the environment sounded like in previous runs to recognize when known sources have changed, new sources are enabled, and where new sources might be located. Furthermore, the accuracy for each of these actions runs 80-90% for a variety of different sources common to indoor environments. This is good for a first pass through the environment. Given that many of these failure cases are likely due to robotic error, as demonstrated in Section 5.2.6, a second pass through the environment is likely to improve the accuracy rate even further.

There are at least two improvements, however, that have yet to worked into this algorithmic solution, and which may improve overall accuracy. The first such improvement is combining the algorithms together into a single cohesive tool for recognizing, and categorizing change. As of now, MFCCs are generally used for detecting that something has changed across the entire range of the robot's patrol route, be it a new source or a change to an existing source. Auditory evidence grids, in contrast,

are used for location dependent searches, including localizing new sound sources in the environment, and determining whether a sound source at a specific location is still active. Still, there is overlap in the tasks, as demonstrated by having to first detect that a new source is enabled before localizing it, or recognizing that something in the environment has changed before determining which source was enabled. Furthermore, results from Section 5.2.2 (new source localization) also suggest that the different tools have different strengths and weaknesses, having similar accuracy, but failing on different trials. Therefore, future work should concentrate on combining these tools probabilistically to improve accuracy in monitoring the environment.

The second improvement is likely related to the first. It is the inclusion of real-time data analysis and dynamic path planning. In the current implementation, the robot finishes patrolling the environment before analyzing the collected data and deciding on its next action. This implementation, however, has some significant disadvantages. The first such disadvantage is reaction time. If the change to the environment is time sensitive (i.e. likely to disappear, or otherwise needing immediate action), then processing the data after completing the patrol could be too late. Another problem is it may be difficult to expand the current implementation out to significantly larger areas. With a significantly larger environment, both the new source localization (Section 5.2.5) and change detection (Section 5.2.6) algorithms, which compare data across the entire patrol run, would require significantly larger changes to the environment in order to work. If, however, the robot was processing data in real-time (which may require the higher accuracy of a combined MFCC and auditory evidence grid), then it could detect that some area of the environment was suspicious while it was still in the area, and

dynamically readjust its movement to investigate that area. For example, if data collected thus far indicates that a source may have changed, the robot can slow down its movement to collect more data, or actively investigate the source. Similarly, unexpected regions of “new” sound samples can be at least partially investigated, so as to better discard them later if better data should occur. In general, working with local subsets of the data in real-time should allow the robot to accurately monitor larger areas, and respond quicker to changes in the auditory scene.

### **5.3 IMPROVING THE SIGNAL-TO-NOISE RATIO**

The previous work demonstrated some significant advantages in using knowledge of sound flow to monitor the auditory scene. What it lacked, however, was an influence over the robot’s navigation while patrolling the environment. If the robot is seeking a particular type of noise in the environment, then maybe it should avoid known, predictable sources of sound that will only mask the signal it is searching for? Alternatively, if the robot detects something odd about the area through which it is traveling, the robot could slow down and/or change its movement pattern to gather more data in the vicinity. Being able to make high-level decisions about further investigations after completing the patrol is important, but so is adapting performance while the robot is gathering data.

In this section, we will try to improve on the surveillance problem by limiting the exposure of the robot to ambient noise, thereby increasing its signal-to-noise (SNR) ratio. We know from the acoustic monitoring scenario (Section 5.2) that a robot is capable of monitoring the auditory scene with some degree of reliability. The robot can determine

when existing sources have been activated, deactivated, or changed. The robot can also detect new sources, and investigate them using the suite of tools provided in Chapter 4. Using its knowledge of the current auditory scene, the next step is to further enhance the listening capabilities of the robot by avoiding those known noise sources in the first place, improving its chances of detecting shorter duration noises from the ambient that are of great importance to a surveillance operation.

The experiments performed in this section can be divided into two parts. In the first part, originally presented at Human-Robot Interaction 2007 [Martinson and Brock 2007], the robot is placed in an initially poor acoustic location and tasked with improving its signal-to-noise ratio using either a noise map or a reactive avoidance behavior. In the second part, presented in Section 5.3.2, the robot is performing the patrol mission described in the acoustic monitoring task (Section 5.2.2), only adapting its route to limit its exposure to ambient noise. Both of the experiments were originally proposed in the noise mapping paper presented at Mobile Robots XVII [Martinson and Arkin 2004]. The preliminary results presented in that paper, however, have since been explored in more depth with different robotic hardware and different environments.

### **5.3.1 CORRECTING FOR A POOR INITIAL ACOUSTIC LOCATION**

While the patrol scenario discussed in Section 5.2.2 uses a robot that is constantly moving through the environment, there is also a need for a less active observational approach in many robot scenarios. Sometimes the robot is designed to gather information over time rather than space, waiting for long periods in one location for something to happen. For instance, continuing with the general theme of robot assisted security, let's

say the police are trying to counter a string of break-ins over a well-to-do neighborhood. They deploy a number of small robots throughout the affected area to observe the environment and report back any problems to on-duty officers. These robots, unlike those used for patrolling a building or other large area, are designed to remain largely where they have been deployed. However, since the auditory scene may change significantly from the initial time of deployment (air conditioners may activate, people may be having parties, sprinklers/fountains may be running, etc.), the robots are still a significant improvement over a fully stationary sensor. When the auditory scene changes non-threateningly, a robotic sensor cannot only move away from the source, but also predict where to move so as to reduce its ambient noise exposure and more effectively monitor the surrounding environment.

### An Avoidance Response

In response to a changing auditory scene, there are at least two types of actions that a robot aware of sound flow through the environment can take. The first such action is a simple avoidance reaction. Since loudness diminishes with distance from a source, the robot can decrease its exposure to a source of ambient noise by moving as far away from it as possible, while remaining within a specified area. This can be done easiest in a reactive fashion by measuring the direction of maximum ambient noise energy on the robot, and moving in the opposite direction. Work by Barbara Webb [Webb 1998] did something similar, except that instead of avoiding the noise, her phonotaxis behaviors moved the robots towards the sound. The drawback to a purely reactive approach is, of course, local minima. Obstacles in the surrounding environment can prevent the robot

from moving far enough away from the sound source, even if there are better locations from which the robot can listen to the ambient noise on the other side of the obstacle. Therefore, since our robots have maps of the environment available to them already, in addition to simple source localization tools, we expanded this approach to include the use of an obstacle map and path planning. The algorithm for the avoidance response is as follows:

- **Step 1 – Identify source direction**

Using spatial likelihoods, determine the most likely angle to the sound source. Appendix B.1 has more details on identifying the best angle from a spatial likelihood.

- **Step 2 – Localize the source**

Assuming an initial source position 1-m from the robot, move for a short distance while sampling tangentially to the source. Create an auditory evidence grid from the samples to actually localize the sound source

- **Step 3 – Identify a better location**

Making use of the obstacle map, the robot identifies the set of reachable locations, and picks the farthest reachable location away from the estimated source position. Appendix C.1 describes in more detail how to pick a reachable location from an obstacle map of the environment.

- **Step 4 – Move the robot**

Move the robot to that location using a path-planning algorithm.

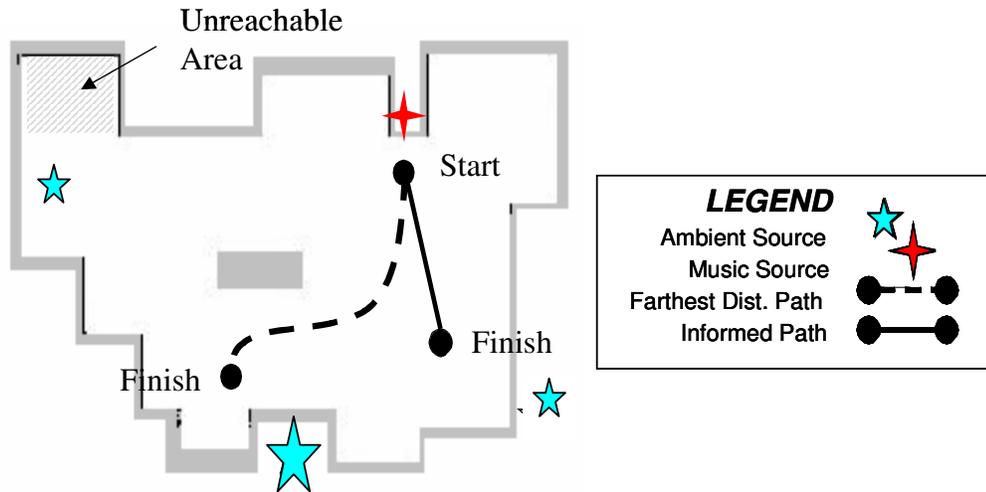


Figure 5.4. Graphical comparison of different relocation strategies the robot can use to avoid a sound source when correcting for a poor initial acoustic location.

It can simply move as far away as possible (dashed line), or it can take into account ambient noise sources (5-pointed stars) when picking a new location. The latter choice has the robot avoiding the largest of the ambient noise sources at the bottom of the map.

In Figure 5.4, the dashed line shows how the robot moves to avoid the sound source (red 4-pointed star) by picking the farthest reachable location, and moving around obstacles to reach its goal.

### An Informed Response

The trouble with the farthest-distance-removed approach is that there might be additional sound sources on the other side of the room opposite the newly interfering sound source. Therefore, by simply picking the farthest location away from the present sound source the robot may not actually be decreasing its noise exposure. Using the tools in Chapter 4, however, an acoustically-aware robot might already have knowledge of these other ambient noise sources in the environment. Instead of simply reacting to the one newly detected source, an alternative, knowledge-based response makes use of the

mathematical framework proposed in Chapter 3. Using the information the robot already knows about the environment and the sound sources within it, the robot can predict what its ambient noise exposure should be at any location in the environment due to the combined effects of all known sound sources. The noise map it creates now provides a guide from which the robot can pick the quietest remaining location in the environment, and move to that goal using a path-planning algorithm.

The algorithm for this informed response to changes in the auditory scene is as follows:

- **Step 1 – Estimate volume**

Measure the volume of the source from the current location, averaging the sampled data results over 10-sec. Appendix B.7 describes how to estimate the sound pressure level from a single sample.

- **Step 2 – Identify source direction**

Use spatial likelihood results to determine the most likely angle to the sound source.

- **Step 3 – Localize the source**

Assuming an initial source position 1-m from the robot, move for a short distance while sampling tangentially to the source. Create an auditory evidence grid from the samples to actually localize the sound source

- **Step 3 – Map the noise**

Create a noise map of the environment using the positions and estimated volumes of all known sound sources, including the sound source just detected and measured. In this work, all of the sound sources used a

simplified omni-directional source model to estimate sound flow, and only the direct field was estimated. Appendix B.5 has more detail on creating maps of the direct field.

- **Step 3 – Identify a better location**

Making use of the obstacle map, the robot identifies the set of reachable locations. Then, instead of picking the farthest reachable location away from the estimated source position, the robot uses its predicted noise map to pick the quietest location.

- **Step 4 – Move the robot**

Move the robot to that location using a path-planning algorithm.

In Figure 5.4, the solid line demonstrates the difference between the path of the robot using this informed algorithm versus the farthest distance response discussed previously. Where the previous algorithm places the robot relatively close to a known ambient noise source (blue 5-pointed star), the informed approach to relocating the sensor places the robot in the middle of the room where it is least affected by the direct fields of any sound source, and is subject primarily to only reverberant effects.

## Results

The testing of these different avoidance strategies was performed at the Navy Center for Applied Research in Artificial Intelligence, on the Naval Research Laboratory campus. The robot used for these tests was the B-21r used in much of the auditory evidence grid testing described in Chapter 4.

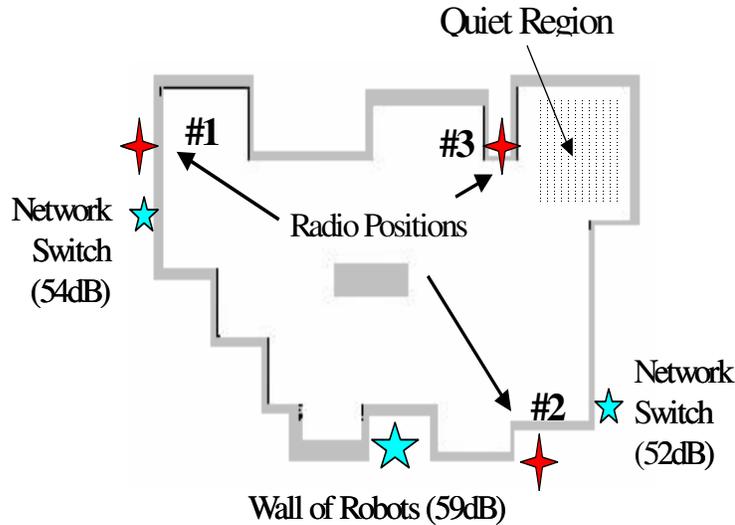


Figure 5.5. Positions of the 3 different ambient noise sources and radios within the testing environment for improving a poor initial acoustic location. The patterned region in the upper right corner indicates the area that the robot moves to while avoiding sources 1 and 2.

The auditory scene affecting the robot in these tests can be seen in Figure 5.5. Three ambient noise sources, represented by blue 5-pointed stars were always active in the surrounding environment. The ambient noise source at the bottom of the lab was the loudest, being caused by 10+ robots idling up against the wall. Their combined effects were 59dB of pink fan noise. The other two significant ambient noise sources were network switches with internal fans generating 52 and 54 dB of noise. The ideal omnidirectional noise map created from these three sources is shown in Figure 5.6.

The other three objects in the auditory scene (red 4-pointed stars) were music sources, each averaging 60-65dB over the course of the music played. For a single test, the robot would start near a single enabled music source (the other two music sources would be off), detect that source, and then move to another location using one of the two

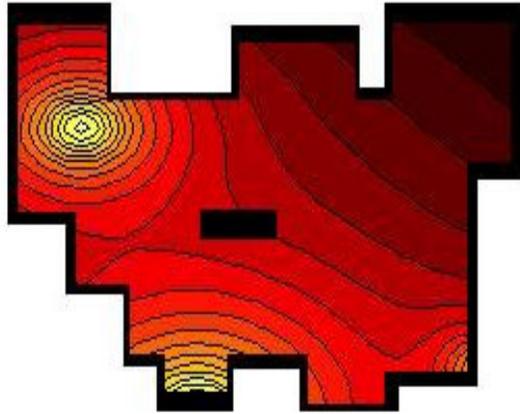


Figure 5.6. Predicted noise map of the poor initial acoustic location testing area modeling the effects of the three ambient noise sources on the auditory scene. This map assumes that each source is omni-directional.

Table 5.7. Average reduction in noise levels using different relocation strategies to avoid music sources

	Farthest Distance	Informed Decision
Source 1	-9dB	-10dB
Source 2	-13dB	-12dB
Source 3	-2dB	-5dB

possible relocation strategies. This test was repeated five times for each of the music sources and relocation strategies, for a total of 30 runs. Table 5.7 shows the average improvement in the auditory scene while avoiding each of the three sources.

The results demonstrate the benefit of avoiding sources and the difference in relocation strategies. To avoid the first two music sources, both relocation strategies led to similar final positions in the quietest part of the room (the upper right in Figure 5-7), resulting in comparable performance improvements. The relocation strategies selected notably different final positions, though, to avoid the third music source, which was located in the quiet area both strategies selected to avoid the first two music sources (again, the upper right in Figure 5.5). With that part of the room now filled with noise, the robot was not able to demonstrate as much of a drop in average noise levels as the other scenarios. However, the farthest distance strategy sent the robot to one of the fan sources where noise levels were only slightly less than the original position (the dashed-line path in Figure 5.4). The informed decision strategy resulted in a location closer to the middle of the room where the robot the robot had 3-dB advantage over the uninformed relocation strategy.

#### Discussion of Relocation Effectiveness

The goal of the relocation work was to test the effectiveness of different strategies for improving the signal-to-noise ratio recorded by the robot. As expected, the most informed strategy for relocating the robot was the most effective overall. If the robot knows about all of the significant sources of sound in the environment, then it can avoid selecting and moving into areas of loud ambient noise that the farthest-distance away

approach might fall into. Furthermore, although not tested here, a well-informed path-planning algorithm will have similar advantages over a purely reactive strategy. Although simply moving away from the loudest direction might also avoid other ambient noise sources, the reactive strategy could fall into areas of local minima when an obstacle, or other ambient noise source blocks its path to the quieter areas of the room.

In general, however, the improvement of the informed path-planning algorithm over the less informed, but possibly simpler alternatives, is going to vary dramatically between environments. As can be seen from the first two source results, the farthest-distance away approach could still demonstrate a 10-dB or better decrease in ambient noise when the robot did not end up next to a loud ambient noise source. If the environment being observed by the robot is relatively benign acoustically (i.e., few significant ambient noise sources), then the chances of the robot ending up in another poor acoustic location after relocating are small. So if this scarcity of sources is known prior to the robots deployment, then the designer may not want to worry about tracking existing sources in the environment.

For an unknown environment, however, there is yet another option to those previously discussed, an uninformed, but knowledge-based response. In the tests already completed in this section, it was assumed that for the informed path selection, the robot had already explored the environment and identified active ambient noise sources that it should avoid in the future. If, however, the robot does not have the time or power to explore the environment ahead of time, the framework will still work with the partial information available to it. For the first relocation, it should perform comparably to the farthest-distance away metric. If the robot ends up in yet another poor acoustic location,

however, then a new source can be added to the list of known sources and the robot can identify yet another location to try and improve its SNR, repeating as necessary. The final result is comparable to an avoid past behavior [Balch 1993], only it incorporates the nature of sound propagation into the algorithm.

### **5.3.2 IMPROVING SNR WHILE PATROLLING THE ENVIRONMENT**

The second set of experiments in improving the signal-to-noise ratio evaluated the performance of a moving robot. We know from the previous section (Section 5.3.1) that the robot is capable of improving a singular position by being acoustically-aware, but, as demonstrated from the earlier patrol scenario, many applications may not involve the robot stopping to listen to the environment, at least not at first. Patrolling the environment, for instance, often requires that the robot complete its patrol in a certain amount of time. The robot can certainly stop and investigate once in awhile, but stopping every few meters to listen to the soundscape may take too long. Therefore, our second round of testing was designed to demonstrate an improvement in SNR while the robot was constantly moving throughout the environment. This is significant because while moving around a mapped sound source, the robot could actually introduce more noise than what is gained by avoiding the source. The new noise could be excessive wheel noise generated by the robot following a gradient, or other noise not accurately represented on the map.

For this second round of SNR testing, we will be using the same obstacle map as for the acoustic monitoring task. Seen in Figure 5.7, a radio is located on the obstacle in the middle room, generating static noise in a cardioid pattern to the left at 67dBA. The

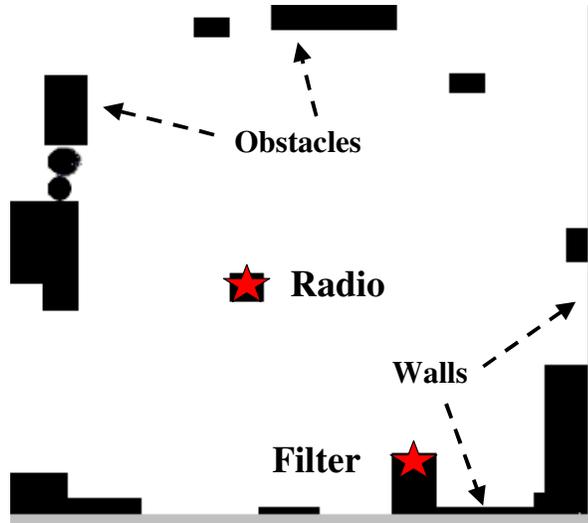


Figure 5.7. Environment for testing the improved SNR movement strategies. This is the same environment as the acoustic monitoring task, only a radio generating static noise replaces the fountain in the middle of the room.

filter used in the acoustic monitoring task is located in the same place as before, generating noise at 55 dbA. In the base case, the robot follows the same patrol route as used in the acoustic monitoring task, while in the noise avoiding case, the patrol route is revised to avoid areas of excessive noise.

### Revised Patrol Task

For the acoustic monitoring task, the purpose of patrolling the environment was to identify new sound sources. Therefore, when picking a patrol route through the environment, the most important characteristic of the chosen route was that the robot passed close enough to all areas of the environment likely to contain a new sound source. In this new set of experiments, we are trying to improve the signal-to-noise ratio of the robot by changing the path that the robot follows through the environment. So that any

performance improvements can be integrated back into the acoustic monitoring task, the basic algorithm for selecting a patrol path through the environment that was presented in Section 5.2.1 is also used in these scenarios, only adapted to take ambient noise levels into account when choosing a path through the environment. The adapted algorithm is as follows (details appear in Appendix C.2):

- **Step 1 - Use the obstacle map to identify areas reachable by the robot.**
- **Step 2 – Divide the reachable area of the map into discrete grid-cells.**
- **Step 3 - Pick a target within each grid cell.**

This is where the first difference between the original algorithm and the adaptive algorithm occurs. Instead of picking a location closest to the center, the robot uses its noise map to pick the location with the lowest expected ambient noise within each grid cell. In the event of a tie, the robot selects the location closest to the center of the grid cell. This aware strategy is described in more detail in Appendix C.2.1.

- **Step 4 - Find the quietest circular patrol route through all targets.**

In the acoustic monitoring task, the ordering of the waypoints was chosen to minimize the distance traveled by the robot. In this case, however, we are trying to minimize sound exposure. The same traveling salesman heuristic can be used to solve this problem, only now the cost function being minimized has changed. Instead of the cost of traversing an edge  $(m,n)$  being the distance between waypoints  $m$  and  $n$ , the cost of traversing the edge is equal to the sum of the noise at all locations  $(\chi)$  between  $m$  and  $n$ :

$$W_{mn} = \sum_{m < \chi < n} \text{Noise}(\chi) \quad \text{Equation 5.9.}$$

The noise levels in this equation would be retrieved directly from the noise map predicting current ambient noise levels in the environment. Since the units of this map are actually in decibels, this weighting function is not attempting to calculate any type of average. Instead, the weighting function is designed to emphasize shorter path lengths. When two lengths are comparable, however, it would be best for the robot to take a longer path if it is less noisy. Appendix C.2.3 describes in more detail how to adapt the travelling salesman heuristic from Section 5.2.2 to the noise minimization problem.

In Figure 5.8, a set of waypoints selected using the given noise map are compared to waypoints located at the grid-cell centers. The flexible algorithm for waypoint selection allows the robot to avoid particularly loud areas of the environment, by sampling at the edges of the grid cell instead of in the middle where the sound levels may be significantly higher. In the following experiments, the grid cell size is 1.8-m (the same as was used in Section 5.1), so the diagonal length is 2.4-m, well less than the estimated maximum distance of 3-m used for localizing sources with spatial likelihoods (see Section 4.2 for more detail).

### Gradient Following Behavior

A second method for improving the SNR recorded by the robot is to follow a gradient through the environment. Originally suggested in [Martinson and Arkin 2004], a gradient following behavior allows a robot to adapt to local areas of loud noise between

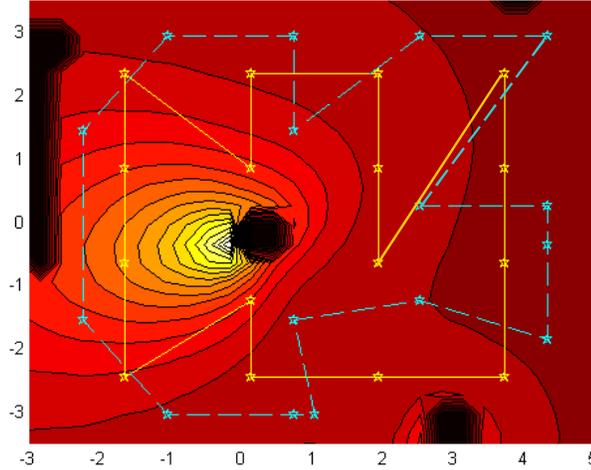


Figure 5.8. Paths taken by the different movement strategies overlaid on the robot-discovered noise map: (solid) the path through the grid-cell centers, (dashed) the path chosen to avoid loud locations.

waypoints. The method our acoustically-aware robot uses to avoid ambient noise in the environment is based on the potential fields approach to robot control. Our noise maps, either predicted or sampled, indicate regions of high volume noise with greater numerical scores. Taking the gradient of the noise volume ( $N$ ) in both the  $x$  and  $y$  dimension, we can easily build a vector field representation of the noise levels by converting these gradients to polar coordinates, indicating the best strength ( $str$ ) and direction ( $dir$ ), for the robot to move to avoid noisy regions (Equation 5-10). In Figure 5.9, a noise map representation made with hand-sampled data is converted to a potential field using this gradient approach.

$$str = \sqrt{\left(\frac{\partial N}{\partial x}\right)^2 + \left(\frac{\partial N}{\partial y}\right)^2}$$

$$dir = \pi + \tan^{-1}\left(\left(\frac{\partial N}{\partial y}\right) / \left(\frac{\partial N}{\partial x}\right)\right)$$

Equation 5.10

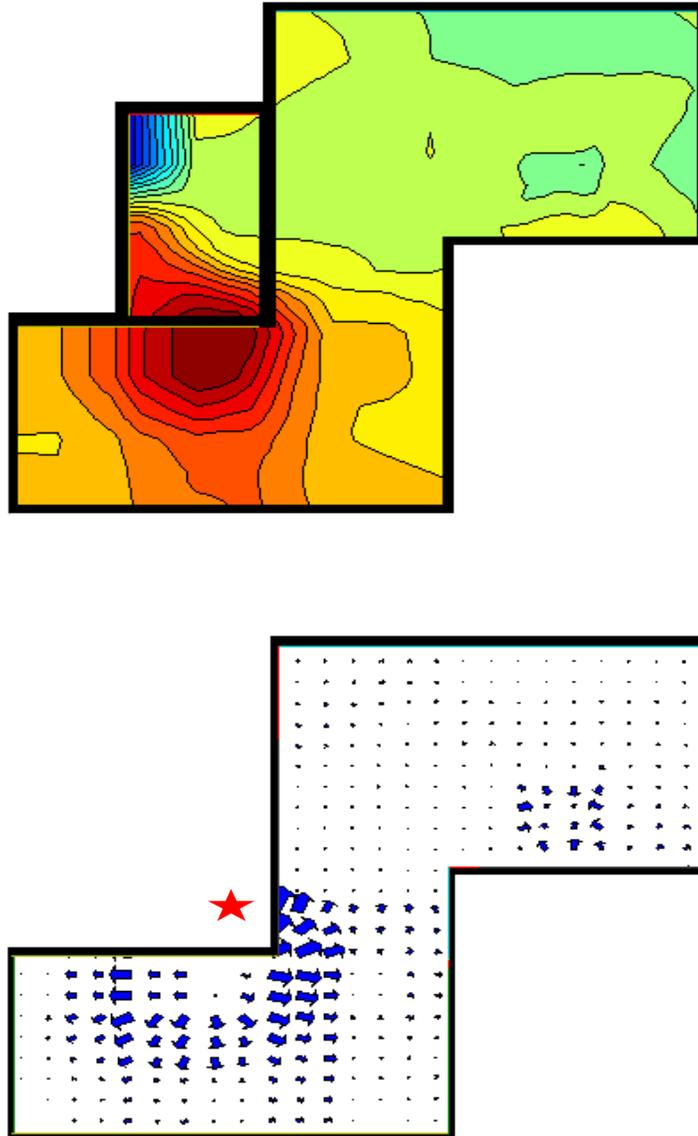


Figure 5.9. A noise map created from hand collected samples (left) is converted to a vector field representation, where strength is indicated by arrow size.

Although the noise map suggests the best course for the robot to follow to avoid noisy areas in the environment, it would, by itself, cause the robot to move to the nearest local minima (quiet location) and stop. For that reason, this gradient following behavior needs to be combined with other behaviors for following the waypoint path through the environment, and avoiding local minima. Our implementation uses vector summation [Arkin 1998] to combine the different behaviors (seen in Figure 5-12). Each of the associated behaviors is also described below:

- **Follow Noise Map**

This function uses the robots own estimated position in the environment to determine the size and direction of a repulsive force using a gradient field created from a noise map of the environment. The end result forces the robot to move around areas predicted to contain particularly loud ambient noise.

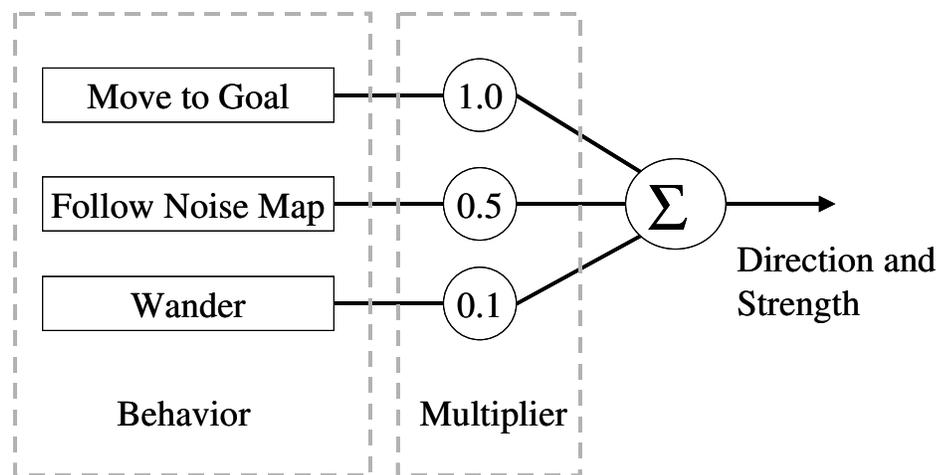


Figure 5.10. The behavioral controller used to reactively follow gradients along a waypoint path. Individual behaviors produce vectors (strength and direction of movement) that are multiplied by some weight and summed together.

- **Move To Goal**

Generates an attractive force towards a target waypoint in the environment [Arkin 1998] using the known position of the robot relative to the target. Whenever a waypoint is reached, then the robot selects the next waypoint in the path list as its new target.

- **Wander**

Generates a unit vector in a random direction every turn. This behavior is designed to get the robot out of local minima created when the other behaviors conflict by pushing the robot equally in opposite directions. The addition of a random force can help the robot get out of these stall points.

A fourth behavior, avoid-obstacles, was included in the original work to reactively guide the robot away from detected obstacles in the environment. In these experiments, however, we use a different controller provided by the Player/Stage environment. This environment has an obstacle avoidance method based on vector field histograms [Borenstein and Koren 1989] built into the software controller, so this potential fields based method for obstacle avoidance is not needed.

### Results - Patrolling The Environment

In this second round of SNR testing, we are particularly interested in the performance of the advanced methods for sound propagation modeling discussed in Chapters 3 and 4. How does the robot-measured data compare to the hand-measured

data? What is the quality of the reverberant field estimates from robot created obstacle maps?

To examine these questions, we tested 3 different types of awareness in the patrol scenario:

- **Unaware**

This is the base case, in which the robot does not have a noise map to help guide it through the environment. Without knowledge of the surrounding ambient noise, the robot moves from waypoint to waypoint along the shortest path without following any gradients.

- **A Priori Information – Direct Field Only**

For this second type of awareness, the robot is provided with hand-measured information about the location, volume and directivity of both sources (radio and filter) in the environment. The robot then predicts the levels of ambient noise in the room, using just direct field calculations. This noise map is shown in Figure 5.11.

- **Robot Discovered Information – Direct Field Only**

For this stage, the sound source location, volume, and directivity were not provided to the robot before hand. The robot first had to patrol the environment without any knowledge, and investigate each of the discovered sources using the area-coverage heuristic discussed in Section 4.4. Then using this discovered sound source information, the robot predicted the levels of noise due to the direct field only, and used that information to patrol the environment. Given that the robot was using its

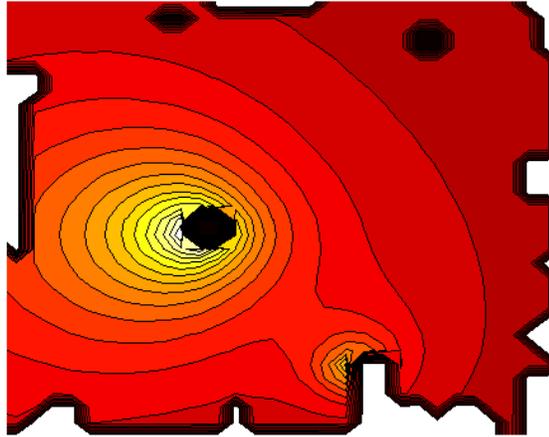


Figure 5.11. Direct field map created from hand-measured data used in testing the improved SNR movement strategies.

own investigatory abilities to determine the current state of the environment, the ideal scenario would have the robot reinvestigate the scene before every patrol. However, as that would be very time consuming, only two investigations of the scene were completed. Half of the patrols were then run using each of the resulting scenes and the results averaged together. The two noise maps created from robot investigations that were used to guide the robot in this stage are shown in Figure 5.12.

### Results – Patrolling the Environment

The three patrol behaviors (unaware, with a priori info, and with discovered info) were each run for 16 trials. At the beginning of each trial the robot was positioned by hand in the same starting location. From there it always circled the room in a generally counter-clockwise fashion, passing the radio first on the left, and then the filter on the right. Although the three paths were similar, the addition of the adaptive waypoint

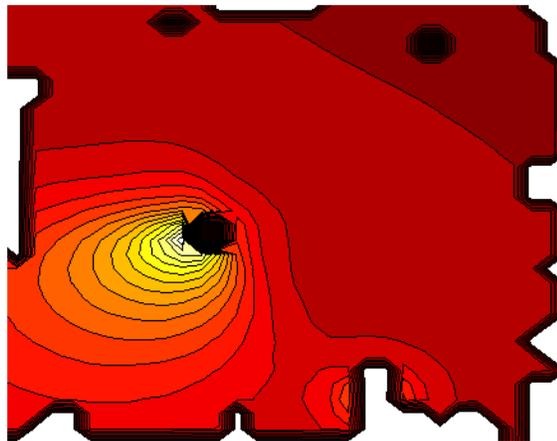
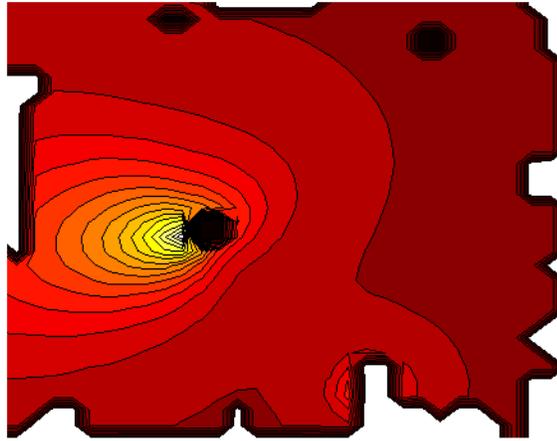


Figure 5.12. Direct field maps created from two different robot-measured data sets used in testing the improved SNR movement strategies.

selection for the last 2 cases (with a priori or discovered information) added roughly 30% more length to the robot’s round trip distance.

Over the entire distance traveled during each robot trial, the addition of the adaptive waypoint selection and gradient following behaviors decreased the robot’s exposure to ambient noise by roughly 1-dB on average. While moving through the environment, the adaptive algorithms averaged 58.5-dB, compared to 59.5 for the non-adaptive algorithm. Table 5.8 summarizes these results for the individual algorithms.

Table 5.8. Results of the adaptive waypoint following algorithm averaged over the entire path. These data were averaged across each trial before estimating mean and standard deviation.

	Unaware	A Priori Information	Discovered Sound Sources
Mean Volume	59.5 dB	58.5 dB	58.5 dB
Standard Deviation	0.6 dB	1.2 dB	0.3 dB

The region of the room that should have experienced the greatest change in volume by following the adaptive algorithm is the area influenced most strongly by the direct field of the loudest source, the radio. Looking at just those samples collected within the 3x3-m region directly in front of the source, Table 5.9 summarizes the slightly improved results recorded by the robot.

Table 5.9. Results of the adaptive waypoint following algorithm for a 3x3-m<sup>2</sup> region in front of the sound source.

These data were averaged across each trial, before estimating mean and standard deviation.

	Unaware	A Priori Information	Discovered Sound Sources
Mean Volume	59.9 dB	58.2 dB	58.6 dB
Standard Deviation	0.7 dB	1.3 dB	0.4 dB

The addition of the adaptive waypoint selection mechanism is simply not making a big difference in the overall ambient noise exposure of the robot. This time we see a slightly larger difference between waypoint strategies, up to a 1.6-dB difference between the base case and the robot using hand-collected information. Figure 5.13 plots this same data as histogram to demonstrate the relative numbers of samples collected at each volume. Clearly, the base case collects a larger percentage of samples at higher volumes. Unfortunately, however, these results are not very interesting in terms of the numerical difference between runs. Where a 5-10 dB drop is potentially valuable when combined with other filtering equipment, a 1-2 dB change is not very significant. More variable noise sources such as music will easily vary 5 or more decibels over a single song. As such, the extra path length required for rerouting the robot in this case do not appear to have been worth the improvement in signal quality. The question that remains is why did the adaptive path planning not work as well as the previous repositioning tests.

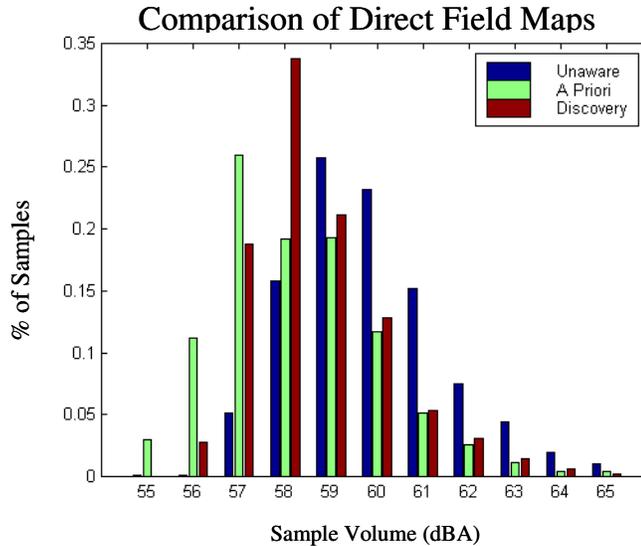


Figure 5.13. Histogram of all data volumes in a 3x3-m<sup>2</sup> region in front of the radio source collected by the robot during the improved SNR movement strategy trials. The trials using the unaware route through the environment recorded more samples of higher volumes than either aware strategy.

Theoretically, if the robot were traveling within 1-m of the sound source, which was reachable by the robot in this scenario, then moving the robot out to a distance of 3-m (the average distance at which the adaptive algorithm traveled) from a 67-dB source should see a nearly 10-dB drop in noise levels due to the direct field. We do not see this drop, and the reasons for this are twofold. The first problem with this calculation is that the reverberant field tends to be stronger than the direct field after 1-2m, so the 10-dB drop is clearly an overestimate. Beyond a couple of meters, the robot should not really see any significant drop in sound for this environment.

The second problem with this calculation is that the robot was never traveling at a distance of 1-m from the source. Given that the robot had a 3-m sensing range, the initial selection of waypoints for detecting new sources placed the robot almost 2-m from the

radio at the closest. Choosing an “informed” waypoint path only moved the robot out another 1-m, a maximum difference of 3-dB in the measured volume of the direct field. From a noise reduction standpoint, 3-dB is already uninteresting in the general case.

Overall, these results suggest that acoustical awareness may not be necessary, when patrolling and sampling along a route, to improve signal-to-noise ratio in the general case. Looking at the results from this environment, using noise gradients had a small impact in improving performance, but the size of that impact would only make a significant difference if the original waypoint placed the robot very close to a sound source.

### **5.3.3 SUMMARY OF THE IMPROVED SIGNAL-TO-NOISE RATIO EXPERIMENTS**

The goal of the robot in this second set of patrol related experiments was to improve the signal-to-noise ratio (SNR) in order to enhance the performance of classification or other general auditory behaviors. The results of these experiments, however, were more mixed than those of the acoustic monitoring task.

When a robot is trying to listen for a sound in the environment while it is not moving through the environment, the use of a noise map allows it to more consistently select better locations in the environment from which to listen. However, a more reactive approach, where the robot simply moves as far away from local sources as possible will also work in most environments. The trade-off in using the reactive approach may be that more time is required to find a good location in the environment, as the robot may need to try out a number of locations first. Furthermore, if the environment is very

cluttered acoustically, then the use of the reactive approach may result in a poorer solution than using a noise-map to make an informed decision.

In contrast to the stationary listening position, the use of a noise map to improve SNR for a moving robot was generally not any more effective than an unaware approach for the one source/environment configuration that was tested. With active sources only 10-15 dB louder than the robot, there was only a measurable difference between the robot paths when the robot was within 2-m of the sound source. In the unaware scenario, however, the robot was not usually within that range. Even when it was, it was not located that much closer to the source than the acoustically-aware path. If the robot had originally passed closer to the source, then there would have been a more significant difference between the chosen paths.

Overall, this series of experiments emphasized the selective use of knowledge-based acoustical awareness for improving SNR. When the robot is going to be exposed to a significant amount of ambient noise, choosing a better path or stationary listening position using predicted sound flow information could make a real difference on performance. When the robot is located in a region dominated by reverberant sound, however, selecting a better path or location in the environment is often unnecessary. Therefore, the type of acoustically-aware positioning or navigation used should depend on the environment in which the robot is being deployed. In acoustically challenging environments, a robot can use knowledge-based awareness to avoid problems advance, while other, more benign environments may only need to react locally in regions of excessive ambient noise. Further testing may clarify the limitations of these methods.

## 5.4 CHAPTER SUMMARY

In Chapter 5, we applied acoustical awareness to the domain of the autonomous mobile security robot, focusing on two specific aspects of the domain: (1) monitoring the auditory scene, and (2) improving the signal-to-noise ratio while listening to the environment. Each of these scenarios made use of the robotic discovery capabilities discussed in Chapter 4 to identify and localize sound sources in the environment. Each algorithm also exploited the sound fields framework described in Chapter 3 for knowledge-based acoustically-aware applications. The ways in which the robot used this knowledge to determine movement through the environment, however, differed between the applications.

In the acoustic monitoring scenario, the robot used its knowledge of the auditory scene to determine when the environment had changed. After completing a data collection run through the environment, it analyzed the data to make predictions about whether a new source was present, where a new source is most likely to be located, and whether known sources in the environment have been turned off or changed in volume/sound function. The sound propagation framework described in Chapter 3, therefore, served primarily as a predictive tool, allowing comparisons with measured data. Although the framework did not directly control robotic movement through the environment, it did still influence robotic movement by detecting change, which would require further investigation by the robot to confirm.

In the enhanced signal-to-noise ratio scenario, the robot used its knowledge of the auditory scene to directly influence its movement through the environment through map building. From knowledge gathered either *a priori* or through robotic investigation, the

robot made predictions (maps) about the current state of the environment and moved to avoid those regions believed to contain loud ambient noise. Although the robot's success in improving the SNR differed substantially between environments, the overall results suggested that making predictions about the room from the knowledge that was available and using those predictions to guide robotic movement could reduce the robots exposure to ambient noise.

The goal of this chapter was to demonstrate the applicability of the sound propagation framework to real robotic applications. Although the underlying physics of sound fields have been repeatedly validated in other research communities, they had never been applied to mobile robotic navigation before, and so the question of their usefulness to this community was in doubt. In this chapter, however, we have concretely demonstrated two different methods by which an acoustically-aware robot can use knowledge of sound propagation to influence movement: (1) the robot can use sound propagation to determine change in the environment, affecting future decision making, and (2) the robot can directly apply sound propagation to plans for future robotic movement.

In the following two chapters, we will expand upon these same general themes of robotic movement in response to the surrounding environment. Chapter 6 will explore in more detail using sound propagation models to guide robotic movement, only from the perspective of the robot as a sound source, rather than the robot as a listener. Chapter 7 will then examine action selection in the presence of transient or short-duration noise, instead of just medium-to-long duration ambient noise sources.

## CHAPTER 6

### THE STEALTHY APPROACH SCENARIO

Chapter 5 focused on the use of acoustical awareness in applications where the robot is the listener. Although the robot still had an effect on the surrounding auditory environment, and, therefore, on the performance of the application, the goal of the application was to listen for changes in the auditory scene. In this chapter, the robot switches roles. Now, instead of the robot listening for something in the environment, some observer in the environment is listening for the robot. Being acoustically aware, the robot needs to adapt to the auditory scene to maximize its performance with respect to the external listener. Combined with the work from Chapter 5 in controlling the listener, this set of experiments answers the third, and final, sub-question posed in Chapter 1. How does acoustical awareness change with control over the source vs. the receiver? The application domain in which we explore this robot-control problem is the stealthy approach scenario.

As an observer, a robot's primary virtues are patience and tolerance. If tasked with watching for a tiger in the environment, the robot, like a stationary camera, can wait as long as its batteries hold up for the animal to finally cross its path. It does not get bored, and it does not get uncomfortable with remaining in place for a long time. Best of all, if the robot, or its human partner, decide that it is located poorly, then it can move to another location. In the future, these robotic advantages of tolerance, patience, and mobility will serve well for observing, not only, animals, but also natural events, people, or even locations (e.g. security guard). In most of the current applications, however, the

robotic platform being used is not a small, unobtrusive robot. Military applications, for instance, often use planes to cover as wide a region as possible, accompanied by all the noise of keeping the plane in the air. Ground robots, either for military, police (bomb-squad), or building security applications, have a similar problem in that they have to be fairly large for the sake of robustness. As such, these robots are noisy due to extra onboard cooling fans and motors designed to move heavier equipment. How can such a noisy robot be used to quietly observe, or approach a target, when the target is a flight risk? The one solution that has been deployed for wildlife observation relies upon cables hanging in the trees out of sight, limiting where, and how quickly, the robot can move towards the target [Estrin et al. 2003]. An alternative solution allowing closer observation is to hide the robot while stealthily approaching the target. But while there has been some limited work in visually hiding the robot from the target [Birgersson et al. 2003; Kennedy et al. 2007], no attention has been paid to hiding the robot aurally. We believe that the solution to this problem lies in making a robot aware of the surrounding auditory scene. By knowing something about the listener, the environment, the sound sources, and the physical principles that govern how they each affect sound flow, a robot can make predictions about how it will be perceived by a listener, and adjust its navigational strategies appropriately.

In the following set of experiments, we implement a navigational controller that incorporates acoustical awareness into a stealthy approach scenario. Assuming that our listener is capable of recognizing either overall changes in volume or significant changes in volume from any given direction, a stealthy robot needs to recognize how its own movements will be perceived by each of these listener capabilities, and incorporate that

into its own movement strategy. Specifically, to reduce the acoustic impact of the approach on the listener, the robot needs to first estimate the overall volume of ambient noise the listener is exposed to at their current location and the relative masking effects of each source in the environment. Then the robot predicts for all reachable locations in the environment how loud it will sound to the listener from that location and, combined with the previous information, identifies the path that will expose the listener to the perceptually least amount of robot-generated noise. In this dissertation, we assume that the listeners' position is known *a priori*, but this information could also have been determined by other onboard sensors such as stereo-vision [Martinson and Brock 2007].

The remainder of this chapter is described in four sections. Section 6.1 describes an initial heuristic developed to take advantage of a few specific masking properties of ambient noise sources in the room. Section 6.2 then describes the experimental results in hiding a real robotic platform using this heuristic. Using the initial results to guide further investigation, Section 6.3 discusses how the initial heuristic can be either replaced or augmented to incorporate more principles of sound propagation through the room and possibly hide a robot in sound functions that vary over time. Section 6.4 then concludes this chapter with a summary of results, and a comparison to the work in Chapter 5 using the robot as a sound source. The work presented in Sections 6.1 and 6.2 is to be published in the proceedings of the 2007 IEEE Conference on Intelligent Robots and Systems (IROS) [Martinson 2007].

## 6.1 HOW TO HIDE A NOISY ROBOT

The scenario proposed for testing the acoustic hiding abilities of a robot is the stealthy approach. The target being approached is a 4-element microphone array capable of detecting changes in the overall volume, as well as identifying changes in the relative volume from each direction. This listening system is designed to mimic the perceptual capabilities of a human target, which can identify changes in overall volume, and separate sound sources from each other by angle. For now, our sensor system is not searching for differences in pitch.

Given this target listener with known location, the robot's goal is to approach the target as quietly as needed, moving from some starting location to within a meter of the sound source. If the environment is loud, however, the robot should also be able to recognize how the loudness limits observation by the listener, and include that into its stealthy approach. For this task, the robot is given knowledge *a priori* of significant sound source locations in the environment, their directivity, and a spatial evidence grid from which it can localize itself with respect to the environment. As demonstrated in Chapter 4, these are all pieces of knowledge that could be acquired by the robot. Their acquisition, though, would require that the robot be deployed to that area at some time before being asked to approach the target.

The methodology used to hide the robot's acoustic signature is based on the capabilities of the target listening device. First, the robot estimates the volume of noise the observer is exposed to without the presence of the robot. Second, using the provided obstacle map, the robot identifies a set of discrete reachable locations in the environment. Then, for each location, the robot estimates a cost of visiting that location based on: (1)

the absolute difference in volume at the receiver due to the robots presence at that location, and (2) the difference in the volume coming from the direction of the robot. Finally, these two cost estimates are combined together using weighted summation, and a path-planner identifies the path of minimal cost for the robot to travel.

### 6.1.1 ESTIMATING VOLUME AT THE TARGET

The first step in hiding a noisy robot is to estimate the overall volume detected by listener. This will be used to determine which areas of the environment are considered safe for the robot to enter undetected.

When making this estimate, the effects of two sound fields need to be considered: direct, and reverberant sound. Any transmitted sound that the robot is aware of can be modeled as a separate source co-located in the wall, so we do not need to include a separate field describing transmitted noise. As discussed in Chapter 3, the direct field is the simplest to estimate, being a linear decrease in pressure amplitude with the distance from the source. Given a sound source ( $S_i$ ) of volume ( $V_i$ ), the angle ( $\alpha_i$ ) and distance ( $d_i$ ) from that sound source to the listener, and the directivity function of that source ( $Q_i(\alpha)$ ), we can re-write Equation 3.3 in terms of the sound pressure level (dB) to better compare with other data:

$$S_i(d_i, \alpha_i) = V_i Q_i(\alpha_i) - 20 \log_{10}(d_i) \quad \text{Equation 6.1}$$

The effects of the reverberant field may also play a significant role in masking or revealing the robot's approach. In particular, sound sources that are not close enough to the target to have any direct effect on the environment, may raise the overall volume of the room to loud enough levels that a robot is not detected from any direction. To

incorporate this effect into its volume estimate, the robot samples the environment at some location far away from known ambient noise sources, and uses the constant reverberant field assumption (see Chapter 3) to adjust Equation 6.1. The estimated combined volume of noise heard by the listener ( $T$ ) is then the logarithmic sum of the volume due to each source plus the reverberant field effects:

$$T = 10 \log_{10} \left( 10^{R/10} + \sum_i 10^{S_i/10} \right) \quad \text{Equation 6.2}$$

As was discussed in Chapter 3 ray-tracing can also be used for this task of estimating the volume of the robot at different locations in the environment. Section 6.3 will demonstrate how this is true even though the robot is a moving sound. Ray-tracing was not used in these experiments, however, so as to first explore the feasibility of the stealthy approach scenario.

### 6.1.2 MINIMIZING CHANGES IN VOLUME

After estimating the volume of noise heard by the listener, the next step is to estimate how loudly the robot will be detected. Specifically, for each location in the environment that a robot can move through, how much additional noise would the listener hear due to the presence of the robot at that location? This is accomplished by again using spherical propagation (Equation 6.1) to estimate the volume of sound reaching the listener. Repeating this direct sound estimation for every unobstructed location in the environment, we can create a map of how loud the robot will appear to the listener for every location (Figure 6.1).

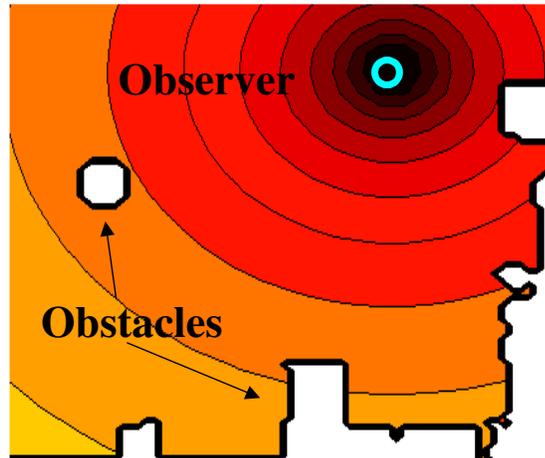


Figure 6.1. Contour map of the estimated noise at the observer due to the robot. Darker is louder.

For now, this model does not include any reverberant obstacle or obstacle effects on sound propagation. Unlike the previous section, this section requires estimating the effects due to a single source, the robot. As such, the robots' own reverberant effects on the environment cannot be easily estimated without more knowledge of the environment, or measured in the presence of other noise sources. In the discussion section, however, we will describe how a ray-tracing model can be used to include both the direct field and reverberant field effects from a moving robot source.

### 6.1.3 AVOIDING DIRECTIONAL CUES

Knowing just the volume of the robot at the target, however, is only part of the problem. Since the target is a microphone array, it is capable of estimating the angle to the detected sound source. So even if the overall volume of noise did not change significantly, it can still detect the robot if there is a significant deviation in angular energy from the baseline. Hiding the robot, therefore, requires choosing a path that also

minimizes the change in angular energy. Since the robot's emitted energy is assumed to be the same for all positions in the environment, the only way to minimize the change in angular energy is to pick an approach angle obscured by large amounts of energy from ambient noise sources. In most cases, such approach angles are along the line from the source to the listener.

Getting the robot exactly in line with the target and the noise source, however, may be very difficult. Not only can errors in robot position estimation cause the robot to misjudge the approach angle, but physical obstacles in the environment can also make it an impossible task. The question becomes how close does the robot need to be to the desired approach angle? For now, we assume that how much the robot is masked by an ambient noise source depends on how loud the source is, and how far the robot is from the axis joining the source and the listener. For this purpose, we use a heuristic to estimate the directional occlusion of each source separately in dB, and sum the results together:

$$D_i(x, y) = W(l_i(x, y)) [V_i Q_i(\alpha_i)]$$

$$D_{x,y} = 10 \log_{10} \left( \sum_i 10^{D_i(x,y)/10} \right) \quad \text{Equation 6.3}$$

Where  $V_i$  and  $Q_i$  are the source volume and directivity,  $D_i(x,y)$  is the resulting directional occlusive effect at position  $(x,y)$  for source  $i$ ,  $l_i$  is the distance from the robot to the line between source and listener, and  $W$  is a normalized bell curve with standard deviation of 1-m.

#### 6.1.4 PICKING A PATH

Now that we have finished estimating the volume at the listener, the volume of noise due to the robot, and an occlusive effect due to each source in the environment, the next step is to estimate the combined impact on the listener ( $I_{x,y}$ ) for a robot being in each reachable location  $(x,y)$ . In particular, we need to quantify the change in the auditory scene at the listener's location due to the robot, either from changes to the total volume heard by the listener or changes in directional volume. This total impact can then be used with a path-planning algorithm to find the path with the smallest impact.

The first step in our heuristic for minimizing impact is to identify the environmental impact on the observer ( $Env_{x,y}$ ). This is calculated as a log summation of the predicted total volume ( $T$ ) at the observers' location, plus directional occlusive effects ( $D$ ) in viewing the robot at position  $(x,y)$ :

$$Env_{x,y} = 10 \log_{10} \left( 10^{T/10} + 10^{D_{x,y}/10} \right) \quad \text{Equation 6.4}$$

Next, the impact of the robot traveling through that location ( $I_{x,y}$ ) is estimated as the total impact on the listener (environmental impact plus the estimated sound heard by the listener due to the robot,  $R_{x,y}$ ) minus the environmental impact:

$$I(x,y) = 10 \log_{10} \left( 10^{R_{x,y}/10} + 10^{Env_{x,y}/10} \right) - Env_{x,y} \quad \text{Equation 6.5}$$

Finally, the robot picks a stealthy approach path by finding the shortest weighted path from the start to the goal using Dijkstra's single-source shortest path algorithm with impact being the weight of being in any given location. This results in a minimal impact approach path to the target. Figure 6.2 shows a contour map of the estimated overall impact of the robot being at any unobstructed location in the environment, using these equations with one 57-dB source.

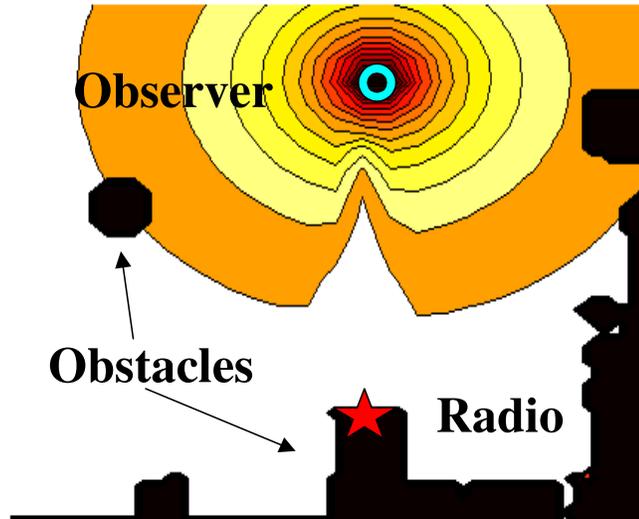


Figure 6.2. Contour map showing estimated impact on an observer due to a robot at any reachable location in the environment. Darker means greater impact.

The reason for using an absolute difference in Equation 6.5 is that we would like the impact maps to vary with reverberant levels in the room. If the listener is overwhelmed by an 80-dB noise in the area, then the impact of the environment should dominate the equation and reduce the impact of an approaching robot generating only 47-dB. If, on the other hand, the environmental impact of the target is a relatively quiet 40-dB, then the approach of the robot should be a lot easier to detect. Using an absolute difference between total impact and environmental impact will reflect this difference, and allow the robot to adjust its path to the current level of reverberant sound in the environment.

## 6.2 EXPERIMENTAL RESULTS

The robot hardware that was used for this task was the Pioneer2-dxe robot equipped with a SICK LMS200 for localization and obstacle avoidance. This robot platform emits roughly 47-dBA of noise in all directions (as measured by a Type II SPL-Meter) from its onboard cooling fans while standing still. Additional ego-noise in the form of impulse sounds from the wheels rubbing on the tile floor is also occasionally observed during robotic movement.

The goal of the stealthy robot is to move from a specified start position to within 1-m of the observer's position as quietly as possible. This work was tested with the pionner robot in a total of four scenarios spread across two different environmental layouts. In each of these scenarios, the performance of the robot trying to approach the target stealthily is compared to a robot taking an alternative, usually shorter path. Figure 6.3 shows the layout of one scenario setup in the Mobile Robot Laboratory, along with the two paths taken by the robot in the first scenario. The obstacles shown in the middle of the lab are all roughly 1-m in height.

### 6.2.1 EVALUATION METRICS

Evaluation of the robot's performance involved analyzing the data collected from a 4-element microphone array located at the target's position. Sampling at 8192-Hz, the array collects 1-sec samples continuously over the duration of the run. This includes collecting 30-sec of data with no robot in the room to set a baseline, and then, roughly 100 samples for longer paths, and 50 samples for shorter paths. Each sample was then analyzed to determine:

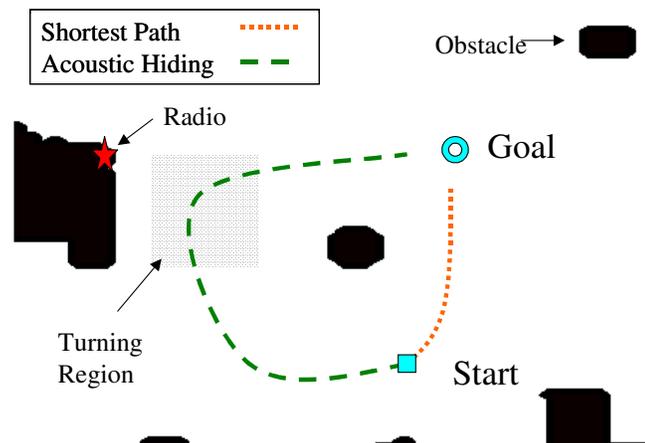


Figure 6.3. Layout of the acoustic hiding scenario. The robot that does not try to hide approaches the observer along the shortest distance path. The robot that tries to hide its acoustic signature moves in line with the radio source, before approaching the observer at the goal.

- **Metric #1 - Overall change in volume from baseline (dB)**
- **Metric #2 - Change in volume from the direction of the robot.**

This second metric required that each sample also include an estimate of where the robot was currently located in the room. For this purpose, we collected the believed location of the robot from the player/stage amcl (Adaptive Monte-Carlo Localization) [Gerkey et al. February 2006] driver whenever a sample was collected. Then, to estimate performance, we used a time-delay estimation algorithm, based on generalized cross correlation measurements, to estimate the energy at 1-m from the listener in the direction of the robot. The difference between this energy (in dB) and the mean energy at that angle from all noise samples (in dB) is the empirical measure of angular impact on the listener due to the robot.

## 6.2.2 FIRST ENVIRONMENTAL LAYOUT – APPROACHING FROM THE LEFT

The layout of the first scenario is a relatively open 8x8-m<sup>2</sup> environment, with an observer located relatively far from any walls and a sound source on the left of the observer, oriented in the observers direction (Figure 6.3). In the first scenario using this environmental layout, the robot uses its knowledge of the radio in the environment to hide itself better than an uninformed robot taking the shortest path to the target. In the second scenario using the environmental layout, the performance effects of a significantly louder reverberant field are examined.

### Hiding in Front of a 67-dB Source

In this scenario, a 67-dB source was placed 4-m to the left of the listening microphone array. That source was an fm radio with a typical cardioid directivity pattern generating static noise. Starting from a location below the listener in the map (Figure 6.3), the shortest path was to move upwards in a roughly straight-line while avoiding obstacles. The robot that was trying to hide its acoustic signature, however, would move upwards to get in line with the source before approaching the target. This scenario was repeated 30 times for each robot path.

Given our open environment, and the listener's positions being all relatively far from the wall, the first metric did not produce significantly different results for the two paths except in one region. For most of either path, the 47-dB robot added little overall volume (<1dB) to the total energy in the room (metric #1). This is not surprising as the reverberant field averaged 54-dB for this environment, while the reverberant field due to the robot (measured with the sound source turned off) added a significantly smaller 43-

dB. The exception to this rule, however, was part of the path taken by the acoustically hiding robot where the robot turned relatively sharply to get in line with the radio. This region is marked “turning region” in Figure 6.3. While turning, the robot generated a noticeably louder amount of noise, mostly tire squeaking and equipment rattling, which violated the original assumption of the robot as a constant 47-dB source.

With the exception of the turning region, the first metric had very similar results for either path. The second metric measuring directional energy, however, demonstrates a significant difference between the paths. Figure 6.4 demonstrates this graphically, plotting the average directional energy measured by the observer vs. distance of the robot to the observer. This data has been smoothed using a Gaussian smoothing function with standard deviation of 0.1-m. Looking at the shortest path energy from 3.5-m to the stopping point 1-m from the observer, a relatively steady volume can be detected until ~1.5-m where the presence of the robot becomes more noticeable. In contrast, the robot trying to hide from the observer first demonstrates higher energy while it is getting in line with the source, but then quickly drops in volume as the robot hides in the radio noise.

After collecting and analyzing the samples from all runs, Table 6.1 presents the results of the directional energy metric (#2), broken up into the percentage of samples that fall into each energy range. Figure 6.5 shows a histogram of the same data. In general, the results demonstrate that the solution for hiding the robot is not perfect, as the robot is still detected more often than not by 3 or more decibels for both the stealthy, and shortest-path approaches. The performance difference between the algorithms, however, becomes more apparent when looking at the number of samples where the change in angular energy was less than 1-dB (i.e. the robot was unnoticeable). While the shortest

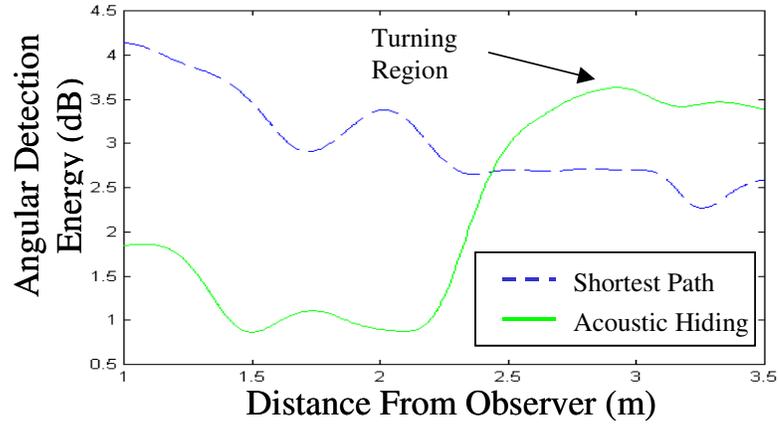


Figure 6.4. Comparison of the angular detection energy observed while the robot was taking the shortest path vs. the acoustic hiding path for the first environmental layout containing a 67-dB source.

Table 6.1. Acoustic hiding results in the presence of a 67-dB radio source. The results describe percentage of overall collected angular energy measurements recorded in each decibel range for distances less than 3-m from the target.

Energy Range	% of Samples with Directional Energy				
	$\leq 1$ dB	1 – 2 dB	2 – 3 dB	3 – 4 dB	> 4 dB
Shortest Path Results	0.28	0.00	0.03	0.32	0.37
Acoustic Hiding Results	0.46	0.00	0.02	0.19	0.33

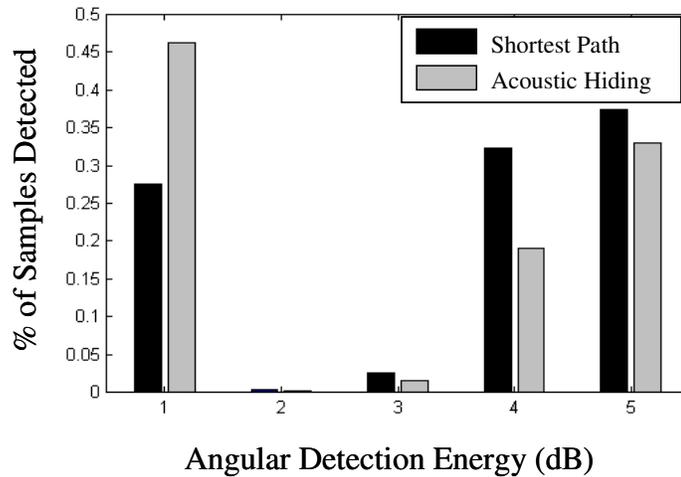


Figure 6.5. Bar chart comparing the angular detection energy recorded by the observer for each robot path.

path was unnoticeable 28% of the time, the robot that hid in the radio static noise was unnoticeable 46% of the time on a similar sized sample set (~870 samples/path).

### Loud Room Scenario

The second test using this room layout was a repeat of the 57-dB source scenario, except that the reverberant field in the room was raised to over 60-dB using a loud floor fan placed in a far corner of the room (away from the testing area). The hypothesis behind this test was that a loud enough room should eliminate the advantage of any particular path, because the addition of the robot will be too small.

The effect of this change to reverberant sound levels on the robot's path-planning algorithm was to logarithmically reduce the cost (or weight) of visiting any grid-cell in the map. This applied nonlinear decrease in all weights meant that the shortest-path became less costly than the longer path, because the robot traveled across fewer grid cells to reach the goal. Therefore, after detecting the change in reverberant noise, the robot

does not try to get in line with the source, but simply approaches the source from the shortest distance path. To determine whether or not this path was chosen correctly, we also tested the path chosen for the quieter room with just the 57-dB source. The results of this testing are shown in Table 6.2.

Table 6.2. Acoustic hiding results in the presence of a 67-dB radio source and a loud reverberant field. The results describe percentage of overall collected angular energy measurements recorded in each decibel range for distances less than 3-m from the target.

Energy Range	% of Samples with Directional Energy				
	≤1 dB	1 – 2 dB	2 – 3 dB	3 – 4 dB	> 4 dB
Shortest Path Results	0.99	0	0	0.01	0
Acoustic Hiding Results	0.96	0.00	0.00	0.02	0.02

The two paths were each tested 15 times in this loud reverberant field scenario. The overall increase in volume detected by the observer (metric #1) was minimal (<1-dB) for all parts of either path. Measuring angular detection energy (metric #2) saw similar results. Taking the shortest path meant a less than 1-dB increase in volume over the maximum reverberant field noise in 99% of the samples, while the robot on the longer path remained unobserved in 96% of the samples. With the longer path, 90% of the detected samples occurred in the “turning region” where the robot is aligning itself with the radio.



### Hiding in Front of a 67-dB Source

In this scenario, the same radio source used in the previous room layout was moved to a location 4-m below the listening microphone array. Starting from a location to the left of the listener in the map (Figure 6.6), the shortest path was for the robot to move to the right in a roughly straight-line while avoiding obstacles. The robot that was trying to hide its acoustic signature, however, would move down, along the wall, before moving upwards to get in line with the source to approach the target.

As expected, this scenario saw a significant decrease in performance, both overall, and relative to the other first room layout. The total volume due to the robot remained small over the entire path, with no region exceeding the average noise level by more than 1-dB. Looking at the angular energy, however, we can see that the robot that is trying to hide in the radio's noise was undetected (energy less than 1-dB) in only 17% of the samples. While this was still better than taking the shortest path, where the robot remained unobserved in less than 9% of the samples, the difference between the two runs was not as significant as when the robot approached from the left in the shadow of the same ambient noise source. The numbers for each volume range are sorted by path taken in Table 6.3.

Figure 6-7 plots this same data against the distance to the target, using a sliding window to smooth out the data. The energy detected from the acoustically hiding robot is visibly less than that the robot taking the shortest path, but not by as much a margin as with the first room layout.

Table 6.3. Acoustic hiding results in the presence of a 67-dB radio source located near a wall. The results describe percentage of overall collected angular energy measurements recorded in each decibel range for distances less than 3-m from the target.

Energy Range	% of Samples with Directional Energy				
	≤1 dB	1 – 2 dB	2 – 3 dB	3 – 4 dB	> 4 dB
Shortest Path Results	0.09	0.00	0.02	0.22	0.67
Acoustic Hiding Results	0.17	0.01	0.02	0.25	0.55

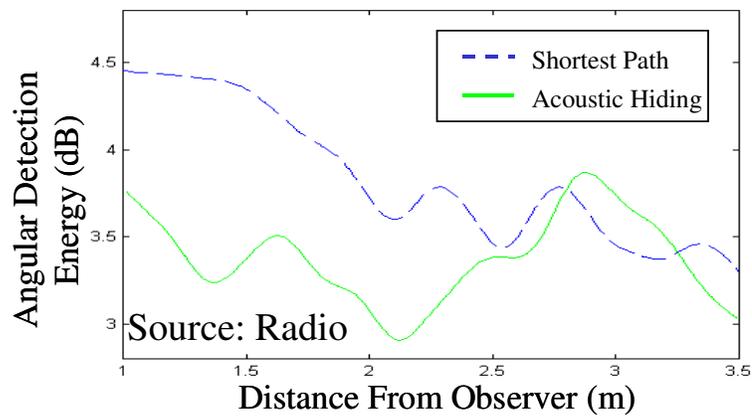


Figure 6.7. Comparison of the angular detection observed while the robot was taking the shortest path vs. the acoustic hiding path for the second environmental layout containing a 67-dB source.

### Hiding in Front of a 54-dB Source

In this last scenario, a 54-dB source was placed 4-m below the listening microphone array. The source was an air filter with a bipolar directivity pattern generating wind noise. With the 3-dB difference between this source and the radio source, it was expected that this configuration would produce another drop in performance.

As with the previous 3 runs through the environment, the first metric of overall volume again showed little difference between the 2 paths to the target. In general, the overall volume does not appear to have been a very useful metric, as the overall volume change remains small until the robot is very close to the target. In contrast, however, the directional energy metric has shown not only a difference between runs, but also a difference between room and now, source layouts (Table 6.4).

Table 6.4. Acoustic hiding results in the presence of a 54-dB filter source. The results describe percentage of overall collected angular energy measurements recorded in each decibel range for distances less than 3-m from the target.

	% of Samples with Directional Energy				
Energy Range	≤1 dB	1 – 2 dB	2 – 3 dB	3 – 4 dB	> 4 dB
Shortest Path Results	0.09	0.00	0.01	0.12	0.78
Acoustic Hiding Results	0.12	0	0.01	0.16	0.71

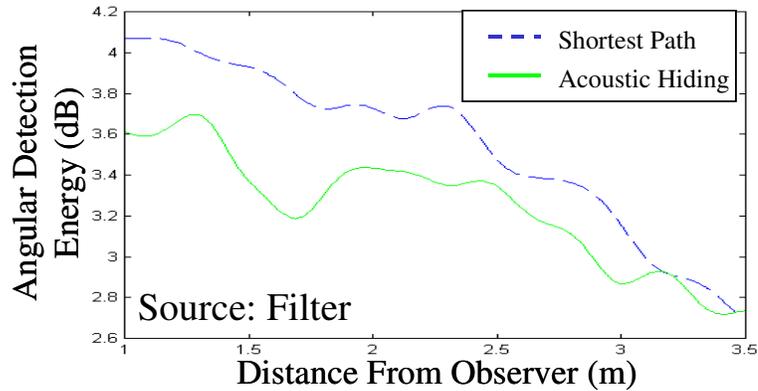


Figure 6.8. Comparison of the angular detection observed while the robot was taking the shortest path vs. the acoustic hiding path for the second environmental layout containing a 54-dB source.

In Figure 6.8, we can see the same problem in the plot of average angular detection energy vs distance to the observer. This plot shows almost a constant offset for the two paths. Where the radio sources produce a sharp dip in the energy once the robot moves in front of the source, the fan did not produce such a dip. Instead, the presence of the fan appears to merely lower the overall detection of the robot by some small amount.

### 6.3 DISCUSSION OF RESULTS

The goal of this initial work in acoustic hiding was to demonstrate that a robot could hide its own acoustic signature in the ambient noise. Using the tools provided in Chapter 4, a robot can collect knowledge of environmental layouts, sound source positions, sound source directivity, and reverberant field estimates. Then, with that knowledge of the auditory scene, a robot can seek to position itself between known sources and a target, reducing the chances of being detected by a listener at some arbitrary location in the environment.

In this work, two different evaluation metrics were used to test the effectiveness of two different approach paths: the shortest path, or a stealthy path. The first metric, measuring the overall volume change, showed no improvement between either path for any of the four scenarios. The addition of a 47-dB robot to the general sound field had little overall effect on the volume of sound observed by the target. The second metric measuring changes to directional volume, however, demonstrated a significant difference for some environments between a robot hiding in front of a sound source and a robot taking the shortest path. This difference depended upon a number of factors. In general, the lower the volume of the source disguising the robot's approach, the easier the robot was detected. This was countered, however, by changes to the reverberant field. If the reverberant field increased substantially, then the robot may not need a stealthy approach to remain undetected. The presence of nearby walls in the environment may also make the robot more detectable, as will certain types of robotic movement that cause the robot to generate more noise. In summary, there is much interesting research remaining in exploring this problem and building a real application.

There are two additional issues in particular, however, which have not yet received much attention in this scenario. The first such issue is that of an all inclusive impact model. The model described in Section 6.1 was designed to force the robot to move in line with a source, without real consideration for sound propagation models. This was important in order to demonstrate the feasibility, and general interest, of the scenario, but it lacks the mathematical rigor appropriate for a general application. Section 6.3.1 will describe how the sound propagation framework covered in Chapter 3 can be extended to the visibility model.

The second such issue that needs to be addressed is hiding in arbitrary sound functions. Filter, or fan, noise is common to many environments, but other sources such as fountains, machinery, speech, and/or music are equally common. In Section 6.3.2, we will discuss the application of auditory masking models to the stealthy approach scenario, allowing a robot to hide in a wider range of sound functions.

### **6.3.1 REPLACING THE HEURISTIC WITH REVERBERATION MODELS**

In order to get a robot in line with a source while including knowledge of environmental reverberation effects, a heuristic using a weighted summation was used to combine disparate effects. It included expected volume increases due to the direct field of the robot. It also included increases in energy from the direction of the robot. This heuristic, however, did not include any reverberant effects due to the robot, so a path in which the robot traveled along a wall was not considered necessarily any worse than any other path. Using the more rigorous mathematical approach to sound propagation described in Chapter 3, we can estimate all of these properties useful for stealthy approach with a single algorithm. The algorithm is ray-tracing.

Chapter 3 described ray-tracing as a mathematical methodology for estimating sound propagation through an environment when a description of the room layout (obstacle-map) is available. Appendix B.6 gives more details on ray-tracing implementation used in this dissertation. In a typical ray-tracing approach, some number of virtual rays (3600 rays in this implementation) would be generated at random angles from the sound source into the room, traveling in a straight line until they hit a surface, at which point they are reflected back into the room. The rays continue traveling in this

fashion, affecting many different receiver positions, until their energy levels become negligible by traveling too far away from the sound source. The estimated effect on a single receiver then is only a matter of determining which rays intersect with the position of the receiver and summing their energy together. From this, we can determine both the overall change in energy due to a robot located at a particular location, as well as changes in the sound intensity profile (energy at any given angle) at the observer's location.

To use ray-tracing as described above, however, with the virtual rays emanating at random directions from the source, would be very computationally expensive for this scenario. Since the source in question is a robot, this form of ray-tracing would have to be repeated for every reachable location in the environment in order to plan an optimal path through the environment. Even with modern computers, this could take a long time, which only increases with the level of detail present about the environment. For this scenario, however, there is a computationally cheaper alternative. The observer does not move in this scenario, so to significantly reduce the number of calculations necessary to build a map of impact, we can reverse the ray-tracing algorithm, generating rays at the receiver instead of the source. It is essentially the same problem as predicting the effects of a single source, since the rays themselves will still act in the same way, reflecting off of surfaces until they have traveled a maximum distance.

Using this alternate formulation of the ray-tracing method, we can calculate each of the criteria in which we are interested in identifying for the stealthy approach scenario:

- (1) changes in volume due to the robot (from both the direct and reverberant field), and
- (2) changes in directional cues due to the robot.

## Estimating Changes in Volume

Change in the overall volume at the observers' location is essentially the same measurement as was used in Section 5.1.1. Only now, the estimated volumes for both the ambient noise and the robot are determined using ray-tracing. In Figure 6.9, the change in volume plot for the first room layout, assuming no active sound sources, shows that the approaching direction of the robot is not very important, so long as the robot moves straight towards the source. With the radio present in the environment, however, this change in volume plot becomes nearly uniform. The change in volume due to the robot is less than 1-dB for all reachable locations in the room, suggesting that in the presence of significant ambient noise, the overall volume change is probably not very useful for gauging how well the robot's acoustic signature is masked. This pseudocode for creating a map of noise due to the robot using ray-tracing, can be found in Appendix B.6.2.

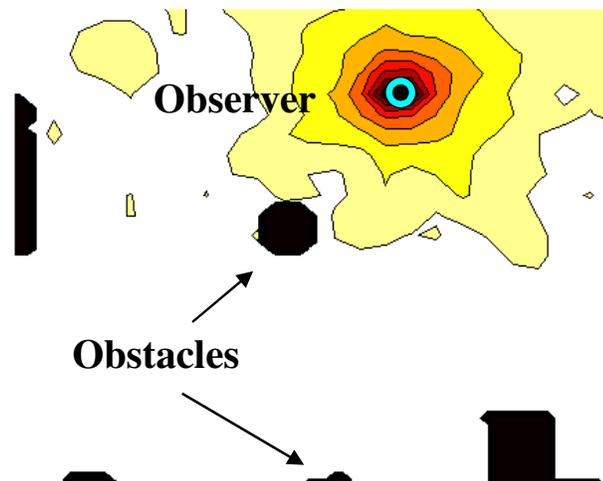


Figure 6.9. Change in total volume plot, as predicted by ray-tracing models, for the first room layout with no ambient noise sources.

### Including Directional Cues

The estimation of directional cues is where the real difference between the ray-tracing method and the heuristic discussed in Section 5.1 occurs. Changes in volume were already based on sound flow, although they did not include the reverberant field. Directional cues, however, were simply designed to get the robot in front of a sound source, without any basis in sound flow. The use of ray-tracing to estimate directional cues applies the physics of sound propagation to the problem.

To estimate the same directional cues as described in Section 5.1.2, ray-tracing is used to build a sound intensity profile at the observers location. For every angle, the sound intensity profile estimates the onset energy coming from that direction. Since each ray in ray-tracing typically estimates energy, rather than volume [Elorza 2005], the sound intensity profile is created by finding the set of rays that pass through the observer's location, and then applying a Gaussian smoothing filter ( $25^\circ$  standard deviation) across the approach angles to find energy. Equation 4-14 gives an example of using Gaussian smoothing across angles. Figure 6.10 shows the sound intensity profile at the observer's location due to ambient noise in the second scenario. This pseudocode for calculating this sound intensity profile is provided in Appendix B.6.1.

Using this same methodology, we can also estimate the effects of the robot on the observer from any location in the room. This time, however, since the rays are being generated from the observer rather than the source (the robot), the important thing to track is the departure or starting angle of each ray from the observer and the distance traveled as the ray crosses each possible robot location in the environment (distance is directly related to energy). Then, for every reachable location, identify the set of rays

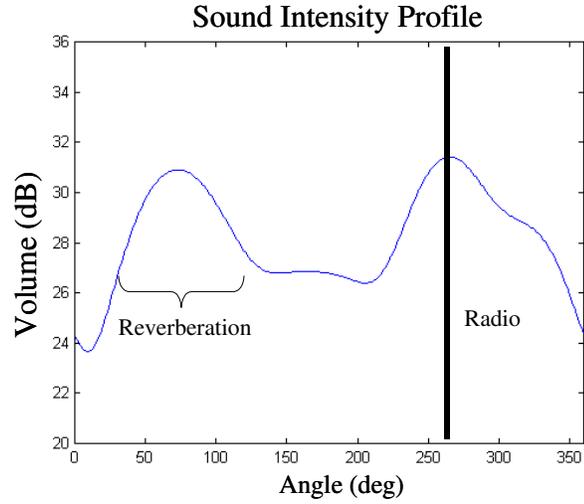


Figure 6.10. Sound intensity profile at the observer’s location due to a 67-dB radio.

that cross that location and apply the Gaussian smoothing filter across the departure angle from the observer. This builds a sound intensity profile at the observer’s location for any robot location in the environment.

Finally, with both an ambient sound intensity profile ( $Amb$ ), and a robot sound intensity profile ( $Robot_{x,y}$ ) available, we can estimate the impact ( $I$ ) of the robot on the observer at some angle ( $\theta$ ) as the log difference between the combined field and the ambient field. Seen in Equation 6.6, all units in this calculation have been converted from energy to pressure (dB):

$$I(x, y, \theta) = 10 \log_{10} (Amb(\theta) + Robot_{x,y}(\theta)) - 10 \log_{10} (Amb(\theta)) \quad \text{Equation 6.6}$$

In Section 5.1.2, the heuristic approach only evaluated the visibility of the angle from the observer to the robot because that was the only angle any predictions could be made for. Using ray-tracing, however, a visibility estimate can be made for every angle. If the angle to the robot is relatively quiet, but reflections from nearby walls are not, then

an observer could still be alerted of the robots approach, causing it to move or search for the noise source. Therefore, it makes sense to approximate the visibility of a location as the maximum visibility across all angles ( $\theta$ ). Figure 6.11 demonstrates the revised directional cues map for the second room layout where the robot approaches from below in the shadow of the 67-dB radio. Notice the similar cone shaped area of low impact in front of the ambient noise source created using the heuristic (seen in Figure 6.2).

The pseudocode for using this reversed form of ray-tracing to create these impact maps is provided in Appendix B.6.2.

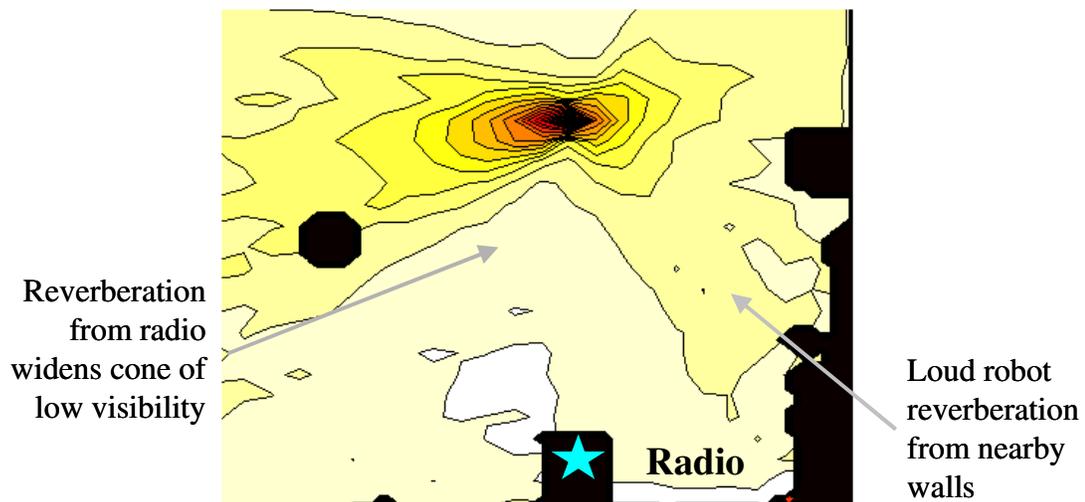


Figure 6.11. Contour plot of the revised approach to estimating directional cues using ray-tracing. Notice the similar cone shaped region of low visibility predicted by the heuristic approach (Figure 6.2). Only now, the region is somewhat wider due to the inclusion of reverberation models.

## Combining the Two Criteria

In Section 5.1, the greatest difficulty in estimating the visibility of the robot was in combining the two different criteria of visibility. If the room was loud, then that knowledge needed to be incorporated into the equation. Similarly, if there were nearby sources that would decrease visibility in certain directions, then those effects also needed to be included in the equation. This was a difficult problem because the two effects were not really being described using the same units. Using ray-tracing, however, both effects are now described in terms of volume, so combining them becomes much easier.

Equation 6.6 already includes some overall volume calculations. If there are no ambient noise sources in the environment to mask directional cues, then this equation produces a visibility contour like the one seen in Figure 6.9. By default, a robot that is closer to the observer, or to any walls, will be louder than a robot that is standing in the middle of the open room.

To make this equation complete, we can also add knowledge about “extra” noise to the visibility calculations. In the loud room scenario (first room layout), we described a loud environment where an unmapped sound source had raised up the ambient noise levels in the room to louder than expected noise levels. Incorporating this additional knowledge to the visibility equation requires simply adding a constant:

$$\Delta V(x, y, \theta) = 10 \log_{10} (Amb(\theta) + Robot_{x,y}(\theta) + R) - 10 \log_{10} (Amb(\theta) + R) \quad \text{Equation 6.7}$$

Where  $R$  is the average reverberation level in the room divided by the number of angles being tested for visibility. This assumes that the reverberant field due to other

unmapped sound sources in the room, as detected by the observer, is roughly equal in all directions.

Figure 6.12 demonstrates the addition of a 61-dB reverberant field to the same room layout seen in Figure 6.11. As validated by the robot testing performed earlier, the expected masking effects of the 67-dB radio source are significantly reduced. Using this impact map, the robot should be able to ascertain that the shortest path is no worse than any other approach path to the target that does not move along the walls.

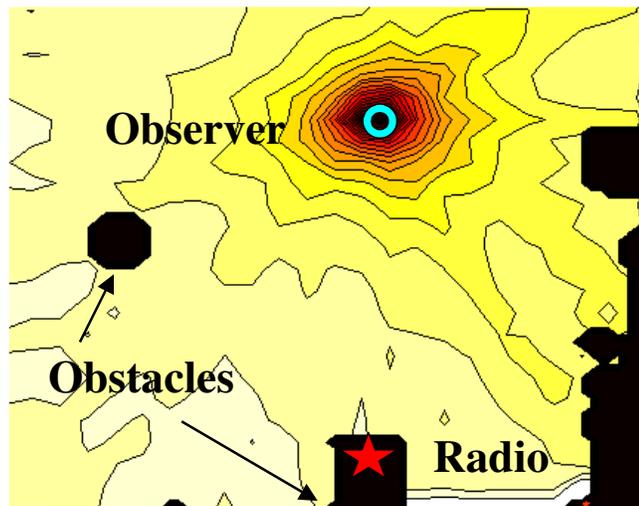


Figure 6.12. Contour plot of the maximum angular impact of the robot for scenario 2 with a loud reverberant field. The cone of low impact the robot usually approaches the source in has disappeared.

### 6.3.2 HIDING IN AN ARBITRARY SOUND SOURCE

The scenarios tested here ignored the sound functions of the ambient noise in the environment. The sounds used, fan and static noise, were generally broad-spectrum

sounds where it was simply assumed that the robot could hide itself. For an arbitrary sound source, however, the robot needs to take into account the sound function in order to determine the true masking qualities of the source. If the sound source is largely a low frequency sound source, then high-frequency noises from the robot are less likely to be masked. Also, if the sound source varies in volume, as is common with heavy machinery conducting a repeating series of tasks, a robot that's emitting a constant sound might become exposed during quieter parts of the sound function. Therefore, a robot needs to have some knowledge of the masking sound function in order to successfully hide its own acoustic signature from an observer. This problem, while certainly difficult, is not unstudied. It is called auditory masking [Goldstein 2007].

In work originally published in 1950, psychologists Egan and Hake [Egan and Hake 1950] performed a series of experiments to understand the masking properties of a single sound. In particular, the goal was to understand what frequencies and at what volume were masked (to the human ear) by a single arbitrary tone. What they found was that a tone masks frequencies around it in a roughly triangular fashion. A sound best masks those frequencies that are closest to its own frequency. However, it also masks frequencies lower and higher, with a greater effect on higher frequencies (Figure 6.13). For instance, a tone with frequencies ranging from 365-455 Hz may mask frequencies as low as 150-Hz, or as high as 4000-Hz, depending upon the volume. The reasons for these masking effects at frequencies other than the stated frequency have to do with the makeup of the human ear. More detail on this subject can be found in [Goldstein 2007].

By themselves, these experiments in auditory masking seem interesting, but possibly difficult to apply to our scenario. However, researchers in digital audio have

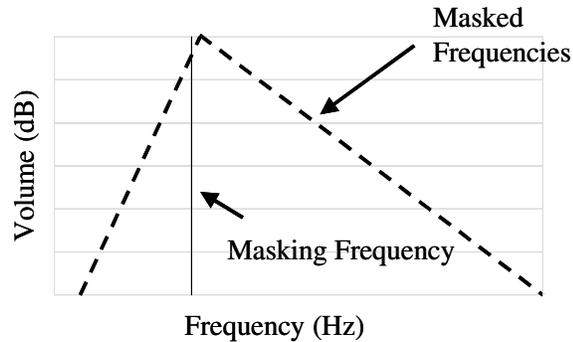


Figure 6.13. General shape of the volume vs. frequency plot for sounds masked by a single tone. A tone played at some frequency masks other frequencies in a generally triangular fashion. The farther away from the tone in frequency, the lower the volume that can be successfully masked.

continued this line of research, developing mathematical algorithms to make use of the perceptual phenomenon [Painter and Spanias 2000]. The reason for this interest is compression. Knowing that the target audience is human and that we share much the same set of hearing processes, a digital audio encoding standard can be created to maximize the perceived quality of the audio versus the size of the encoding by taking advantage of auditory masking. MP3 is such an encoding standard [Noll 1998]. For example, let us say that the song being encoded has both quiet moments with a single instrument, and loud moments where many instruments are playing at once. During those loud points in the song, the quality of the encoding does not need to be that great, as a lot of noise could be introduced during the encoding without being detected by a human listener. During those quiet moments, however, it is important to encode at very high quality, because little bits of introduced noise would be very detectable to the human ear.

Applied to our stealthy approach scenario, this work in auditory masking is potentially very relevant. Given some knowledge of the sound functions of sources in the

environment, a robot can pick a trajectory in which it will be best hidden from a human, or possibly animal, observer by masking its own acoustic signature with perceptually similar sources. With greater knowledge of the sound function over time, the robot could even adjust its own movement patterns, moving slow and quietly (or not at all) during the quiet moments, and faster, and more loudly when the sound function is at its peak volume.

#### 6.4 CHAPTER SUMMARY

In Chapter 6, we described the stealthy approach scenario, where a robot used its knowledge of the auditory scene to hide its own acoustic signature from an observer located someplace in the environment. The knowledge that the robot made use of in order to hide its own auditory emissions as a sound source included environmental obstacle maps, sound source locations and directivity models. Although this information about the auditory scene was provided *a priori* for experiments in this chapter, all this knowledge could be obtained using the tools described in Chapter 4. Furthermore, while the current implementation uses a heuristic to estimate masking effects due to a single source, we have also demonstrated (Section 6.3.1) that this heuristic can be simplified using the physics of sound flow through the environment to model effects on the listener. The algorithmic tool that can make this combination of information easier, ray-tracing, comes from the general set of sound propagation tools included with the framework in Chapter 3. In general, the work demonstrated in this chapter suggests the inclusion of more physics-based estimation, as well as possible variations in sound functions over

time (Section 6.3.2) should make hiding of the robot in ambient noise more effective under a variety of auditory scenes.

The work presented in Chapter 6 is particularly interesting when compared to the work completed in Chapter 5. In Chapter 5, the same set of tools were applied to both the acoustic monitoring task and the improved signal-to-noise ratio task as in this chapter. The robot needed environmental obstacle maps, sound source locations and directivity models, to determine differences between the believed state of the environment and the measured state of the environment. The only difference in the use of those tools was that for the stealthy approach scenario, the robot was the sound source, and for the acoustic monitoring scenario, the robot was the listener. Otherwise, the physics of sound propagation through an environment do not change, and the tools useful for gaining that knowledge are potentially appropriate no matter what the robot is. The distinction between sound source and listener is really only important in choosing the perspective. If the robot is listening, then the focus should be on the sound sources. In contrast, if the robot is the sound source, then the focus should be on the listener. This is the answer to the third, and final, sub-question posed in Chapter 1 – “How does acoustical awareness change with control over the source vs. the receiver?” The emphasis changes from the target being the listener to the target being the sound source, but the basic principles of sound propagation, and tools useful in gathering information about the auditory scene are the same.

## **CHAPTER 7**

### **ACOUSTICAL AWARENESS FOR HUMAN-ROBOT INTERACTION**

This dissertation has so far focused on medium to long duration sounds, which a robot can map out and identify on a regular basis. Using these maps, the robot can construct plans of action for future robotic movements, whether to investigate a sound source to collect more information (Chapter 4), make predictions about the environment (Chapter 5), or move to areas of loud noise in the environment (Chapter 6). Unfortunately, while these sounds certainly make up a significant portion of the auditory scene, they are by no means the only types of noise present in the environment. Speech, for instance, is a very common transient noise in home and office environments. Given that it is a known, and expected, sound in the environment, a robot may be able to map out where it most commonly occurs, but predicting when and how it will interfere with an application may be difficult. Similarly, transportation noise, such as that generated by planes, trains, and automobiles may be a very common part of the acoustic landscape that a real robotic application will have to be able to handle.

In this chapter, we explore the application of acoustical awareness to a more dynamic auditory scene. While the medium-to-long duration noises are still present in the environment, and are important for an acoustically-aware robot to adapt to, unexpected transient noises also have a significant effect on the application. The domain for this task is Human-Robot Interaction. The robot application is a mobile information

kiosk. The information kiosk is primarily a vocalization application, where a robot uses speech to communicate with a human listener. In a dynamic auditory scene, it is the responsibility of the robot to adapt its speech and non-speech behaviors to maintain intelligibility under a variety of acoustic conditions. As with previous work, there is a significant knowledge-based acoustical awareness component to the application in identifying where best to move the robot with respect to auditory scene. Speech, however, being a short duration vocalization itself, can be significantly masked by transient noises in the environment. Therefore, in addition to the knowledge-based awareness, the robot also needs to have a significant reactive acoustically-aware component to handle transient noise in real-time. The model for this short duration interaction is human speech behavior in the presence of ambient noise.

The remainder of this chapter is covered in four sections. The first section describes our vision of an acoustically-aware information kiosk, modeled after human behavior in dynamic auditory scenes. The second section then describes our current implementation of this vision on a real robotic platform. The third section goes beyond just the information kiosk algorithm to discuss general rules in incorporating in knowledge and, therefore, reaction to different types of sound in the environment. Finally, the fourth section summarizes the chapter.

The interactive information kiosk was originally researched in cooperation with Derek Brock at the Naval Research Laboratory in Washington D.C. The implementation was originally published at the 2<sup>nd</sup> Annual ACM/IEEE International Conference on Human-Robot Interaction [Martinson and Brock 2007].

## 7.1 A MODEL OF HUMAN ACOUSTICAL AWARENESS

The purpose of an information kiosk, traditionally, has been to provide information about the environment to interested people. The types of kiosks differ dramatically. A very simple kiosk might just relate the day's weather conditions, or list the set of departing flights at an airport. A more advanced kiosk could be a computerized map, where people use a mouse, keyboard, or touchscreen to read reports about different objects on the map. At the farthest end of the spectrum, even people could be considered as a type of mobile information kiosk prepared to answer an arbitrary set of questions to the best of their abilities. Within this large range, our vision of a robotic information kiosk fits somewhere between a stationary computerized map and the extreme of a person. An interested participant speaks the title of a story or object that he or she would like to have information about, and then the robot uses text-to-speech (TTS) to read aloud the pre-compiled story matched to that title. Such an interface may be of particular use in environments or to particular subjects where constantly reading a screen is not possible (e.g. blind people, or people simultaneously performing some other task). The challenge to this vision is a dynamic auditory scene. A typical TTS interface is very tough to comprehend, even when accompanied by visual instructions, when in the presence of large volume or large changes in ambient noise. The clue to overcoming these interface difficulties lies in human-human interaction.

When people communicate with each other, they can achieve relatively high comprehension levels using speech under a variety of acoustic conditions. Within a single auditory scene filled with many types of noise sources (including other human conversations, cars, telephones, and machinery hums), people can still communicate.

The reason for this success is that people are acoustically-aware of the auditory scene, adapting their speech and non-speech behaviors to maintain intelligibility. Confronted with an arbitrary auditory scene, the ways in which people react to the environment are extremely varied. Some of them are conscious behaviors requiring thought and analysis of their human partners' capabilities, while others are more reactive, happening without our necessarily even being aware of the adaptation. In general, however, the ways in which people compensate fall into three categories, each requiring some knowledge of the auditory scene in order to be successful: (1) the word choice is altered to increase contextual cues and repetition, (2) the speech waveform is altered to maximize intelligibility in the presence of ambient noise sources, and (3) people move about the environment to minimize their noise exposure and maximize speech intelligibility. The following sub-categories describe each of these adaptations in more detail, and discuss what a robot could do using current technology.

### **7.1.1 ALTERED WORD CHOICE TO INCREASE CONTEXT**

When the word level intelligibility of a spoken sentence is low due to poor acoustic conditions, people can still understand the meaning of sentences when there is context to the utterance [Goldstein 2007]. Examples of such context that have been demonstrated to improve intelligibility of synthesized speech are topic cues and familiar phrases [Drager and Reichle 2001]. If the listener knows the topic ahead of time, then they may consciously or subconsciously be listening for a different vocabulary. Similarly, familiar phrases are easier to recognize because they change for the length of the phrase the active vocabulary being listened for by the receiver.

In addition to the research on listener intelligibility, there is also evidence that a speaker takes advantage of this contextual intelligibility improvement. In selecting their utterances, speakers use a model of the listeners' capabilities, adapting to their needs as best as possible in order to maximize intelligibility and knowledge transfer. People repeat words or phrases more often, change sentences syntactically, use different forms of expression, etc. to raise contextual cues and generate common ground in order to communicate whatever they wished to communicate. Furthermore, this model appears to change over time as discrepancies between the model and real life become apparent, forcing further adaptations in speaking patterns [Bard and Aylett 2000].

Given the current state of technology in natural language interfaces, these contextual adaptations to a speech interface are very difficult for a robot to do properly. An acoustically-aware robot has information about the auditory scene, and can estimate word level intelligibility of its synthesized speech from the ambient noise conditions (predicted and measured) in the room, and, specifically, at the location of the listener. However, the robot is unable, currently, to make sensible changes to an arbitrary text without substantial human intervention.

An action that an acoustically-aware robot can take, however, is to be attentive for requests for repetition from the user. Maybe a listener needs to have something repeated. Maybe they would like more information on a given subject, which could be obtained through a web search. Providing some simple interaction tools to the user is not only feasible, but allows the listener to get more contextual information when it is needed, and forego it when not needed. In an even more aware scenario, the robot could actually use its knowledge of the auditory scene to make predictions about when intelligibility is

likely to be low, and ask the user if they need to have anything repeated, rather than simply waiting for a request.

### **7.1.2 CHANGING THE WAVEFORM TO MAXIMIZE INTELLIGIBILITY**

The second category of human adaptation to the auditory scene is to improve the intelligibility of the speech waveform itself. This phenomenon is commonly called “clear speech” or “Lombard speech” [Junqua 1993]. In the presence of noise or stress, people reflexively adapt their own speech, changing the shape and tightness of their vocal tract to produce a different volume, prosody, pitch, and/or timbre of their speech. The resulting signal can be more intelligible to the human auditory system than an unmodified signal under the same noise conditions [Langer and Black 2005]. Unfortunately, this effect has so far proven difficult to duplicate with computers.

The phenomenon of clear speech is best understood in the speech recognition community. For a human listener, these adaptations to the speech waveform increase intelligibility, as they are expected in the presence of masking noise. For a computer speech recognition system, however, these same adaptations are difficult to model, and cause decreased word recognition rates. As such, there are a number of researchers currently working on alternative features for recognizing phonemes under a variety of noise and stress conditions [Bou-Ghazale and Hansen 2000; Boril et al. 2006].

Speech synthesis research has run into similar problems in increasing intelligibility under adverse noise conditions. For limited speech synthesis, there has been some success in using recordings of people talking under different noise conditions. By analyzing the noise present in the environment, a computer can then pick which

recording of the desired utterance would be best understood by a human listener using the similarity between the current and recorded noise conditions [Langer and Black 2005]. Efforts at the waveform generation level which use fully synthesized speech, however, have so far been able to duplicate this success.

Despite these failures, there are still adaptations that an acoustically-aware robot can make to improve its auditory presentation. One such adaptation is to change the volume. Although prosody, pitch, and timbre improvements have been difficult to duplicate so far, automatic volume adjustments have been applied successfully to a number of commercial systems such as car stereos [BOSE 2007]. Another adaptation is to stop talking when volume adjustments are no longer feasible. For instance, when a military jet flies overhead, producing very loud ambient noise levels, people commonly pause the conversation and wait for the noise to end. Either of these responses, changing volume or pausing, allow a robot to act in the presence of noise to preserve intelligibility.

### **7.1.3 CHANGING THE SPEAKER'S POSITION**

The final category of human adaptation to the auditory scene is to adjust the position of the speaker relative to the listener and/or the noise sources. If there are just too many sources of interference, be it masking noise, or simply distractions, people do not have to remain stationary. They can employ gestures, move closer to the listener, face the listener as much as possible, and if all else fails, move to someplace else where there is less interference. With the possible exception of gestures (depending on the available hardware), our acoustically-aware robot can make the same decisions. It can use knowledge-based awareness of the auditory scene to select ideal interaction positions

in the environment prior to any human-robot interaction. It can also follow a moving receiver to maintain a reasonable conversation distance, and, when noise levels are simply too much to cope with, work with the human participant to move to another, quieter location in the environment.

## 7.2 ROBOTIC ADAPTATIONS

From this knowledge of acoustically-aware actions taken by humans, we developed an acoustically-aware information kiosk using a mobile robot base. Given some of the technological limitations discussed above, only some of the proposed actions could actually be implemented. In this section, we will discuss five such actions based on human adaptations for improving intelligibility in a dynamic auditory scene. These actions include:

- Listening and responding to simple spoken commands from a human partner controlling the flow of information.
- Adapting the volume of the speech output in response to changing noise conditions and a speaker' distance from the robot.
- Pausing for speech and excessive noise that can interrupt reading and distract the listener
- Rotating to face the listener, maintaining the interaction and orienting the loudspeaker in the correct direction.
- Move to another location when sound levels stay too high in the current location.

Each of these five actions has been fully implemented as part of an acoustically-aware information kiosk application on a B21r mobile robotic platform. While there

have been no formal evaluation studies of the completed interface to date, the interface has been informally tested by other members of the laboratory, visitors to the lab, and news media.

### **7.2.1 ROBOT**

The information kiosk application was developed for the B21r (Figure 7.1) robot equipped with:

- An overhead microphone array for ambient noise monitoring. This array is composed of (4) Audio-Technica AT831b lavalier microphones mounted at the top of the robot. These microphones are each connected to battery powered preamps mounted inside the robot body and then to an 8-Channel PCMCIA data acquisition board.
- A monitor mounted at eye-level to display for new users the available topics the robot may talk about and list the set of speech commands the robot can understand. Figure 7.1 (Right) demonstrates an example visual interface discussing navy ships. Other interfaces featuring current news briefs, biographies of interesting people, and NRL robotics projects have also been developed.
- A speaker and internal amplifier to allow the robot to speak at a variety of volumes to a human listener.
- A stereo vision system for person tracking.
- A SICK LMS200 to be used with continuous localization [Schultz and Adams 1998] to provide reliable robot pose estimates.

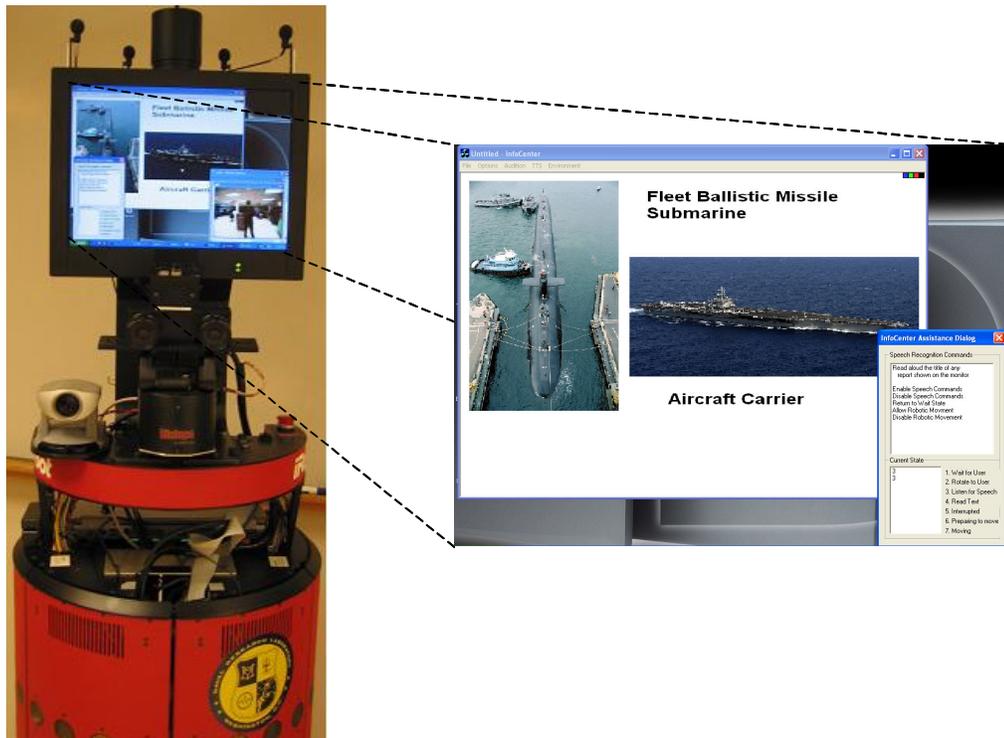


Figure 7.1. (Left) B21R robot from iRobot, outfitted with a four microphone overhead array, bi-clops stereo vision system, and monitor for visual feedback. (Right) A screen capture of what is shown on the monitor while the Information Kiosk is running (not including the person-tracking interface). The larger window lists stories that the robot can talk about (here, 2 Navy ships). The second, smaller, window lists the set of available speech commands the user can employ at any given time.

In addition to the above hardware the interface also uses a separate wireless microphone headset to capture speech for speech recognition tasks performed using freely available speech recognition software (Microsoft SAPI 5.1). This separate microphone was necessary to get reliable speech recognition results, as the overhead array was not appropriate for speech recognition tasks. Future implementations, however, should be able to replace the wireless microphone with a directional microphone mounted on the robot body. Combined with other efforts by the robot to always face the user, a directional microphone should provide reasonable speech recognition results with a minimum of additional effort by a user.

### Visual Person Tracking

The vision system on the robot is an actuated TRAC Labs Bi-Clops. The rotatable stereo camera provides dual color images from which depth information (Figure 7.2, top) can be extracted. Combined with face detection software (created using OpenCV [Bradski et al. 2005])<sup>12</sup>, the robot can use the camera to track, localize, and follow a detected person through a 180 degree arc in front of the robot (Figure 7.2, bottom). To start tracking a person, the individual's face needs to be at least 20 pixels in width, which corresponds to a distance of roughly 1.5-m from the robot. After initializing a track, however, the camera can continue to provide depth information up to 3 meters away from the robot.

---

<sup>12</sup> The face tracking software was implemented by Vlad Morariu



Figure 7.2. Stereo vision results. (Top) Scene depth as estimated by stereo vision. (Bottom) The graphical output of the person tracking interface, showing a person's face and hands being successfully tracked.

## Speech Detection

Before the cameras can be used to find a person, however, the robot needs to first identify and localize on speech sounds in the environment. To detect speech sounds, we calculate the first 2 mel-frequency cepstrum coefficients (Appendix B.4 describes the creation of MFCCs from sampled data) for each microphone in the overhead array. Each coefficient is averaged across all microphones, and then compared to an environment dependent threshold. While this speech detection system is relatively simple, and prone to errors when classifying a single sound sample, it works well over time to augment other auditory and vision sensors tracking humans in the environment.

### **7.2.2 RESPONDING TO SPEECH COMMANDS**

The purpose of this adaptation is to allow for a human user to control the flow of information from the robot, as suited their individual needs, so as to add some level of redundancy to the application even though a full natural language interface is not feasible. Using a commercial speech recognition package (SAPI 5.1), the robot recognizes a small of set phrases spoken into the wireless microphone (to be replaced in the future). These phrases include titles of stories the robot can talk about plus a set of phrases for disabling options in the interface and controlling the flow of text during an interruption. The different commands available to a listener are:

- “Repeat the last line”
- “Repeat from the beginning”
- “Continue where you stopped”
- “Change to a new subject” or “Stop talking on this subject”

Each of these phrases effect the flow of information similar to what they mean. For more detail about how exactly they affect the implementation, details are available in pseudocode format in Appendix D.

### 7.2.3 CHANGING THE VOLUME

After a user selects a topic by speaking the title of the topic, the robot reads a corresponding paragraph or two of information aloud, sentence by sentence. Before each sentence, the robot measures the current level of ambient noise in the room, and the distance at which the listener is standing to estimate an ideal volume at which to speak in order to maintain the desired intelligibility levels. The louder the ambient noise in the environment, the louder the robot needs to speak. Similarly, the farther away the listener stands, the louder the robot needs to speak. Conversely, if the ambient noise volume or the listener's distance decreases, the robot should lower its volume to avoid being excessively loud. Ambient noise levels in the room are measured by the microphone array, and the distance to the user is measured by the stereo vision system.

Since each of these variables, volume and distance, are measured in different units, they needed to be combined somehow before applying them. Assuming that each variable is related exponentially to the volume (this should actually be dependent on the specific amplifying hardware being used), we can combine these variables together using the equation of an ellipse. Equation 7.1 demonstrates how this heuristic relates the volume output to each of these two variables, ambient noise volume and distance.

$$SF^2 = \left( \frac{V - MinN}{MaxN - MinN} \right)^2 + \left( \frac{D - MinD}{MaxD - MinD} \right)^2 \quad \text{Equation 7.1}$$

$$output\_volume = SF * (MaxV - MinV) + MinV$$

Where [MinD-MaxD] is the range of distances over which a person interacting with the robot might be expected to stand, [MinV-MaxV] is the range of ambient noise that the robot might encounter in this environment, V is the current noise level, and D is the current distance from the robot to the human listener (as detected by the visual system). A more detailed description of the volume adjustment implementation and how it is used within the overall HRI application can be found in Appendix D.1.

#### **7.2.4 PAUSING FOR INTERRUPTIONS**

Sometimes, transient noise from the surrounding environment requires that the robot stop reading for some period of time. Even though the robot can raise the volume at which it speaks, some masking noise is too loud to talk over. If such an event should occur, and the robot continued to read its text, the robot speech would not be intelligible during the event, thereby losing any knowledge being transferred at that time and frustrating the listener. A robot with some knowledge of the primary entities in the auditory scene, however, knows the maximum volume at which it can speak, can estimate how much the listener can hear, and can then choose to pause during periods of excessive noise. When ambient noise levels finally return to a reasonable level (i.e. the robot predicts that speech is again intelligible to the listener), the robot can resume speaking. To alert the listener to the fact that it is about to continue speaking, the robot starts the next sentence by saying, “As I was saying...”

Another source of interruptions for a robot speech interface are other people. Unlike general auditory events, which reduce intelligibility by masking the speech signals, other speech in the environment does not necessarily reduce intelligibility below

acceptable levels for a human listener. However, if that speech is directed at the listener, then the user's attention will be diverted from the robot and focused on the new human speaker. In this case, to preserve intelligibility, a robot should recognize that it is no longer the focus of attention and should pause until it regains its audience. The audience is assumed to have returned when the human listener issues a speech command to the robot (Section 7.2.2).

The specifics of the implemented algorithm for pausing in the presence of noise or speech are provided in Appendix D.2.

### **7.2.5 ROTATING TO FACE THE LISTENER**

A person arriving at the information kiosk might approach from any angle. Although the robot ultimately uses the vision system to track its listener, it first waits for the person to say something and uses the speech detection and localization tools discussed earlier to determine the direction it should turn to face. Then the vision system is initialized and the biclops camera takes over the job of continuous tracking. As the camera is actuated, it can rotate independent of the robot body to follow the person through arcs of up to 90 degrees in each direction. However, for intelligibility and ease of use, it is best to restrict this range to 30 degrees or less in each direction, and rotate the robot body when the person moves too far to one side or another. This algorithm is described in pseudocode in Appendix D.3.

The purpose of rotating the robot is twofold. First, it promotes ease of use because it frees the listener of the need to remain in place while interacting with the kiosk and it

places the flat-panel monitor, which displays information topics and speech commands for using the system, in front of the user.

The second purpose of rotating the robot is to maintain the desired intelligibility levels. The loudspeaker on the robot is not omni-directional, meaning that its apparent volume changes with the angle of the perceiver. Consequently a person standing to the side of the robot will not hear its speech output as well as a person standing directly in front of it. By not allowing the listener to stand too far to either side of the robot (i.e., by rotating the robot to face the listener after more than 30 degrees of angular displacement), an acoustically-aware interface minimizes the effects of loudspeaker directionality on volume levels and general intelligibility at the listener's location.

#### **7.2.6 MOVING THE ROBOT**

The last action that can be taken by an acoustically-aware robot to improve the intelligibility of its speech output is to move the robot. When confronted with excessive amounts of noise that have not faded after some period of time, a robot should recognize that the sound is not going to disappear, and that a new location in which to hold the interaction is necessary. As human listeners typically stand very close to the robot during an interaction, this work assumes that the noise levels heard by the human are comparable to those heard by the robot. Therefore, reducing the ambient noise exposure on the robot should also reduce the noise exposure on the human listener, so the robot can use its own knowledge of recently sensed data and combine it with knowledge of the auditory scene to quickly pick a quieter location in the environment and move. The

effectiveness of this relocation process using the B21r was already reported on in Chapter 5 (Section 5.3.1) under improving the signal-to-noise ratio.

When confronted with a new medium-to-long duration sound source, the robot also needs to take into account whether or not it is currently involved in an interaction. Depending upon the answer to this question, the robot may need to take slightly different actions:

- **No Ongoing Interaction**

When there is not ongoing interaction, the robot can follow the same sequence of steps as those outlined in Section 5.3.1 for relocating the robot:

*Step 1.* Estimate volume

*Step 2.* Identify source direction

*Step 3.* Move the robot to localize the source

*Step 4:* Map the noise

*Step 5:* Identify a better location

*Step 6:* Move the robot

- **Currently Interacting with a Human**

When the robot is interacting with a human partner, then it needs to make two changes to this algorithm. The first change is that the robot needs to ask the human if they want to move, and only relocate if the human feels that this is appropriate. Otherwise, it is possible that this listener is not having any difficulties understanding the robot. The second change is the robot should not move during step 3 to localize the sound source. By

simply assuming the sound source is 1-m away, the robot can react quicker without annoying or frustrating the human user.

*Step 1.* Ask the user if they want to move

*Step 2.* Estimate the volume

*Step 3.* Identify the source direction

*Step 4:* Map the noise, assuming a sound source 1-m away from the robot

*Step 5:* Identify a better location

*Step 6:* Move the robot

The pseudocode for this implementation of acoustically aware relocation is provided in Appendix D.4. More details on individual steps can also be found in Section 5.3.1.

### **7.3 COMBINING TYPES OF AWARENESS**

The purpose of the acoustically-aware information kiosk is to provide information to people using speech. As such, it has to be able intelligible in the presence of a wide variety of sounds, including medium-to-long duration ambient noise, short duration ambient noise, and a separate category, speech sounds. Given the complexity of the environment, however, how can a robot effectively adapt to a changing auditory scene? The answer is inspired by peoples' reactions to acoustic interference when using speech. People are aware of their surrounding acoustic environments, whether consciously or not, and the actions that people take allow them to respond to each of the above categories of sound. Our robot interface should do the same.

For our acoustically-aware implementation of the information kiosk, the robot could adapt in five different human-inspired ways to the auditory scene. The choice of action, by necessity, depended on the type of interference from the surrounding environment. A finite-state-machine (FSM) describes the selection of these responses for a robot in the middle of an interaction with a human partner. Seen in Figure 7.3, the robot can change its volume, wait for conversations to end, pause for short duration noises, and ultimately select and move to a new location in the environment if a short duration noise persists for too long. Note that rotating to face the listener is not listed in this diagram. Since the stereo vision system rather than the auditory array is used to

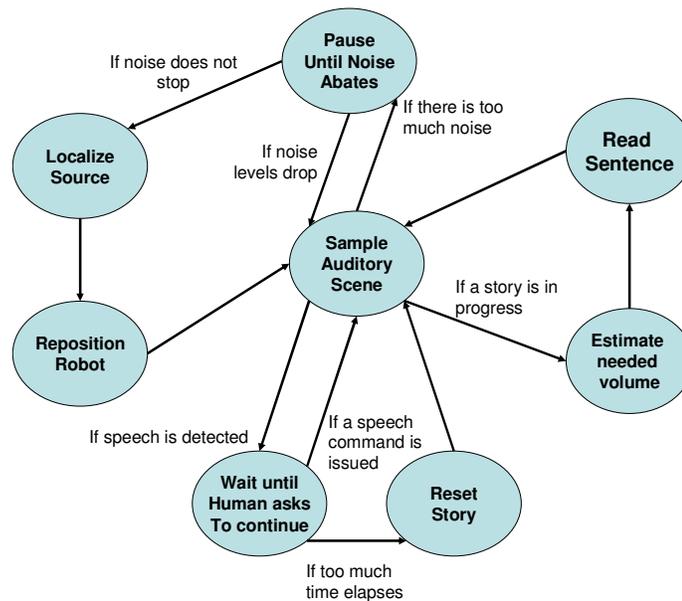


Figure 7.3. The sequence of steps the robot takes while reading a story to a human listener. Starting with the step in the center, the robot samples the auditory scene, and then if nothing is wrong, estimates a volume, reads a sentence and repeats. When excessive noise or speech is detected, the robot takes a different action to adapt its auditory output before reading the next sentence.

maintain the robot's orientation once a person has initially been detected, the process of rotating to face the listener is run in parallel with all other actions. Appendix D presents a pseudocode description of both this FSM implementation and the individual sub-components involved in the application.

In general, many of these adaptations are reactive in nature. This is different from previous applications where the robot always had time to first investigate the sound, determine where it is coming from, estimate models of sound propagation through the room, and then react appropriately. Although some medium-to-long duration sources do occur in most environments, which would remain active long enough to be detected and explored, many of the sounds being dealt with are transient and difficult to plan for. The robot needs to be able to react appropriately to any type of sound, regardless of its duration. The question therefore becomes how best to react to the unexpected? This is a classic question in mobile robotics, for which many different robot architectures have been developed, and which we will not go into here. Suffice it to say that the same problem exists in dealing with transient noise in auditory scenes. Some of it can be planned for, and some of it cannot. In this dissertation, we used an FSM to combine reactions to both short duration and medium-to-long duration sounds, but there are many other hybrid architectures available. Whatever the choice of architecture, the important thing is that the robot does not simply ignore the auditory scene when the auditory scene can affect the application, as is the case with applications involving speech.

In the long run, it may be possible to move even more of the currently reactive adaptations to the deliberative side of acoustical awareness by extending the predictive capabilities of the sound field framework described in Chapter 3 to the problem of sound

propagation over time. When a repeating sound function the robot knows very well is detected, the robot can make predictions about peaks and valleys in the sound function to adapt its speech patterns (or word choices) ahead of difficult acoustic conditions, or otherwise change its plan of action to accommodate the extra noise. For instance, if the robot knows that the machine in the next room will reach a crescendo in the next 5-10 minutes, then it can suggest early on that the conversation is moved to a different location to avoid a situation where intelligibility decreases slowly over time until the robot has to move anyways. Since such predictions will only work for sound sources that the robot is familiar with, the robot will still have to react somehow to unknown sounds. But the additional capabilities, which are currently beyond the computational limits of modern computers to perform in real-time, could prove very useful in integrating more knowledge about the auditory scene into an action selection mechanism.

#### **7.4 CHAPTER SUMMARY**

In summary, this work with an information kiosk has demonstrated the application of acoustical awareness to an important new domain: human-robot interaction (HRI). As a field in which there is much use of sound, both in audition and vocalization, it is only appropriate that we should have examined the application of sound propagation knowledge to the design of HRI applications. As is also appropriate, we found that people are the best model for this type of acoustically-aware interaction. When confronted with a dynamic auditory scene, people naturally adapt their speech and non-speech behaviors to maintain the conversation. A robot should do the same. By taking into account the effects of a dynamic auditory scene on a listener's comprehension of its

speech output, we built an application where an acoustically-aware robot could act in human-inspired ways to maintain intelligibility and, ideally, improve its user's interaction experience.

In addition to exploring a new domain, however, this chapter also touched upon the important topic of transient noise in the auditory scene. While previous applications discussed in this dissertation focused on medium-to-long duration sound sources in the environment, the information kiosk was forced to handle a wide variety of sounds, both in volume and duration. In order to respond appropriately to whatever came up in the auditory scene, we implemented acoustical awareness as part of a hybrid controller. This way, when the robot recognized it had time, it could acquire the information it needed about the scene to plan out its actions. But for the short duration sounds, it could quickly respond to preserve intelligibility.

With respect to the general problem of acoustical awareness, the use of a hybrid architecture to handle transient noise is an important addition. There has been significant work, as demonstrated in Chapter 2, in reacting to, or simply incorporating into an application, particular types of transient sounds from the environment. While much of the earlier work has been focused on speech sounds, there has also been work in detecting and using animal sounds, transportation noise, and music. The use of a hybrid architecture allows the robot designer to combine this body of research by others in the field of mobile robotics with the deliberative approach to acoustical awareness emphasized in this dissertation. Together, they provide a comprehensive picture for intelligently reacting to a wide variety of noises in the auditory scene.

## CHAPTER 8

### SUMMARY AND CONTRIBUTIONS

The use of sound with mobile robotics is fast gaining attention among researchers. Although vision has usually stolen the spotlight with the large quantities of data present in a stream of images, it does not show the complete state of the environment. In particular, vision only works in straight lines when nothing is between the target and the camera. Sound, in comparison to light, travels around and even through solid objects in the environment. For a robot, audition can signal the occurrence of significant events when the robot is facing the wrong way, or is not even located in the same room. Audition can also be used for diagnostics of systems that a camera cannot reach or see.

Unfortunately, despite these advantages, there has been only limited effort so far to incorporate audition into general robot navigation. The extent of most of the work to date has been event-oriented. Specifically, if the robot hears something that can be considered significant, it repositions itself to focus another sensor on the problem. This dissertation has argued that, while the event-oriented response can be valid, there are many situations in which having general knowledge about sound propagation through a room, i.e. an acoustical awareness, can allow a robot to more effectively adapt its navigational controller to perform tasks involving sound. With this knowledge available to it, an acoustically-aware robot can make predictions about the auditory scene, separating signals of interest from ambient noise, and dynamically adjusting its plans to monitor areas for the occurrence of an auditory event. A robot can also reposition itself with respect to areas of significant ambient sound, either to avoid the noise or move to it

in search of a good listening position. It can also make predictions about the effects of noise, ambient or robot vocalized, on other listeners and change its own behaviors to achieve the desired effect. Each of these is a specific scenario in which we have already demonstrated in this dissertation the successful use and advantages of being acoustically-aware. This list is not designed to be all-inclusive. Instead, it only demonstrates the wide variety of tasks to which acoustical awareness can be applied. In general, having knowledge about the auditory scene allows a robot more flexibility with which to accomplish the tasks that people desire of it.

## **8.1 HOW CAN ACOUSTICAL AWARENESS BE APPLIED TO MOBILE ROBOTICS?**

While the application of sound propagation to robot navigation may seem a good idea, the big question is how? This was the key question answered by this dissertation, providing direction on how this idea of acoustical awareness be effectively incorporated into a navigational controller. The answer was sub-divided into three subsidiary questions designed to make the result as general as possible for use by the wider robotics community: (1) What information or data about the auditory scene is useful to a mobile robot? (2) How can a robot gather this information? (3) And finally, how can a robot incorporate this information into its own navigational behaviors? The answers to each of these three questions are summarized in the following sub-sections.

### **8.1.1 WHAT INFORMATION IS USEFUL?**

The first subsidiary question asked, “what *a priori* information or sensory data are useful for a mobile robot performing an acoustic application?” This question was addressed primarily in Chapter 3. The answer to this question came from physics. Sound

generated by a source in the environment travels around the environment in a relatively well-understood fashion, bouncing off walls or objects in the room until its energy decays completely. In order to make perfect predictions about sound flow through the room, a robot needs a complete description of geometric and material properties of every sound source, every object, and every receiver in the environment. Since no one can ever have a perfect description, physicists and, more recently, acoustical engineers have been working on a number of different approximation frameworks using different assumptions and sets of information. Several of these frameworks are summarized in Chapter 3. The set of information useful to an acoustically-aware robot therefore depends on the choice of sound propagation frameworks.

Of these techniques, the sound fields framework stands out as particularly amenable to robotic deployment. The conceptual idea of sound fields uses superpositioning to break up the auditory scene into a number of different independent components. Unlike some of the other sound propagation estimation techniques, this is particularly advantageous to a robot where the set of information available is likely to vary wildly from application to application. The bare minimum for estimating sound flow through the environment is a sound source location. Then, as more information becomes available, either through robotic or human efforts, the estimates of sound flow can add volume, directivity, reverberation, transmission, etc. The sound fields framework is also particularly advantageous to mobile robotic deployment because of its adjustable computational complexity. Although the upper-bound on complexity can be quite large, the bare minimum can estimate sound levels at any location in the environment using a single equation without iteration. Given the wide variety of

applications and hardware requirements, the flexibility of the sound fields framework makes it ideal for robot use. More detail on exactly how the sound fields framework can be used to model sound propagation through an environment, and the set of information that a robot needs to model different aspects of the auditory scene is found in Chapter 3.

### **8.1.2 GATHERING KNOWLEDGE ABOUT THE ACOUSTIC ENVIRONMENT**

The second subsidiary question asked, “how can we combine sensory data from multiple sources to build effective representations of the acoustic environment?” This question was the focus of Chapter 4, where we explored a set of tools available to a mobile robot for storing, retrieving, and fusing together sensory data to gather acoustic knowledge about the environment. Even though the sound fields framework is flexible enough to work with a wide range of data, its accuracy varies with the amount and quality of the information that the robot has about the auditory scene. Therefore, adding the ability to autonomously gather a wide range of information about the acoustic surroundings increases the flexibility of the system even further. Now a robot can make use of *a priori* information when it is available, and gather additional data on its own as needed.

The set of tools discussed in Chapter 4 for autonomously gathering information about the environment were varied in purpose and origin. Many of the tools focused specifically on identifying properties of sound sources in the environment, since knowing about them is so critical to accurate sound propagation estimates. Two tools in particular, auditory evidence grids and a sound source discovery process, were developed for this dissertation to localize sound sources in the environment, and then estimate properties of

volume and directivity for each sound source. Another tool, mel-frequency cepstral coefficients (MFCC's), was developed by others for classifying sound source functions. This dissertation demonstrated, however, how MFCC's could be used successfully with the data from the sound source discovery process to detect different types of sound sources in the room from a moving robotic platform.

Other tools described in Chapter 4 for augmenting robotic knowledge about the auditory scene primarily focused on environmental effects. Robotic mapping of the obstacles in the environment, researched extensively by others in the robotic community, was demonstrably applied to the problem of reverberant field estimation (part of the sound field framework). Although there were a number of improvements to be made for quality, the use of ray-tracing with a basic evidence grid representation was shown to produce effects expected of a reverberant field, including reduced acoustic shadows and increased volume near walls or other hard surfaces. Future work has already been proposed in Chapter 6 for evaluating the effectiveness of these reverberation models when applied to a mobile robot.

The final tool described in Chapter 4 was noise mapping. These are maps created directly from samples of noise levels in the environment, without using a predictive sound flow framework such as sound fields. Acousticians have used noise maps for many years as a visual guide to sound flow through the environment. This dissertation applied them for the first time to sensory data collected autonomously by a mobile robot. For now, they are largely used as validation tools for the robot designer, but in the future they may serve as an autonomous method of verifying the local result of the sound fields

estimation framework, allowing a robot to identify areas of missing knowledge that should be investigated further to improve accuracy.

### **8.1.3 APPLYING ACOUSTIC KNOWLEDGE TO NAVIGATIONAL CONTROL**

The third subsidiary question asked, “how does acoustical awareness change with control over the source vs. the receiver?” Chapters 5-7 were dedicated to answering this question, by breaking the problem into parts: (1) How can acoustical awareness be applied to a auditory task? (2) How can acoustical awareness be applied to a vocalization task, and what are the differences? (3) How does the type of auditory scene affect the choice of control? The problems behind each of these subjects were explored by one or more applications, providing examples and some guidelines for applying knowledge about the auditory scene to robotic control.

#### Applied Acoustical Awareness - Audition

The problem of auditory task improvement was explored in two primary tasks described in Chapter 5, as well as in some of the acoustic knowledge gathering tasks described in Chapter 4. The two primary tasks were an acoustic monitoring task where a security robot kept track of changes to sound sources in the auditory scene, and an improved signal-to-noise ratio problem, where the robot sought to reduce the amount of ambient noise it was exposed to in order to improve signal quality. Between these two tasks, we demonstrated two different approaches to producing change in robotic movement due to knowledge about the auditory scene: dynamic re-planning, and map-based navigation.

Dynamic re-planning was employed in both the acoustic monitoring scenario and the robotic discovery experiments performed in Chapter 4. While performing the patrol task, the robot gathered a large collection of sampled auditory data. When the robot had time, it processed the data, using its knowledge of the auditory scene to separate signals of interest from uninteresting ambient noise. The acoustic monitoring task in Chapter 5 demonstrated how the robot could use this methodology to detect the presence of new sound sources or changes to existing sound sources in the environment. Then, knowing that something had been altered in the environment, the robot could use the tools presented in Chapter 4 to localize the new sound source or focus on existing sound sources in choosing where in the environment it should return to and investigate. Though not implemented in this dissertation due to computational and data limitations, the robot would ideally process the data in real-time, determining areas of likely change while still in the vicinity of the change, so that it could cover wider areas and reposition itself with a minimum of backtracking. The drawback to the real-time data analysis is that the robot may not have as much knowledge available to it when making a decision.

Map-based navigation was the second type of acoustically-aware robotic movement strategy investigated in Chapter 5. The scenario using this movement strategy was the improved signal-to-noise ratio task. In this scenario, the robot was seeking to minimize its exposure to ambient noise by either repositioning itself within the environment, or moving dynamically away from areas predicted to contain high ambient noise. To accomplish this task, the robot used its knowledge of the auditory scene to first build a map of the ambient noise levels in the room (using the sound fields framework), and then pick either a quiet stationary listening position or relatively low volume path

through the environment. The results of this movement strategy, however, were mixed. Using such a map, a robot could clearly improve its position over areas of high ambient noise, and do so more consistently than avoidance strategies that did not take advantage of knowledge about the auditory scene. There was not a significant improvement, however, in robot noise exposure in using a map to avoid areas of medium noise. While there was a consistent improvement with the map, the delta change in noise exposure was relatively small due to the reverberant field dominating all but the loudest areas of the room. As a result, this mixed performance suggested that the success of map-based movement strategies may be highly dependent on the specific situation for which they are used. If the robot's current situation indicates an exceptional need, such as a high volume sound source in close proximity, then they could be very useful. However, if the acoustic situation is not very difficult (i.e. the number and effects of ambient noise sources in the environment is small) then an estimated map of the auditory scene may not help the robot much. Such a situation may call for either more accurate modeling of the environment or alternative movement strategies, such as speeding up the robot to escape regions of moderate interference.

#### Applied Acoustical Awareness - Vocalization

Robot vocalization tasks were explored in two chapters: a stealthy approach scenario in Chapter 6, and a human-robot interaction problem, the acoustically-aware information kiosk, in Chapter 7. The information kiosk, however, will be summarized in the next section, due to its emphasis on environment type.

From a control standpoint, the stealthy-approach scenario used a similar map-based movement strategy to that used in the improved signal-to-noise ratio experiments. The difference, however, was that the robot was not trying to change the signal that it detected, but rather change the signal detected by an external observer listening for a robot. Therefore, when the robot constructed a map, it did not make predictions about what it was going to hear at different locations. Instead, the robot predicted what the observer would hear, given different robot locations. Since the observer remained stationary, the map remained 2-dimensional, and the best path through the environment was fairly obvious. But, with enough processing power, and modeling of user actions, this same map-based approach could serve as the basis for a cat-and-mouse scenario, where the listener actively avoids the robot.

For now, the stealthy approach scenario has only been tested with heuristics for estimating environmental masking of robot ego-noise. However, this dissertation also proposed an extension of the ray-tracing algorithm described as part of the sound fields framework to solve this problem in the more general case. Assuming that the ray-tracing predictions are accurate enough to mask the robot, the same algorithm should also work in reverse, allowing the robot to improve the signal quality perceived by a listener over regions of extraordinary sound.

From the overall standpoint of this dissertation, the important point of Chapter 6 was less the application, and more the similarity between the domains of audition and vocalization. In all tasks using knowledge about the auditory scene, maps were made of the ambient noise in the environment. The information used in these maps for both the vocalization and auditory applications included sound source locations, directivity,

volume, and obstacle maps, each of which could be acquired using the tools in Chapter 4. The only real difference between these domains was the identity of the receiver. For the vocalization applications, it was someone or something located someplace other than the robot. For the auditory applications, it was the robot. In both cases, the receiver could possibly move, the actions of the robot could affect what was perceived, and the underlying sound-fields framework could make predictions about how the robots actions would affect the observer.

#### Differences in Control Due to Environmental Factors

The final point of discussion in the applications part of this dissertation was on the effects of the environment type on robotic control. For most of this dissertation, the emphasis was on medium-to-long duration sources. These are a very common set of sources in human environments (fans, fountains, heavy machinery, radios, etc) that can usually be expected to remain unchanged while the robot sampled the room. Because these sources remain relatively static, a robot can build action plans to detect, investigate, and avoid such sources.

In Chapter 7, however, we explored a scenario in which speech was used by a mobile robot to interact with a human partner. The robot, an information kiosk, was located in a dynamic auditory scene where there were not only medium-to-long duration sources, but also transient, short-duration noises. The robot's goal was to preserve intelligibility of its speech output in the face of this dynamic auditory scene, but it did not always have time to respond, as had the previous scenarios. Medium-to-long duration sources could be treated as they were treated earlier, mapping them out and selecting a

better location from a map. Transient noise, however, appeared and disappeared quickly enough that the robot did not have time to gather information, map the result, and plan an alternative behavioral response. In the long run, as processor speeds improve, the sound fields framework may be extended to model some of these sounds over time, but for this application, behavioral responses were hard-coded into the robot controller by the designer.

The important lesson learned from the acoustically-aware information kiosk was how to respond to different types of ambient noise. Ultimately, knowledge-based approaches and reactive approaches to dealing with an auditory scene are both types of acoustically-aware control. As described in Chapter 3, acoustically-aware navigation can be performed by reactive, deliberative, or hybrid controllers. The choice of which controller to use should depend on how noise exposure affects the application. If the application is particularly sensitive to transient noise, such as was the case with our speech application, then a reactive approach to handling these sound may be the best choice. If medium-to-long duration sounds are a problem, however, and the robot has time to acquire some knowledge about the environment, then a more deliberative approach can be taken, predicting the effects of robotic movement, and selecting the action with the best possible outcome. If both types of sounds significantly affect the application, then a hybrid controller can be implemented to handle both types of sound.

## 8.2 CONTRIBUTIONS

This dissertation has explored the use of knowledge-based acoustical awareness in guiding mobile robotic navigation and decision-making. In support of this stated goal, three contributions have been made to the field of robotics:

- **Conceptual Framework**

A sound propagation framework based on the theory of sound fields from physics was validated in this dissertation for use by a mobile robot. The room-level maps of the auditory scene that can be constructed using this framework allow for a wide range of available information, and demonstrably improved robotic performance in multiple applications involving real robots. The sound fields that were specifically explored as part of this dissertation included:

1. *Direct Field* – estimates of the direct field provide a quick, computationally simple approach to estimating the sound levels across a wide environment from a limited amount of information.
2. *Reverberant Field* – building estimates of reverberant sound in the room requires more knowledge about the environment than estimating the direct field, but the potential for more accurate representations is also higher. Ray-tracing was explored in this dissertation as an approach to representing both the direct and reverberant fields. Although the incorporation of ray-tracing into robotic applications remains future work, this dissertation demonstrated how a robot could

create models of the reverberant field using ray-tracing with robot gathered information about the auditory scene.

- **Tools for Gathering Acoustical Information**

A set of tools was identified in this dissertation for gathering knowledge to use with the sound fields framework. Some were developed for this dissertation [#2-3], while others were developed in other research areas and applied for the first time to a sound propagation estimates with a mobile robot [#1, #4-6]. These tools include:

1. *Spatial Likelihoods* – an algorithmic tool for estimating the direction of sound detected by a microphone array (Section 4.1.2).
2. *Auditory Evidence Grids* – a representation for combining spatial likelihoods over time and space to localize sound sources in the environment (Section 4.2.1)
3. *Volume and Directivity Estimation* – an algorithmic approach to identifying the volume and directivity of a sound source from a collection of sampled data (Section 4.2.2).
4. *Mel-Frequency Cepstral Coefficients* – an algorithmic representation of the source sound function that can be identified by a mobile robot, and used to detect changes to the auditory scene (Section 4.2.4).
5. *Evidence Grid Representations of Obstacles* – although the creation of evidence grids has been explored in much greater detail elsewhere by others, this dissertation demonstrated that they could also be applied to

the sound fields framework for estimating the reverberant field using ray-tracing (Section 4.3).

6. *Sampled Data Noise Maps* – a representation of the auditory scene that was derived directly from the sampled data. This representation complements the sound fields framework by providing a comparative metric against which missing information may potentially be identified (Section 4.4).

- **Examples and Guidelines for Implementation**

Four different scenarios were developed and explored as part of this dissertation work. The range of these scenarios demonstrates the versatility of being acoustically aware:

1. *Identifying Change in the Auditory Scene* – by using knowledge gathered while patrolling the environment in conjunction with the sound fields framework, a robot can identify changes to existing sound sources, as well as the presence and location of new sound sources in the environment (Section 5.2). The test domain of this application is robotic security.
2. *Improving the Signal to Noise Ratio* – using maps of the auditory scene generated by the sound fields framework, a robot can reduce its exposure to ambient noise in the environment (Section 5.3). The test domain of this application is robotic security.
3. *Hiding the Acoustic Signature* – when stealthily approaching a target, a acoustically-aware robot can use maps of the auditory scene to hide

its own ego-noise from an observer (Chapter 6). The test domain of this application is robotic surveillance.

4. *Information Kiosk* – a robotic information kiosk combined information from a number of sensory modalities in a hybrid acoustically-aware architecture to handle both medium-to-long duration and transient noise sources in the environment (Chapter 7). The domain of this application is Human-Robot Interaction.

In exploring these scenarios, we have also encountered on multiple occasions some general design guidelines for the application of acoustical-awareness to mobile robotic applications:

1. Even limited information about the auditory scene is still better than no information. This was demonstrated particularly well in Section 5.3.1, where just the knowledge of the sound source position was enough to significantly improve performance. All of the applications, however, showed some improvement with limited information.
2. The incorporation of robot ego-noise into sound propagation models can substantially improve performance. Without the inclusion of robot ego-noise models in Section 5.2, changes to the environment could not have been identified. Other applications, including acoustic hiding in Chapter 6, also demonstrated how robot ego-noise could impact performance. Therefore, when available, a model of robotic movement may have a significant effect.

3. Applications where robot are generating sound and applications where robots are listening to the auditory scene can make use of the same sound fields framework, with the same robot gathered knowledge. The only real difference involves for whom the auditory scene representations are being created. This guideline was demonstrated in Chapters 5 (robot as listener) and 6 (robot as sound source).
4. Finally, in application to robotic systems, the sound fields framework is designed to intelligently handle significant medium-to-long duration interference from ambient noise. When there is a significant transient noise component to the auditory scene, however, a reactive awareness (Chapter 3) can be used in conjunction with the deliberative aspects of being acoustically aware to improve overall performance (Chapter 7). This allows the large body of research performed by others in reacting to auditory cues (Chapter 2) to be integrated together with the work in this dissertation to achieve a comprehensive awareness of the auditory scene.

### 8.3 CONCLUSION

The title of this dissertation is *Acoustical Awareness for Intelligent Robotic Action*. The focus of this work was on improving the quality of robotic applications using sound by adding more domain knowledge about sound propagation into the decision making process. This dissertation combined knowledge from other domains with that of mobile robotics, sometimes developing new algorithms where none were

currently available, to accomplish this goal. Given the insight, the examples, and the means, this dissertation has now enabled others in the field of mobile robotics to utilize knowledge of sound flow in their own acoustic applications.

## Appendix A - SOFTWARE DESIGN

The robotic experiments discussed in this dissertation all made use of the same underlying acoustically aware system. Although two different robots were involved in these experiments, an Activ-Media Pioneer2-dxe vs. the iRobot B21r, the hardware/software configuration was largely the same. The only real difference was in the choice of controller, Player v1.6.5 for the Pioneer robot vs. Wax [Schultz et al. 1999] for the B21r. Otherwise, even the commands sent to Player or Wax were identical. This appendix, therefore, describes in more detail this hardware and software configuration used for all experimentation in this dissertation. Where there are differences between robots, the emphasis is on the implementation for the Pioneer robot.

### A.1. HARDWARE

Our specific hardware implementation made use of 4 different computers, one of which was dedicated to sampling the auditory scene, and three of which were used in processing sensor readings and controlling the mobile robot. Figure A-1 demonstrates how these four computers were networked together.

- **Internal Computer**

The Pioneer2-dxe robot was equipped with a 700-Mhz internal computer, connected by a 10-Mhz wireless connection to other computers in the room. The onboard computer ran RedHat Linux v7.2, and was responsible for (1) passing sensor data over the wireless to a desktop computer, and (2) running the Vector-Field-Histogram controller for moving while avoiding obstacles.

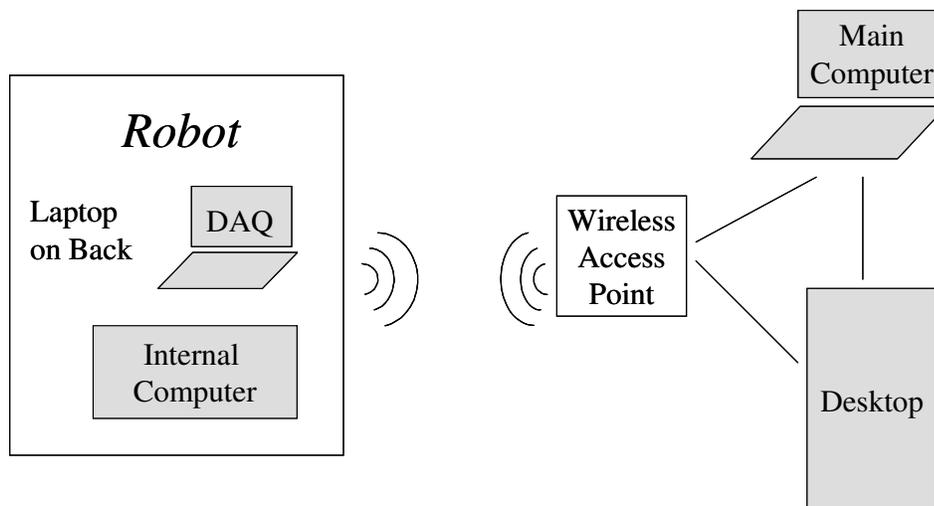


Figure A.8.1. The network configuration of the 4 computers used in the acoustically aware experiments. 2 computers on or in the robot were connected by wireless access point to a desktop machine and the main computer.

- **DAQ Computer**

An 800-MHz laptop running Windows-XP was mounted on the Pioneer robot, behind the SICK LMS so as to gather samples as requested in real-time from the microphone array. On the Pioneer robot, 4 Audio-Technica ATR35S series lavalier microphones were mounted to a box attached to the robots back (above the wheels). On the B21r, 4 Audio-Technica AT831b series lavalier microphones were mounted to a metal frame attached to the top of the robot (above and behind the monitor). Both robots then used a Measurement Computing PC-CARD-DAS16/16 to collect data from all 4 channels at 8192-Hz and pass it over the wireless network to the Main Computer running the database application.

- **Desktop Computer**

A 900-MHz dual processor desktop computer running Fedora Linux was dedicated to localization and path-planning. This computer ran Player v.1.6.5, running the ‘amcl’ and ‘wavefront’ drivers locally and accepting the sensory data from robot’s internal computer with passthrough drivers. The desktop computer also ran a Player controller program based on the “simple” example program provided with Player v1.6.5. This controller program was responsible for accepting target goals and localization requests from the taskmanager program over a socket interface.

- **Main Computer**

The computer responsible for running most of the acoustically aware programming was a 2.0-GHz Celeron laptop with 512-Mb of RAM, running Windows XP Professional. This computer collected and stored samples from the auditory interface, processed the data to build representations of the auditory scene, and provided high-level control for guiding robotic movement.

## **A.2. SOFTWARE PROCESSES**

The software implementation of acoustical awareness, as used in this dissertation, was based on the idea of separable executable processes communicating via socket interfaces. For instance, processing sensory data was handled separate from map building, which was handled separately from robot control. This separation of processes made the design and debugging phases simpler, by allowing processes to be seamlessly

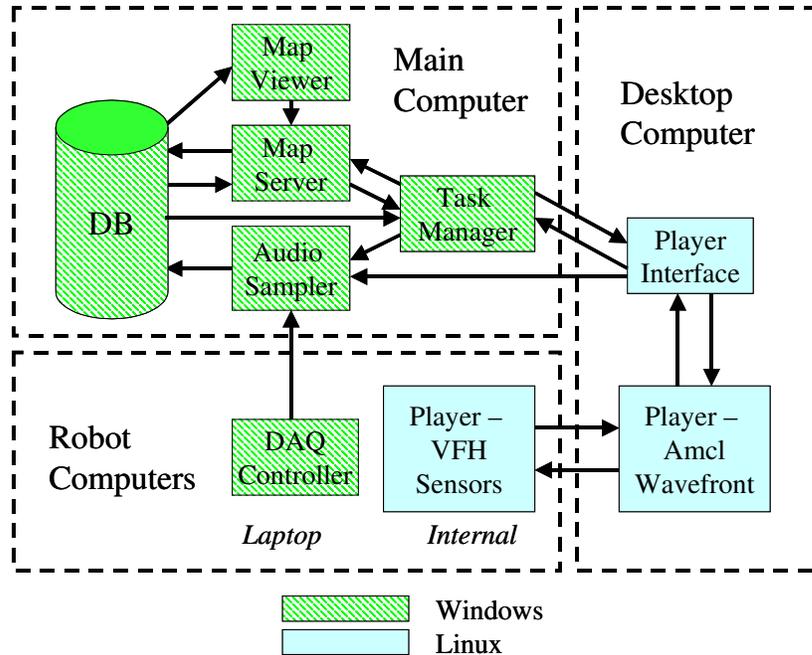


Figure A.8.2. The software configuration used for acoustically-aware navigation in this dissertation. A total of 9 different executable programs were involved in collecting information from onboard sensors, storing and manipulating that information, and then converting representations into robotic navigational commands.

distributed across different computers, and allowing for greater flexibility in extending the capabilities of the mobile robot to cover new application areas. At the conclusion of this dissertation, the final set of processes included one database application, 6 c++ programs, and 2 instances of the Player robot control environment. Figure A-2 shows how each of these processes were distributed, and how information flowed between the different executables. The following bullets then discuss in more detail the purpose of each process in the acoustically-aware system.

- **Database – implemented in Microsoft SQL Server 2000**

Central to the system design used in this dissertation was a database. The database acted as a blackboard, allowing processes to store all incoming

sensory data for future use, as well as store intermediate results such as spatial likelihoods, sound pressure levels, likely sound source locations, and all maps created by the system. These data could then be retrieved at later dates to rebuild evidence grids or noise maps as many times as possible with different combinations of measured data.

- **DAQ Controller – c++ program (Windows)**

The DAQ (Data Acquisition) controller software, located on the DAQ Computer mounted on the back of the robot, was designed to sample as needed from the microphone array. As the available laptop was relatively slow, no processing was actually performed in this process. Instead, when a sample was requested, it would collect as many seconds of data from each microphone and transmit it over the wireless network. The controller could even apply a FIR filter to the incoming data if requested.

- **Audio Sampler – c++ program (Windows)**

The purpose of the audio sampler program was to request samples from the DAQ controller on a regular basis (every 250-msec) and store those samples to the database. Whenever the audio sampler requested a sample from the DAQ, it also requested the current position of the robot from the Player interface so that processes could identify the position and time at which samples were recorded. Also, as discussed in Chapter 4, some sampling strategies require sampling only when the robot is moving. To enable this functionality, the task manager software could turn sampling by the audio sampler on and off.

- **Map Server – c++ program (Windows)**

The map server program was designed to handle all map building tasks, including the creation of auditory evidence grids, sampled noise maps, predicted noise maps, and sound source directivity. At the request of the task manager software, or the map viewer software, the map server took the necessary samples from the database, performed all necessary mathematical calculations on them (including determining sound pressure level and spatial likelihoods), and created a new map from those samples. The created map was stored in the database for later re-use, as well as returned to the requesting program for immediate use.

- **Task Manager – c++ program (Windows)**

The task manager software was the high-level controller behind acoustically aware movement. The task manager controlled when the robot samples the auditory scene by communicating with the audio sampler. The task manager controlled the creation of new maps of the environment from sampled data through the map server program. Finally, the task manager controlled where the robot moved by specifying waypoint targets to the Player robot server (by way of the Player interface program). Appendix B includes pseudocode descriptions of the different robotic control strategies employed by the task manager throughout this dissertation.

- **Player Interface – c++ program (Linux)**

The player interface served as a network interface between the task manager and the Player robot server. This program ran on the same machine as the robot server, so the task manager had to communicate with it over the wireless network. There were three movement commands that the player interface accepted: (1) move the robot to a particular location in the environment, as specified by  $\{x,y,\theta\}$ ; (2) move the robot in a particular direction at a given speed, allowing a robot to follow a gradient noise map through the environment; and (3) stop all robotic movement. A fourth command also requested the current location of the robot. This command was used by both the task manager and the audio sampler programs.

- **Player Components/Drivers (Linux)**

With the exception of the work in developing auditory evidence grids (Chapter 4), most of the robotic experiments performed in this dissertation used the Pioneer2-dxe mobile robotic platform made by ActivMedia Robotics. To test, communicate with and control this platform, we used the player/stage robot environment. This environment was chosen because it already contained basic obstacle avoidance, path-planning, and localization behaviors for the Pioneer2-dxe robotic platform equipped with a SICK Laser Measurement System on the top-front. To allow for others to more easily duplicate our experimental work, we provide below the Player configuration files used for all of the robotic experiments.

Robot Internal Computer Configuration

```
driver (  
  name "p2os"  
  provides ["odometry::position:0"]  
)
```

```
driver (  
  name "sicklms200"  
  resolution 100  
  range_res 1  
  provides ["laser:0"]  
  port "/dev/ttyS2"  
  rate 38400  
  pose [0.1 0 0]  
)
```

```
driver (  
  name "vfh"  
  provides ["position:1"]  
  requires ["position:0" "laser:0"]  
  safety_dist 0.1  
  distance_epsilon 0.15  
  angle_epsilon 20  
  free_space_cutoff_0ms 1200000.0  
  weight_current_dir 0  
  min_turn_radius_safety_factor 0.3
```

```
max_speed 0.1
```

```
)
```

#### Desktop Configuration

```
driver (
```

```
name "passthrough"
```

```
provides ["position:0"]
```

```
remote_host "128.61.119.103"
```

```
remote_port 6665
```

```
remote_index 0
```

```
access "a"
```

```
)
```

```
driver (
```

```
name "passthrough"
```

```
provides ["position:1"]
```

```
remote_host "128.61.119.103"
```

```
remote_port 6665
```

```
remote_index 1
```

```
access "a"
```

```
)
```

```
driver (
```

```
name "passthrough"
```

```
provides ["laser:0"]
```

```
remote_host "128.61.119.103"
```

```

remote_port 6665

remote_index 0

access "r"

)

driver (

  name "mapfile"

  provides ["map:1"]

  filename "maps/fast_lab11.pgm"

  #filename "maps/fast_lab9.pgm"

  resolution 0.03

  negate 1

)

driver (

  name "mapfile"

  provides ["map:0"]

  filename "maps/new_HandMap8.png"

  resolution 0.03

  negate 1

)

driver (

  name "amcl"

  init_pose [-3.3 -2.1 0]

  init_pose_var [.1 .1 .2]

```

```

alwayson 1

update_thresh [0.1 5]

provides ["localize:0"]

requires ["odometry::position:0" "laser:0" "laser::map:1"]

)

driver (

  name "wavefront"

  provides ["planner:0"]

  requires ["position:1" "localize:0" "map:0"]

  safety_dist 0.1

  distance_epsilon 0.2

  angle_epsilon 10

)

```

- **Map Viewer – c++ program (Windows)**

The map viewer program is not really necessary for autonomous movement. However, development of reliable control requires being able to duplicate the efforts of the robot. For this purpose, a separate map viewer program acted as an interface to the map server in lieu of the task manager, allowing both the creation of maps and the recall of existing maps from the database. The map viewer could be run simultaneously with the task manager to monitor the status of the robot's efforts to model the auditory scene. Maps were displayed in real time with minimal processing overhead.

### A.3. DATABASE DESIGN

The design of the database that supports our acoustically implementations was based on the three primary acoustic entities discussed in Chapter 3: sound sources, paths, and receivers. Each of these three primary entities is related to each other: sound sources and receivers are found at some location in a particular environments, and sources are detected by (and possibly dominating) samples collected by a particular receiver. In addition to these relations, each of these three primary entities is also involved in the creation of another composite entity, the auditory scene. The goal of our database implementation was to implement these known relationships in tabular format. Figure A.8.3 summarizes these relations graphically. The squares in this figure represent a

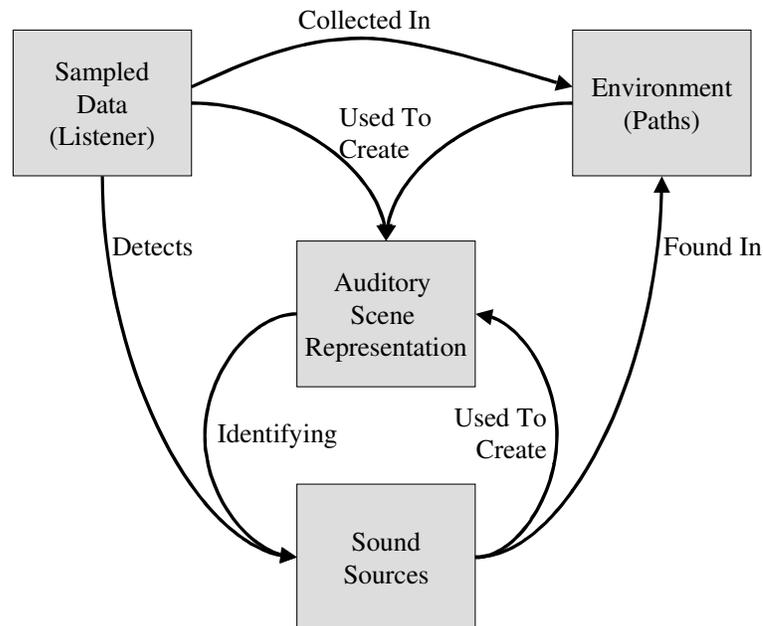


Figure A.8.3. Groups of information in the database are grouped by the entities they relate to: sound sources, environments, listeners, and representations of the auditory scene. The arrows represent the types of relationships between these groups.

group of tables in the database describing a particular entity. The arrows then indicate relationships between the entities. The direction of the arrow indicates how to read the sentence, i.e. a sound source is “found in” a particular environment.

In the following sub-sections, we discuss in greater detail the shape of each of these table groupings, illustrating how all the necessary information for the sound fields framework and other acoustically aware tasks (such as sampled noise maps) can be stored in a database.

### **A.3.1. SAMPLED DATA**

The sampled data entity was designed to contain all of the known information about, and collected by, the receiver. This includes information about specific microphones, microphone arrays, samples collected by the microphones, algorithmic transformations of the sampled data, and collections of sampled data that occur over similar periods of time. Each of these concepts are represented by their own table in the database (illustrated in Figure A.3 by Microsoft SQL Server):

- **Data Session**

As the auditory scene changes over both location and time, it is important to group samples by when and where they were collected. The data session table performed this grouping, connecting samples collected over a single run together, and identifying the environment (Loc\_Name) in which the samples were collected. The data session also stored active process information about whether these samples have been searched for sound sources yet, and if the session has been completed already.

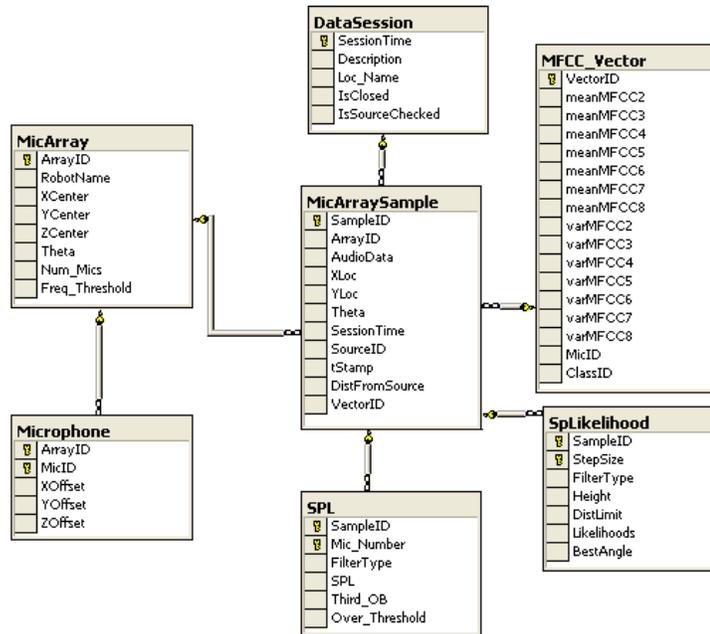


Figure A.8.4. A graphical description of the tables/relationships that make up the sampled data entity in the database. This figure was generated by Microsoft SQL Server 2000, where the database was implemented.

- **Microphone**

The microphone table represented a single physical receiver. This table recorded information about which array the microphone belonged to, and where the microphone was located within that array.

- **Microphone Array**

The microphone array was a group of synchronized microphones that may be used for estimating spatial likelihoods or sound pressure levels. The microphone array was most important in grouping together different microphones, and storing the centroid of the array with respect to a particular robot's centroid (i.e. the offset from the robot's estimated

location). Since each robot had a different microphone array (microphone types and positions), each robot would have its own row in the microphone array table.

- **Microphone Array Sample**

This table stored samples collected by a particular microphone array. The audio streams from all microphones in the array were stored together, along with the time of the sample and the robot's position, as estimated by the Player robot server, at the end of sample collection. This table also stored information about the source that is expected to dominate this sample, as determined by the distance from a source active during this data session.

- **MFCC Vector**

The mel frequency cepstral coefficient (MFCC) was used to relate samples to particular sound sources. In this table, the 2<sup>nd</sup> through 8<sup>th</sup> coefficients were stored for this purpose, including the mean and variance across some number of samples. The VectorID stored in the MicArraySample table determined the set of samples that contributed to this vector.

- **Sound Pressure Level (SPL)**

The SPL table was a mathematically derived transformation of a sample from a single microphone. Therefore, for each sample in the MicArraySample table, there were  $N$  samples in the SPL table, where  $N$  corresponded to the number of microphones in the array. This representation of SPL stored both the overall sound pressure level, as well

as the Third Octave Band sound pressure level. This table also stored whether or not any filters were used in removing particular bands of noise, and whether the sample had any infinity values indicative of sampling error.

- **Spatial Likelihood**

Like the SPL table, the Spatial Likelihood table (SpLikelihood) was a mathematically derived transformation of a single sample in MicArraySample. Unlike SPL, however, the entries in this table did not vary with the number of microphones in the array. Instead, spatial likelihoods could be created with different step sizes to allow for different levels of precision when building auditory evidence grids. This table also stored for each sample the height at which the spatial likelihood was being estimated, the distance limit to which spatial likelihoods were being calculated, whether or not any filters were used in estimating the spatial likelihood, and the most likely angle to a sound source, as determined by this spatial likelihood.

### **A.3.2. ENVIRONMENTS**

The environment entity is represented in this database as a group of three tables: obstacle maps, waypoints, and obstacle rectangles. These three tables allowed a robot (task manager) to retrieve obstacle maps and lists of waypoints for these maps from the database for use in path planning. These tables also allowed the map server to create

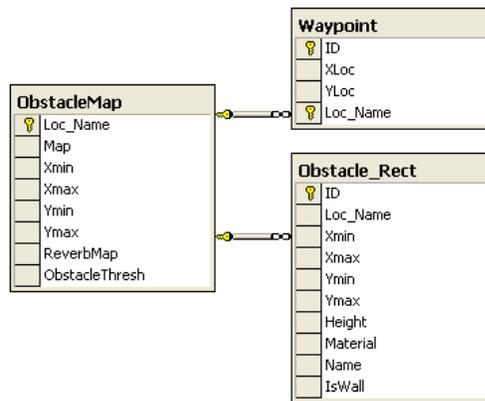


Figure A.8.5. The three tables describing environmental information in the database.

estimates of the reverberant field using ray-tracing. Figure A.5 demonstrates what information is stored in each of these tables, and how they are related to each other.

- **Obstacle Map**

The obstacle map table stored maps of the obstacles in the environment. The “Map” attribute stored the map created by the pmap utility [Howard 2004] from the laser data collected by the robot. Also, the table stored the size of the environment (in terms of min/max), the threshold at which grid cells in “Map” contained an obstacle, and any hand created maps (such as those without small obstacles) used for estimating reverberation in place of the robot created map.

- **Obstacle Rectangle**

The purpose of this table was to store information about particular obstacles or walls in the environment, so as to improve the accuracy of the ray-tracing results. Although this table has been implemented in the

database, it was not used for any of the applications described in this dissertation. The implementation of an enhanced ray-tracing estimation process that would use surface information remains future work.

- **Waypoint**

Waypoints are merely stored paths for a robot to follow through an environment. Although the patrol scenario in Chapter 5 created these waypoints dynamically from obstacle maps of the environment, the work in Chapter 4 used hand created waypoint paths. Such hand created paths were stored in this table for repeated use, identified by the environment that the robot needed to patrol.

### **A.3.3. SOUND SOURCES**

The sound source was another relatively simple entity to represent in the database. This dissertation only used three tables in its work: a table representing possible source locations in the environment, a table representing confirmed sources, and a table for representing sound functions. Figure A.6 shows the configuration of each of these three tables. What this configuration for the sound source entity did not store, however, was temporal information. Future work in modeling sound sources, as discussed in Chapter 6, should include information about how sources change over time. Do these sources repeat? And if so, is there a representation of the sound function that can estimate this repetition? As these are not simple questions, we did not attempt to address them in the current database implementation.

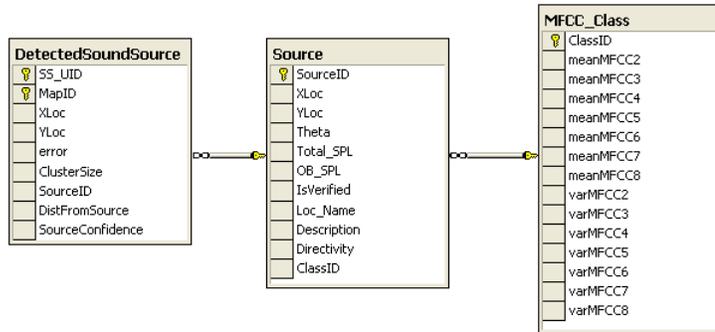


Figure A.8.6. Tables storing sound source information in the database.

- **Detected Sound Sources**

Entries in the detected sound source table were those locations indicated by clusters in an auditory evidence grid as being likely to contain a sound source. The map server was responsible for entering detected sound source information into the system when it built an auditory evidence grid. Along with the location of the potential sound source, the detected sound source table also stores the size and error criteria for the cluster, to be used with the Iterative Source Hunter for separating real sound sources from false positives.

- **MFCC Class**

Following the investigation of a sound source, a sound function was identified for that source as the mean MFCC Vector recorded during the investigation. The MFCC Class table stored that information to be used in classification. This table was not merged with the Source table because it was possible for two sound sources to have similar sounding source functions, and therefore, overlapping MFCC Class vectors.

- **Sound Sources**

Although there already existed a detected sound source table in the database, an additional table was necessary to separate real sound sources from possible locations. Once a sound source had been investigated, or verified by hand, it was stored in this table along with its volume, directivity, and a pointer to its MFCC Class. This table also allowed a human developer to describe sound sources, and indicate whether or not they were discovered by a robot or verified by a human.

#### **A.3.4. REPRESENTATIONS OF THE AUDITORY SCENE**

The final group of tables in the database represented the composite auditory scene. Shown in Figure A.6, this group of tables is actually simpler than it appears. The only table of real interest is the “Map” table, which stored all kinds of maps created by an acoustically aware robot. This same table stored sampled noise maps, auditory evidence grids, and predicted noise maps in the same field. It simply separated those maps by type, the height the scene being modeled and the grid cell size of the map. All of the remaining tables in this auditory scene entity then defined the set of information used in building each map, setting constraints on sampled or derived data to be included. The following set of bullets summarizes the different sets of information that our map server could use in creating a map:

- **Area**

Maps could be created with four different types of area constraints. The area constraint could restrict the set of samples included in the map to

those collected in a particular area. The area constraint could also restrict samples to those that pointed at a particular area using the spatial likelihood measurements. Then, with either previous constraint, the map could be built from those samples that fit the constraint, or those samples that did not fit the constraint (i.e. the inverse). Multiple area constraints were OR'd together, so, for instance, multiple target areas constraints would include all samples that pointed at either of those two areas.

- **Data Session**

The data session constraint limited the set of samples included in the map to one or more selected data sessions.

- **MFCC Class**

The MFCC Class constraint limited the set of samples included in the map to those that belonged to one more or selected MFCC Classes.

- **Noise Type**

In the final implementation, the Noise Type constraint had mostly been replaced by the MFCC Class constraint. It was included here, however, because the original paper on Auditory Evidence Grids [Martinson and Schultz 2006] reported that auditory evidence grids could be created from just the set of samples determined to contain speech. If an MFCC Class vector were available for defining speech, then it could be used instead of the Noise Type constraint. However, the approach taken in the original paper was to simply use those samples that passed a particular noise

threshold. This constraint, therefore, specified that only those samples in a particular preset noise range were to be included in the map.

- **Sound Sources**

In an auditory evidence grid or sampled noise map, the sound source constraint could be replaced with an area constraint. Maps could be created with samples that were collected in the vicinity of a known sound source (or not), as well as from samples that pointed at a known sound source (or not). With predicted noise maps, however, the sound source constraint indicated which set of sound sources should be included the sound fields model of the auditory scene.

- **Time of Sampling**

The time of sampling constraint was used to limit the range of times from which collected samples are included in an auditory evidence grid or sampled noise map. This constraint is only particularly useful in modeling sound sources that are not on throughout an entire data session. This way, a map can be created of only those times where the source is known to have been enabled.

### **A.3.5. SUMMARY**

Each of these four entities were then connected to each other through a series of relationships. Figure A.7 shows all of the tables used in the database along with the relationships between them. Sampled data was connected to sound sources and environments. Sound sources were connected also connected to the particular

environment, and to sampled data through MFCC results. Finally, representations of the auditory scene were connected to many of the tables in the database to allow for a wide variety of scene creation mechanisms.

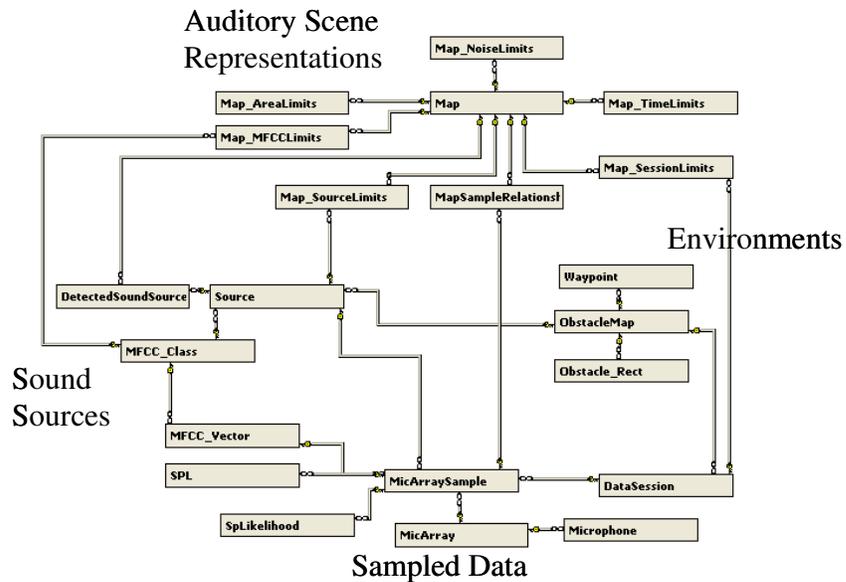


Figure A.8.7. Summary of the database implementation used in this dissertation. All of the tables seen in previous sections are included.

## Appendix B - KNOWLEDGE GATHERING TOOLS

In this appendix, we concentrate on providing algorithmic descriptions of the acoustical knowledge gathering tools used in this dissertation. This includes implementation level descriptions of: (1) spatial likelihoods, (2) auditory evidence grids, (3) determining the source volume and directivity, (4) mel-frequency cepstral coefficients, (5) building maps of the direct field, (6) the ray-tracing algorithm for direct and reverberant field estimates, and (7) creating sampled data noise maps. From an implementation standpoint: all of these algorithms were used by the Map Server program so as to provide accurate maps for the Task Manager, all sampled data used by each of these algorithms came directly from the database, and all intermediate results were stored back to the database for future use by the Map Server.

### B.1. SPATIAL LIKELIHOODS

The spatial likelihood implementation used here was developed from [Mungamuru and Aarabi 2004]. The theory behind them and their use with auditory evidence grids is described in greater detail in Chapter 4.

#### Variables

- **Number of Microphones,  $numMics$**
- **Desired Height of the Estimate,  $H$**
- **SampleSize, 2048**
- **signal <sub>$i$</sub>**

This is the sampled data retrieved from the  $i^{th}$  microphone in the array.

- **mic\_pose <sub>$i$</sub>**

This is the [x,y,z] position of the ith microphone in the array, relative to the array center.

- **Spatial Likelihood Output, *SpLikelihood***

*SpLikelihood* is an 18x18 matrix, representing a 6x6-m<sup>2</sup> area of 0.3x0.3-m<sup>2</sup> gridcells, and the cross correlation energy from the signal associated with a source being at each of these locations in the environment.

- **Frequency, *w***

*w* contains the frequency of each element in the FFT output, in [rad/sec]

**Pseudocode**

/\*\* the spatial likelihood is estimated pairwise... So for each microphone pair, estimate the chance of there being a sound source at each of the desired locations, and then sum the results across all microphone pairs \*/

1. **for** each pair of microphones [i,j]

2. [f1real,f1imag] = FFT of signali

3. [f2real,f2imag] = FFT of signalj

4. ffreal = f1real.\*f2real + f1imag.\*f2imag /\*\* The operator “.” indicates an element-wise multiplication of arrays \*/

5. ffig = f2imag .\*f1real - f2real.\*f1imag

/\*\* estimate the weights for the phase transform \*/

6. 
$$G = \frac{1}{\text{magnitude}(f1_{real}, f1_{imag}) * \text{magnitude}(f2_{real}, f2_{imag})}$$

7. **for** cell [a,b] in the array

/\*\* estimate the time delay for this microphone pair \*/

8. [x,y] = real coordinates of cell [a,b], relative to the array center

9. 
$$d_1 = \sqrt{(x - mic\_pose_i.x)^2 + (y - mic\_pose_i.y)^2 + (H - mic\_pose_i.z)^2}$$

10. 
$$d_2 = \sqrt{(x - mic\_pose_j.x)^2 + (y - mic\_pose_j.y)^2 + (H - mic\_pose_j.z)^2}$$

11. 
$$TD = (d_1 - d_2)/343 - \text{m/s};$$

/\*\* build generalized cross correlation variables \*/

12. 
$$X_{real} = G * (\cos(-w * TD) * ffreal - \sin(-w * TD) * ffig)$$

13. 
$$X_{imag} = G * (\cos(-w * TD) * ffig - \sin(-w * TD) * ffreal)$$

/\*\* perform trapezoidal integration across the half sample size \*/

14. 
$$S_{real} = 0.5 * \sum_{i=2}^{1024} [(w[i] - w[i-1]) * (X_{real}[i] - X_{real}[i-1])]$$
15. 
$$S_{imag} = 0.5 * \sum_{i=2}^{1024} [(w[i] - w[i-1]) * (X_{real}[i] - X_{real}[i-1])]$$
16. 
$$SpLikelihood[a][b]+ = magnitude(S_{real}, S_{imag})$$
17. **end for**
18. **end for**

### B.1.1. ESTIMATING THE BEST ANGLE

The estimation of the most likely angle from the Spatial Likelihood result was performed after the spatial likelihood had been completed. For this work, the estimation process uses a gaussian smoothing filter on all grid cell angles, to estimate the energy at 1-degree intervals. Another approach for determining angle would have been to re-estimate the energy values at a number of set angles and constant distance from the array.

#### Variables

- **The Spatial Likelihood result, *SpLikelihood***  
 an 18x18 matrix, representing a 6x6-m<sup>2</sup> area of 0.3x0.3-m<sup>2</sup> gridcells, and the cross correlation energy from the signal associated with a source being at each of these locations in the environment.
- **Grid Cell Angles, *Th***  
 The angle from the center of the array to each of the grid cell centers in the spatial likelihood result.
- **Standard Deviation,  $\sigma$**   
 Used a standard deviation of 10-degrees for the smoothing function.
- **Angular Increment, *ang\_increment***

The angular increment indicates the delta angle between successive angular estimates. This work used an increment of 1-degree.

### **Pseudocode**

1.  $P = \text{zeros}[360];$
2. **for**  $\text{ang} = 0: \text{ang\_increment}: 2\pi$
3.      $\text{num} = 0$
4.      $\text{den} = 0$
5.     **for** each cell  $[a,b]$  in SpLikelihood
6.          $\text{theta} = \text{Th}[a][b] - \text{ang}$ , normalized to  $-\pi \leq \text{theta} < \pi$
7.         
$$W = \frac{e^{-\text{theta}^2/2\sigma^2}}{\sigma\sqrt{2\pi}}$$
8.          $\text{num} += W * \text{SpLikelihood}[a][b]$
9.          $\text{den} += W$
10.     **end for**
11.      $P[\text{ang}] = \frac{\text{num}}{\text{den}}$
12. **end for**
13.  $\text{Best\_Angle} = \text{ang}$ , where  $P[\text{ang}]$  is maximized

### **B.2. AUDITORY EVIDENCE GRIDS**

The auditory evidence grid is an adaptation of the evidence, or occupancy grid to auditory localization information. As used in this dissertation, the input to the auditory evidence grid is a set of spatial likelihoods. The set of likelihoods in particular that are used for this purpose, however, depends on the need of the application. The application chooses some criteria for creating the auditory evidence grid, and the creation algorithm then extracts the necessary set of spatial likelihoods from the database. Chapters 4 and 5 provide examples of both collecting the necessary sampled data, and then determining which samples should be included in the auditory evidence grid.

### **Variables**

- **Auditory Evidence Grid, AEG –**

A map of size  $N \times M$ , initialized to zero at all locations

- **AEG gridcell size, *stepsize***

All gridcells were assumed to be square. Most maps were of 0.3-m to each side.

- **AEG Range, [*MinX-MaxX,MinY-MaxY*]**

The range of area described by the auditory evidence grid.

- **SessionTime**

the time at which the source was investigated, as indexed in the database

- **Set of Samples, *samples***

This is the set of samples to be included in the creation of this auditory evidence grid. This set includes the following information as sub-variables:

$[x,y,\theta]$  - location of the robot when the sample was collected

*BestAngle* – the most likely angle towards a sound source, as determined by the spatial likelihood

*SpLikelihood* – spatial likelihood for this sample. The spatial likelihood covers an area of  $6 \times 6\text{-m}^2$  around the robot location, in an  $18 \times 18$  square grid.

These samples can be pulled from the database using the following SQL statement (variables are indicated by quotation marks):

```
SELECT Sp.SampleID, MAS.XLoc, MAS.YLoc, MAS.Theta,  
       Sp.BestAngle, Sp.StepSize, Sp.DistLimit, Sp.Likelihoods  
FROM SpLikelihood Sp, micArraySample MAS
```

```
WHERE Sp.SampleID=MAS.SampleID
```

```
AND abs(Sp.StepSize-“stepsize”)<0.01
```

```
AND MAS.SessionTime in “SessionTime”
```

Additional sampling area limitations, such as those used in Chapter 5, can be included with another constraint:

```
MAS.XLoc>= “MinX” AND MAS.XLoc<= “MaxX”
```

```
AND MAS.YLoc>= “MinY” AND MAS.YLoc<= “MaxY”
```

- **Set of Samples with no calculated math, *samp\_noMath***

Samples in this program are always in the database as they are retrieved from the DAQ without any spatial likelihood or SPL information. Therefore, prior to their use in building an auditory evidence grid, we have to identify those samples which do not yet have the necessary mathematical results stored in the database, determine the spatial likelihood for these samples, and store the results back in the database. Since auditory evidence grids are the only place that need spatial likelihoods, this is done as the first step in their creation. The SQL code for retrieving the set of samples with no math from the database is as follows:

```
SELECT *
```

```
FROM micArraySample MAS
```

```
WHERE SampleID not in (
```

```
SELECT MAS.SampleID
```

```
FROM SpLikelihood Sp, micArraySample MAS
```

```

WHERE Sp.SampleID=MAS.SampleID
AND abs(Sp.StepSize-“stepsize”)<0.01
AND MAS.SessionTime in “SessionTime”)
AND MAS.SessionTime in “SessionTime”

```

- **Excluded Source List, *S***

When building auditory evidence grids, this dissertation sometimes wanted to exclude previously detected sources from being included in the next version of the auditory evidence grid. For this reason, a list of excluded sources was needed, which included the following information:

[x,y] – the location of the source being excluded

*radius* – the size of the region around the source that was also being excluded. This was typically a radius of 1-m.

- **Angle Increment, *ang\_increment***

When the best angle was calculated, it was only estimated at regular intervals of 1 degree. The *ang\_increment* describes the angular increment in radians, i.e. 0.02-rad.

Pseudocode

```

/** first step is to build the spatial likelihoods for all samples that do not yet have a
spatial likelihood entered into the database */

```

1. **for** each sample *k* in *samp\_noMath*
2.     calculate spatial likelihood
3.     store the result to the database
4. **end for**
5. update *samples*
6. **for** each sample *k* in *samples*
7.     **for** each cell [a,b] in *samples[k].SpLikelihood*
8.         isGoodCell = true;

```

9.      *** now check each of the excluded sources, and make sure that this
10.     sample does not point at any of them ***/
11.     for each source  $A$  in  $S$ 
12.          $d_s = \sqrt{(S[A].x - samples[k].x)^2 + (S[A].y - samples[k].y)^2}$ 
13.          $\beta = \arctan\left(\frac{S[A].y - samples[k].y}{S[A].x - samples[k].x}\right)$ 
14.          $\alpha = \arctan\left(\frac{S[A].radius}{d_s}\right)$ 
15.          $\Delta ang = \beta - (samples[k].BestA + samples[k].\theta)$ 
16.         normalize  $\Delta ang$  to between  $-\pi \leq \Delta ang < \pi$ 
17.         if  $|\Delta ang| \leq (\alpha + ang\_increment)$ 
18.             isGoodCell = false;
19.         end if
20.     end for
21.     *** only continue if not pointing at an excluded source ***/
22.     if isGoodCell == true
23.          $[x_l, y_l]$  = local coordinates of grid cell, relative to the robot's
24.         position
25.         *** translate the local coordinates to global***/
26.          $rad = \sqrt{x_l^2 + y_l^2}$ 
27.          $th = \arctan(y_l / x_l)$ 
28.          $x_{global} = sample[k].x + rad * \cos(sample[k].\theta + th)$ 
29.          $y_{global} = sample[k].y + rad * \sin(sample[k].\theta + th)$ 
30.         *** we need to scale the likelihood stored in the database, because
31.         the calculated value is in terms of energy ***/
32.          $K_1 = \left( \frac{\min(samples[k].SpLikelihood) - \max(samples[k].SpLikelihood) * 0.1 / 0.95}{1 - 0.1 / 0.95} \right)$ 
33.          $K_2 = (\max(samples[k].SpLikelihood) + K_1 * (0.95 - 1)) / 0.95$ 
34.          $prob_{a,b} = (samples[k].SpLikelihood[a][b] - K_1) / (K_2 - K_1)$ 
35.         *** use log-likelihoods to update the correct cell in the map
36.          $[i, j]$  = grid cell containing  $[x_{global}, y_{global}]$ 
37.          $AEG[i][j] = \log(prob_{a,b} / (1 - prob_{a,b})) + AEG[i][j]$ 
38.     end if
39. end for
40. end for

```

### B.2.1. IDENTIFYING CLUSTERS IN THE AUDITORY EVIDENCE GRID

The purpose of the auditory evidence grid is to localize sound sources in the environment. By itself, however, the auditory evidence grid is just a likelihood map. To convert that into sound source position estimates, we need to identify peaks in the auditory evidence grid. The algorithm we use for this purpose is nearest-neighbor clustering. After thresholding the map at some value combining neighboring grid cells together forms clusters. The resulting set of clusters describes the most likely positions to contain a sound source.

Variables

- **Cluster List,  $C$**

$C$  is a queue that is initialized with the coordinates/values of all grid cells in the auditory evidence grid which exceed some threshold. In this dissertation, this threshold was always 1.0, or approximately 75% likely. For each cluster in  $C$ , the following information was tracked and updated:

$\mu_w$  – weighted centroid of the cluster

$\mu_u$  – unweighted centroid of the cluster

$energy$  – summation of all component grid cell values

$error$  – the variance

$count$  – the number of grid cells contained in this cluster

$nodes$  – the positions [x,y] of all grid cells contained within this cluster

- **AEG gridcell size,  $stepsize$**

All gridcells were assumed to be square. Most maps were of 0.3-m to each side.

### Pseudocode

1. **for** each cluster  $i$  in  $C$
2.     **for** each remaining cluster  $j$  in  $C$
3.         **if** any node in  $C[i]$  is less than 0.4-m from a node in  $C[j]$   
         $/**$  then add the two clusters together  $*/$
4.             
$$C[i].mu_w.x = \frac{C[j].mu_w.x * C[j].energy + C[i].mu_w.x * C[i].energy}{C[j].energy + C[i].energy}$$
5.             
$$C[i].mu_w.y = \frac{C[j].mu_w.y * C[j].energy + C[i].mu_w.y * C[i].energy}{C[j].energy + C[i].energy}$$
6.             
$$C[i].mu_u.x = \frac{C[j].mu_u.x * C[j].count + C[i].mu_u.x * C[i].count}{C[j].count + C[i].count}$$
7.             
$$C[i].mu_u.y = \frac{C[j].mu_u.y * C[j].count + C[i].mu_u.y * C[i].count}{C[j].count + C[i].count}$$
8.              $C[i].energy += C[j].energy$
9.             Add  $C[j].nodes$  to  $C[i].nodes$
10.             $C[i].count += C[j].count$
11.            delete cluster  $j$  from  $C$
12.         **end if**
13.     **end for**
14. **end for**
15. **repeat** steps 1-15 until the list  $C$  does not change any more  
         $/**$  last step... update the variance for all remaining clusters  $*/$
16. **for** each remaining cluster  $i$  in  $C$   
        
$$C[i].error+ = \frac{1}{C[i].count} \sum_j C[i].nodes[j].value * ...$$
17.             
$$... * \sqrt{(C[i].mu_u.x - C[i].nodes[j].x)^2 + (C[i].mu_u.y - C[i].nodes[j].y)^2}$$
18. **end for**

### **B.3. DIRECTIVITY MODELS**

After sampling extensively in the vicinity of a sound source, we can create a model of the sources directivity. The directivity is an estimate of volume vs. angle for a sound source at a known, or pre-determined (possibly using auditory evidence grids)

location. The output of this algorithm is a maximum volume, and a directivity estimate describing each angle in terms of percentage of maximum volume emitted. This work was covered theoretically in Section 4.2.2.

### Variables

- **Source Location, [Sx,Sy]**

This is the location of the sound source, as indicated by either a priori information, or auditory evidence grids.

- **SessionTime**

the time at which the source was investigated, as indexed in the database

- **Set of Samples, *samples***

The set of samples is an array collected from the database. It contains as sub-variables the following information:

*SPL* – sound pressure level from each sample, for a particular microphone

*dist* – distance of the sample from the source centroid

*angle* – angle from the source to the sample location

The set is retrieved directly from the database, using the following query(variables are indicated by quotation marks):

```
SELECT  A.SPL,  SQRT((B.XLoc-("S.x"))*(B.XLoc-(S.x))  +
          (B.YLoc-("S.y"))*(B.YLoc-("S.y")),  ATN2(  B.YLoc-
          ("S.y"), B.XLoc-("S.x"))
FROM SPL A,MicArraySample B
WHERE A.Over_Threshold=0
```

AND A.SampleID=B.SampleID AND Mic\_Number=2

AND B.SessionTime in ("SessionTime")

AND  $\text{SQRT}((\text{B.XLoc}-("S.x"))^2+(\text{B.YLoc}-("S.y"))^2) < 2.0$

- **ReverbVolume**

This value is also retrieved directly from the database, only it is the average of all samples in this SessionTime located more than 2.0-m from the source centroid.

- **Directivity,  $Q$**

$Q$  is an array of real numbers, length 360 to be returned as an output of this function. When completed, each element  $i$  of the array corresponds to the percentage of the maximum volume of the sound source at degree  $i$ .

- **Source Volume,  $sVolume$**

This corresponds to the maximum volume of the source and is determined by the directivity algorithm at the end.

### Pseudocode

/\*\* The numerator and denominator storage variables are necessary for gaussian smoothing. In them, we store the intermediate results for each angle we are estimating directivity \*/

1. numerator = zeros[360];
2. denominator = zeros[360];
3. **for** k=1:# of samples

```

    /*** determine the volume of the sample at 1-m from the source, setting to zero if
    less than the estimated reverberant field volume ***/
4.  if ReverbVolume < samples[k].SPL
5.       $vol = (10^{SPL/10} - 10^{ReverbVolume/10}) * sample\{k\}.dist$ 
6.  else
7.       $vol = 0;$ 
8.  end if
9.  for i=1:360
10.      $dA = (samples[k].angle - i * \pi / 180);$ 
11.     convert dA to between  $[-\pi, \pi]$ 
12.      $tmp = e^{-1 * (dA)^2 / 2\sigma^2};$     /***  $\sigma = 0.5$  rad ***/
13.      $denominator[i] += tmp;$ 
14.      $numerator[i] += tmp * vol;$ 
15.  end for
16. end for
17.  $Q = 10 \log_{10}(numerator ./ denominator)$  /** where “./” indicates element-wise
    division of the arrays **/
18.  $sVolume =$  maximum of  $Q$ 
19.  $Q = \frac{Q}{sVolume};$     /*** convert Q to a percentage ***/

```

#### **B.4. MEL FREQUENCY CEPSTRAL COEFFICIENTS**

The implementation of mel-frequency cepstral coefficients used in this dissertation is derived from Malcolm Slaney’s Auditory Toolbox [Slaney 1994]. Our implementation is simplified in terms of some of the options available to the user, especially in terms of the number of coefficients it calculates. For the stated assumptions, however, the two implementations generate the same numeric results.

##### **Variables**

- **signal**

The 2048 sample signal recorded by one of the microphones in the array.

Note that the MFCC results, unlike Sound Pressure Level, were not

calculated for all microphones, because of difficulties in classification across different microphones.

- ***framesize, 10-msec***

The size of the frame over which each mfcc vector is calculated.

- ***fft\_size, 512 bytes***

The size of the fft window used with each frame,

- **SampleRate, 8192 Hz**

- **WindowSize, 256 bytes**

- **Output FeatureVector**

This is the 16-element MFCC feature vector describing the signal.

- **filterBank[i]**

The filterBank variable describes the range and height of the filters to be used in determining each coefficient. The following sub-variables will be determined in the code:

*lower* – indicates the lower edge of the triangle included in coefficient *i*

*center* – indicates the center of the triangle included in coefficient *i*

*upper* – indicates the upper edge of the triangle included in coefficient *i*

*height* – indicates the height of the triangle filter used for coefficient *i*

- **frameLength**

the length of the frame in bytes, calculated  
 $\text{round}(\text{SampleRate}/(\text{frameSize}/1000));$

- **frameCount**

the number of frames in a single sample, calculated as  
 $\text{floor}(\text{size}(\text{signal})/\text{frameLength})$

- **hamm**

a Hamming window of length *frameSize*

### Pseudocode

```
1. results = array[frameCount][8];    /*** store the first 8 MFCC's for each frame ***/
2. window = zeros[fft_size];
/*** need to build the mel filter bank***/
3. for j=1:8
    /*** for the first 13 coefficients, the mel-filter bank is actually linear spacing ***/
4.     if j==1
5.         filterBank[1].lower = 133.3333
6.     else
7.         filterBank[j].lower = filterBank[j-1].lower + 66.66666666
8.     end if
9. end for
10. filterBank[8].center = filterBank[8].lower + 66.66666666;
11. filterBank[8].upper = filterBank[8].center + 66.66666666;
12. filterBank[8].height = 2/(filterBank[8].upper-filterBank[8].lower);
13. for j=7:-1:1
14.     filterBank[j].center = filterBank[j+1].lower;
15.     filterBank[j].upper = filterBank[j+1].center;
16.     filterBank[j].height = 2/(filterBank[j].upper-filterBank[j].lower);
17. end for
/*** now go through each frame, identifying the first 8 mfcc values and storing them in
results ***/
18. for i=1:frameCount
19.     start = 1+(frameCount-1)*frameSize;
20.     window[1:256] = signal[start:(start+frameSize)].*hamm; /*** bitwise multiply ***/
21.     data = magnitude(FFT(window));    /*** take the absolute value of the FFT ***/
/*** now for each coefficient, apply a triangle filter based on the filter bank
calculated earlier to the power spectrum data ***/
```

```

22.  for j=1:8
23.      results[i][j] = 0;
24.      a = index of FFT window corresponding to filterBank[j].lower
25.      b = index of FFT window corresponding to filterBank[j].center
26.      c = index of FFT window corresponding to filterBank[j].upper
27.      if b>fftSize
28.          b = fftSize
29.      end if
30.      if c>fftSize
31.          c = fftSize
32.      end if
      *** first add the data from the rising edge ***/
      
$$slope_{rising} = \frac{filterBank[j].height}{(filterBank[j].center - filter[j].lower)}$$

33.
34.      for k=a:b
35.          freq = frequency of kth element in FFT
36.          if freq>=filterBank[j].lower
37.              weight = sloperising * (freq - filterBank[j].lower)
38.              results[i][j] += weight*data[k];
39.          end if
40.      end for
      *** now add the data from the falling edge ***/
      
$$slope_{falling} = \frac{filterBank[j].height}{(filterBank[j].upper - filter[j].center)}$$

42.
43.      for k=b+1:c
44.          freq = frequency of kth element in FFT
45.          if freq>=filterBank[j].center
46.              weight = slopefalling * (filterBank[j].upper - freq)
47.              results[i][j] += weight*data[k];
48.          end if
49.      end for
      ***take the log of the result to build the coefficient ***/
50.      results[i][j] = log10(results[i][j])
51.  end for
52. end for
*** final step, determine the mean and variance across all frames to produce the feature vector for coefficients 2-8 ***/
53. for j=2:8
54.     FeatureVector[j-1] = mean of results[i][j], across all frames i
55.     FeatureVector[j+6] = variance of results[i][j], across all frames i
56. end for

```

## B.5. CREATING DIRECT FIELD MAPS

The theory behind the creation of direct field maps is described in Chapter 3 as part of the sound fields framework. Their use with robot collected data about sound sources is then described in Chapter 4. All of the applications described in this thesis make use of these direct field maps. The only difference between the applications in the implementation is the use of sound source directivity. When the sound source directivity is not known, the algorithm automatically assumes an omni-directional source (i.e.  $Q=1$  for all angles).

### Variables

- **Direct Field Map Output,  $Map$  –**

A noise map of size  $N \times M$ , initialized to zero at all locations

- **Map gridcell size,  $stepsize$**

All gridcells were assumed to be square. Most maps were of 0.3-m to each side.

- **Map Range,  $[MinX-MaxX, MinY-MaxY]$**

The range of area described by the auditory evidence grid.

- **Active Sound Source List,  $S$**

$S$  contains all needed information about active sound sources in the environment. This information included the following sub-variables:

Directivity,  $dir$  – 360 element array

Position,  $[x, y, \theta]$

Maximum volume,  $SPL$

If the database was up to date, then the set of active sources could be obtained with a very simple SQL query:

```
SELECT Xloc,Yloc,Theta,Directivity>Total_SPL
FROM Source
WHERE IsActive = 1
```

### Pseudocode

```
1.  $k = 1$ 
2. while  $k \leq \#$  of elements in S
3.   for each cell [i,j] in Map
      /** First... find the distance and angle from the source to the cell***/
4.    $[Px,Py]$  = real coordinates of the center of grid cell [i,j]
5.    $dist = \sqrt{(Px - S[k].x)^2 + (Py - S[k].y)^2}$ 
6.    $theta = \arctan\left(\frac{Py - S[k].y}{Px - S[k].x}\right)$ 
7.    $deg = \text{round}\left(\left(theta - S[k].\theta\right) * \frac{180}{\pi}\right)$  /*convert to degrees*/
/** Now calculate the effect on each cell, assuming a minimum distance of one gridcell from the centroid***/
8.   if  $dist < \text{stepsize}$ 
9.      $V_{effect} = dir[deg] * SPL - 20 \log_{10}(\text{stepsize})$ 
10.  else
11.     $V_{effect} = dir[deg] * SPL - 20 \log_{10}(dist)$ 
12.  end else
      /** Add the result to the running total for each cell ***/
13.   $Map[i][j] = 10 \log_{10}(10^{V_{effect}/10} + 10^{Map[i][j]/10});$ 
14.  end for
15.   $k = k + 1;$ 
16. end while
```

### **B.6. RAY-TRACING FOR DIRECT AND/OR REVERBERANT FIELD MAPS**

The ray-tracing implementation used in this dissertation is designed after [Elorza 2005]. Our implementation, however, makes some simplifying assumptions to work with a coarse-grained evidence grid representation of obstacles in the environment. In

particular, rays only propagate along a plane, since our map is only 2-dimensional. Furthermore, it is assumed that all surfaces are flat and of only two alignments, 0-degrees or 90-degrees to the x-axis. For further discussion of the limitations of this approach, see Chapter 4.

### **Variables**

- **Field Map Output, *Map* –**

A noise map of size  $N \times M$ , initialized to zero at all locations

- **Map gridcell size, *stepsize***

All gridcells were assumed to be square. Most maps were of 0.3-m to each side.

- **Map Range, [*MinX-MaxX,MinY-MaxY*]**

The range of area described by the auditory evidence grid.

- **Active Sound Source List, *S***

*S* contains all needed information about active sound sources in the environment. This information included the following sub-variables:

Directivity, *dir* – 360 element array

Position, [*x,y,θ*]

Maximum volume, *SPL*

- **Ray**

This variable describes a single ray being traced through the environment.

The following sub-variables are used to describe this ray at different points along its path:

*sAngle* – starting angle of the ray

*sPower* – starting power of the ray

*total\_dist* – distance traveled by the ray

*bin\_dist* – distance traveled by the ray through the cell

*refl\_count* – the number of times the ray has been reflected

$[x,y,\theta]$  – the last position and angle of the ray

- **Copies of Ray at Different Locations, *bins***

*bins*[i][j] is a list, storing copies of each ray that crossed grid cell [i,j]. If the ray crossed the same grid cell more than once, then it is listed multiple times in the *bins*[i][j] list.

- **Obstacle Map, OBS**

OBS is a boolean obstacle map created by applying a threshold to an evidence grid. Given an evidence grid that indicates the likelihood of containing an obstacle, a threshold of  $-1$  would mean that all locations in the grid with value higher than  $-1$  contain an obstacle, and all locations with value less than  $-1$  do not contain an obstacle. After applying the threshold, *OBS*[x][y] indicates whether or not an obstacle is located at arbitrary position (x,y).

### Pseudocode

1.  $k = 1$
2. **while**  $k \leq \#$  of elements in S
3.     **for**  $m = 1:3600$                              `/** 3600 Rays per source */`  
       `/** initialize the ray with an origin at the source, and a random angle */`
4.         `ray.refl_count = 0;`
5.         `ray.x = S[k].x;`
6.         `ray.y = S[k].y;`
7.         `ray.θ = select random direction;`
8.         `ray.sAngle = ray.θ;`

```

9.      ray.total_dist = -1;  /** the ray should technically emanate at 1-m from
                                the centroid, since that is where the SPL is
                                calculated ***/
10.     deg = round((ray.θ-S[k].θ)*180/π)  /*convert to degrees*/
/** convert SPL to power (in pico-Watts), assuming a surface area of a 1-
m cylinder, sampled at 1-m from the centroid. See [Raichel 2000] for
more details... finally, divide ray power by # of rays to represent an even
power distribution **/
11.     ray.sPower = 10dir[deg]*SPL/10+0.78
12.     ray.sPower = ray.sPower/3600;
/** identify the current cell, and add the ray to it's list ***/
13.     [i,j] = grid cell containing source centroid [S[k].x,S[k].y]

/** Now follow the ray as it travels through each cell in the map, saving
it to the appropriate lists as it travels, and changing angles when it hits an
obstacle ... the loop stops when the ray leaves the map, or the total
distance traveled is greater than 20-m (i.e. no power left), or the number of
reflections is to high (i.e. power is lost through surface affects) ***/

14.     do      /** loop ***/
15.             ray.BinDist = distance traveled across cell [i,j]
16.             bins[i][j].add(ray)
17.             [a,b] = next cell in map, based on ray's current trajectory
18.             ray.[x,y,θ]=where ray exits cell [i,j]

/** check for reflections at the boundary of the next cell. To
allow for sound sources that are mounted on top of an obstacle, we
will ignore reflections that happen within 1-m of the source. This
will cause problems for sound sources within 1-m of a wall. ***/

19.             if (ray.TotalDist>1.0-m) and (OBS[a][b]=false)
20.                 if reflecting surface is horizontal
21.                     ray.θ=π- ray.θ,
22.                 else reflecting surface is vertical
23.                     ray.θ=-1* ray.θ,
24.                 end else
25.                 [i,j] = [a,b] /** ray is reflected back into same cell ***/
26.             end if
27.             ray.TotalDist = ray.TotalDist + ray.BinDist;
28.     until (i<1 or i>N or j<1 or j>M) or (ray.TotalDist>20-m) or
(ray.refl_count>20)
29.     end for
30. end while

```

/\*\* now that we have all of the rays recorded, go through each list and build whatever field needs to be built \*\*\*/

```

31. for each cell [i,j]
32.   for each ray, r, in bins[i][j]
33.      $onset\_power_r = bins[i][j][r].sPower * e^{-bins[i][j][r].TotalDist}$ 
34.      $Intensity_r = onset\_power_r * \frac{bins[i][j][r].BinDist}{cell\_volume}$ 
35.   end for
36.    $SPL = 120 + \log_{10}\left(\sum_r Intensity_r\right)$    /** convert intensity to sound pressure
                                                    level ***/
   /** this equation built the combined direct + reverberant field. We could also
   build just the direct field by only including rays with no reflections. Or, we could
   build just the reverberant field by only including rays with one or more
   reflections. ***/
37.    $Map[i][j] = 10 * \log_{10}\left(10^{SPL/10} + 10^{Map[i][j]/10}\right)$ 
38. end for

```

### B.6.1. CREATING AN INTENSITY PROFILE

The intensity profile answers the question, for a given location in the environment, what direction is the sound energy coming from. The resulting profile estimates energy vs. angle, and can be determined directly from the ray-tracing results described above. This intensity profile is used in Section 6.3.1 to estimate environmental impact on a target listener.

#### Variables

- **Copies of Ray at Different Locations, *bins***

*bins*[i][j] is a list, storing copies of each ray that crossed grid cell [i,j]. If the ray crossed the same grid cell more than once, then it is listed multiple times in the *bins*[i][j] list.

- **Target Location, *T***

the [x,y] position of the target the intensity profile is being created for.

- **Output Intensity Profile,  $I$**

A 360 element array describing the resulting intensity profile for the specified location, due to a particular source

**Pseudocode**

1. numerator = zeros[360];
2. denominator = zeros[360];
3. [i,j] = grid cell in which target  $T.[x,y]$  is located
4. **for** each ray,  $r$ , in  $bins[i][j]$
5.  $onset\_power_r = bins[i][j][r].sPower * e^{-bins[i][j][r].TotalDist}$
6.  $Intensity_r = onset\_power_r * \frac{bins[i][j][r].BinDist}{cell\_volume}$
7. **for** i=1:360
8.  $dA = (samples[k].angle - i * \pi / 180)$ ;
9. convert  $dA$  to between  $[-\pi, \pi]$
10.  $tmp = e^{-1 * (dA)^2 / 2\sigma^2}$ ; */\*\*  $\sigma=0.4$  rad \*/*
11.  $denominator[i] += tmp$ ;
12.  $numerator[i] += tmp * Intensity_r$ ;
13. **end for**
14. **end for**
15.  $I = numerator ./ denominator$ ; */\*\* where “./” indicates element-wise division of the arrays \*/*

**B.6.2. MODELING THE EFFECTS OF A MOVING ROBOT**

In Section 6.3.1, we discussed the use of a reversed form of ray-tracing to estimate the effects on a target by a robot located at any number of locations throughout the environment. Described in the following pseudocode, this implementation estimates the maximum difference between the intensity profile due to environmental sources, and the intensity profile due to the robot being located at any position in the room.

**Variables**

- **Active Sound Source List,  $S$**

- **Target Location,  $T$**
- **Environmental Intensity Profile,  $Amb$** 

The intensity profile at the target's location due to known active sound sources in the environment (see Appendix B.6.1)
- **Average Reverberation Level,  $R$** 

The average reverberation level at target location  $T$  due to ambient noise sources.
- **Volume of the robot,  $rVol$** 

Average volume of the robot at a distance of 1-m.
- **Output Impact Map,  $iMap$** 

The resulting impact map measuring the maximum difference in angular energy between the intensity profile due to the robot and the environmental intensity profile.

### Pseudocode

1. Use  $S$  to estimate  $Amb, R$  at location  $T$  (Appendix B.6.1).
2. Let  $robot\_source$  be a new sound source located at  $T$  with volume  $rVol$
3. Use ray-tracing to build the  $bins$  variable, due to the single source  $robot\_source$
4. **for** all rays  $[i,j,r]$  in  $bins$ , let  $bins[i][j][r].\theta = bins[i][j][r].sAngle$ 

/\*\* If only the volume difference between ambient noise sources is needed, then at this point, only regular ray-tracing (lines 32-37) is required to estimate total volume due to the robot. Otherwise, for estimating the difference in angular energy, continue with lines 5-9 \*/
5. **for** each location  $[i,j]$  in  $iMap$
6. Build an intensity profile  $Robot_{i,j}$  for location  $[i,j]$ 

/\*\* now estimate the detectability across all angles as the difference between the auditory scene at target location  $T$  with the robot, and without the robot \*/
7.  $detectability = 10 \log_{10}(Amb - Robot_{i,j} + R) - 10 \log_{10}(Amb + R)$ 

/\*\* the impact is the maximum angular detectability of the robot \*/
8.  $iMap[i][j] = \max(detectability);$
9. **end for**

## B.7. SAMPLED DATA NOISE MAPS

Maps of the auditory scene can also be constructed using sampled data directly, rather than relying on sound propagation models and derived information about sound sources. These sampled data maps rely upon interpolation to estimate the volume of ambient noise over a wide area. Shown below is a description of how to create these maps using cubic interpolation. As the interpolation function is taken directly from a mathematics library (GNU Scientific Library), it could easily be substituted for any number of other interpolation functions, including K-nearest neighbor, or linear interpolation. More details about using sampled data maps can be found in Section 4.4.

### Variables

- **Sampled Noise Map Output,  $Nmap$**  –

A noise map of size  $N \times M$ , initialized to zero at all locations

- **Set of Samples,  $samples$**

$samples$  is the set of samples to be included in the creation of this sampled data noise map. Usually, it is associated with a single sampling session, but can be connected to multiple session to increase the sampling area and/or number of samples used to create the noise map. The following sub-variables are associated with this variable:

$[x,y,\theta]$  - location of the robot when the sample was collected

$MicNumber$  – the id of the microphone that collected the sample

$SPL$  – the sound pressure level for this sample and microphone.

$OverThreshold$  – identifies whether or not the sample contained

errors from the DAQ and, therefore, should be discarded.

The sample can be pulled from the database using the following SQL statement:

```
SELECT    Sp.SampleID,    Sp.Mic_Number,    MAS.XLoc,
          MAS.YLoc, MAS.Theta, Sp.Over_Threshold, Sp.SPL
FROM SPL Sp, micArraySample MAS
WHERE Sp.SampleID=MAS.SampleID
AND MAS.SessionTime in “SessionTime”
```

- **Set of Samples with no calculated math, *samp\_noMath***

As was done with the creation of auditory evidence grids, the sound pressure level was not calculated for individual samples until it was needed. At that point, however, the result was stored to the database for future re-use. This variable identifies the set of samples that do not yet have SPL results stored in the database.

```
SELECT *
FROM micArraySample MAS
WHERE SampleID not in (
SELECT MAS.SampleID FROM SPL Sp, micArraySample MAS
WHERE Sp.SampleID=MAS.SampleID
AND MAS.SessionTime in “SessionTime”)
AND MAS.SessionTime in “SessionTime”
```

- **Mic Array Info, *mArray***
- **[*x,y,θ*] – offset from robot position**
- **Microphone Position, *mic\_pose<sub>i</sub>***

the [x,y,z] position of the  $i^{th}$  microphone in the array, relative to the array center.

- **Calibration constant, *calibrateRMS<sub>i</sub>***

CalibrateRMS identifies the rms pressure of a 40-dB sound source detected by microphone  $i$ . It is a calibration constant that allows us to compare samples recorded by the different microphones in the array. The value of the constant was determined by measuring a single source of known volume from a number of different positions with each of the microphones and then averaging the result.

**Pseudocode**

```

/** the first step is to calculate the sound pressure levels for all samples that do not yet
have an SPL value entered into the database */
1. for each sample  $k$  in samp_noMath
2.   for each microphone in the array
3.      $M = \text{magnitude}(\text{FFT}(\text{signal}))$ ;
4.     
$$\text{energy} = \frac{2}{\text{SampleSize}} \sum_{i=2}^{\text{SampleSize}/2} M$$

5.      $P_{rms} = \sqrt{\text{energy} / \text{SampleSize}}$ 
6.      $P_0 = \text{calibrateRMS} / 100$ ;
7.      $\text{SPL}_i = 20 * \log_{10}(P_{rms} / P_0)$ ;
8.     save the result to the database
9.   end for
10. end for
11. update samples
12. for each sample  $k$  in samples
    /** identify the global coordinate of the microphone that recorded the sample */
13.   
$$\text{rad} = \sqrt{\text{mic\_pose}_{\text{samples}[k].\text{MicNumber}}.x^2 + \text{mic\_pose}_{\text{samples}[k].\text{MicNumber}}.y^2}$$

14.   
$$\text{th} = \arctan(\text{mic\_pose}_{\text{samples}[k].\text{MicNumber}}.y / \text{mic\_pose}_{\text{samples}[k].\text{MicNumber}}.x)$$

15.   
$$x_{\text{global}} = \text{sample}[k].x + \text{rad} * \cos(\text{sample}[k].\theta + \text{th})$$

16.   
$$y_{\text{global}} = \text{sample}[k].y + \text{rad} * \sin(\text{sample}[k].\theta + \text{th})$$

    /** add the global coordinates and volumes to lists */

```

```
17. X.add(xglobal);
18. Y.add(yglobal);
19. Z.add(samples[k].SPL)
20. end for
21. Use cubic interpolation with data X,Y,Z to estimate the value of all cells in Nmap
/** this last step was a function taken from a public library, such as the GNU scientific
library, or Matlab. The resulting map sets all values of NMap outside the convex hull
created from {X,Y} as 0. */
```

## **Appendix C - GUIDING ROBOTIC MOVEMENT**

This second appendix provides more detailed descriptions and pseudocode for the algorithms guiding robotic movement in Chapter 4. The algorithms discussed in this Appendix are: (1) creating a map of clear space from an evidence grid representation of obstacles in the environment; (2) patrolling and environment by ordering a set of waypoints and then following the ordered path; and (3) investigating the environment by picking some set of waypoints from the clear-space map, and then dynamically choosing new targets to move towards depending upon the robots current position. Each of these three algorithms were implemented in the Task Manager software (see Appendix A for more detail) and required communication with Player to control robotic movement.

### **C.1. CLEAR-SPACE MAP**

A number of the robotic movement algorithms discussed in this dissertation require the identification of clear, reachable locations in the environment, including: (1) the investigative movement proposed in Chapter 4 for identifying location, volume, directivity, and sound function of a sound source; (2) the circular patrol algorithm proposed in Chapter 5 for surveying the auditory scene; (3) planning a path to avoid noise in Chapter 5; and (4) planning a path to hide in the noise in Chapter 6. This first section of the appendix on robotic movement provides a more detailed description of how such locations in the environment are identified. The result is a map of clear, reachable space that the robot can use to plan future movement.

#### **Variables**

- **Obstacle Map – *OBS***

OBS is a boolean obstacle map created by applying a threshold to an evidence grid. Given an evidence grid that indicates the likelihood of containing an obstacle, a threshold of  $-1$  would mean that all locations in the grid with value higher than  $-1$  contain an obstacle, and all locations with value less than  $-1$  do not contain an obstacle. After applying the threshold,  $OBS(x,y)$  indicates whether or not an obstacle is located at arbitrary position  $(x,y)$ .

- **Robot Initial Position – *pose***

The starting position of the robot is needed to identify at least one known region of clear space. For purposes of identifying clear space, the robot *pose* only needs to contain the  $(x,y)$  location of the robot.

- **Robot Radius – *robot\_rad***

In addition to needing a known region of clear space, we also need to know the minimum size of the region reachable by a robot. Ideally this would be no larger than the size, or radius, of the robot. In practice, however, this dissertation added 0.4-m to the radius of the robot to allow for errors in obstacle-avoidance that would prevent the robot from moving to close to obstacles.

- **Clear Space Map – *CLEAR\_MAP***

The clear space map is initially set to UNKNOWN for all locations. When the algorithm has concluded, the value of a given cell could be any one of the following: (REACHABLE) meaning that the cell is both clear and reachable by the robot, (CLEAR) meaning that the cell is clear and

adjacent to a reachable location, but too close to an obstacle for the robot to reach, (OCCUPIED) meaning that the cell contains an obstacle, (UNKNOWN/UNREACHABLE) meaning that this cell is not reachable by the robot due to the path being blocked.

- **Location Stack – *STACK***

The location stack contains locations that have been identified as being clear, but not yet checked for reachability. The location stack is initialized with the robots current pose (*robot\_pose*), and the algorithm terminates when the location stack is empty. The stack has two operations, push and pop. Push places the item at the top of the stack, above all other items in the stack. Pop returns the item from the top of the stack, removing that item from the stack.

**Pseudocode**

1. `STACK.push(pose.x,pose.y)`
2. **while** *STACK* is not empty
3.     `Location = STACK.pop()`
4. **if** (`OBS(Location.x,Location.y)` is UNKNOWN) **and** (for all (x,y) within *robot\_rad* of *Location*, `OBS[x][y]` is false)
5.     `CLEAR_MAP[Location.x][Location.y] = REACHABLE;`
6.     **for** each neighboring cell [i,j] of [*Location.x*, *Location.y*]
7.         **if** `OBS[i][j]` is true
8.             `CLEAR_MAP[i][j] = OCCUPIED`
9.         **else if** (`CLEAR_MAP[i][j]` is UNKNOWN)
10.             `CLEAR_MAP [i][j] = CLEAR`
11.             `STACK.push(i,j)`
12.         **end if**
13.     **end for** loop
14.     **end if**
15. **end while** loop

**C.2. PATROLLING THE ENVIRONMENT**

Patrolling the environment in this dissertation was implemented with a dynamically created finite state automaton (FSA). At run-time, the robot would select some set of waypoints through which it needed to pass, choose the shortest route through those points, and then build and follow an FSA to completion. An example FSA guiding the robot through a series of waypoints is seen in Figure C.1. More examples of how FSA's can be used to guide robotic navigation can be found in [Arkin 1998]. In this section of the appendix, we discuss in more detail the waypoint selection, path ordering and path following algorithms.

### C.2.1. SELECTING WAYPOINTS

In Section 5.3.2, we described an algorithm for selecting waypoints for a patrol robot, based on the levels of ambient noise in the room. The primary goal of the waypoint selection process is that there is a waypoint located within some minimum range of all reachable locations in the environments. The secondary goal is then to

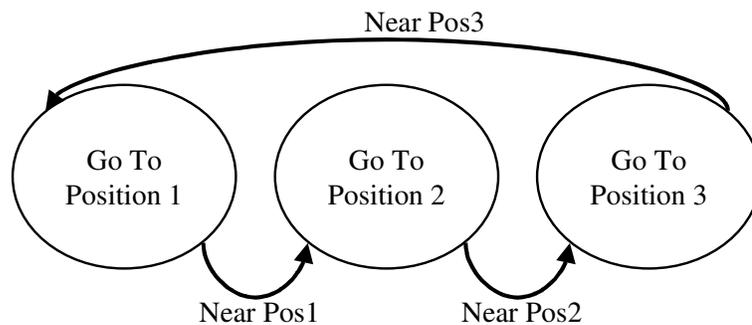


Figure C.1. A Finite State Automaton guiding a robot through a series of three arbitrary waypoints in the environment. This FSA actually guides the robot in a loop, however, as the robot will return to the first position after moving to the third position and start all over again.

minimize noise impact on a robot listening at each of those locations. The following pseudocode describes how we met both of these goals in the dissertation.

### **Variables**

- **Clear Space Map, *CLEAR\_MAP***

The clear space map identifies regions of clear, reachable space in the environment. Appendix C.1 has more detail on creating such a map from an evidence grid representation of obstacles in the environment.

- **Range of reachable space, [*MinX-MaxX,MinY-MaxY*]**

The range of reachable space identifies a bounding box about all clear, reachable areas of the clear space map.

- **Noise Map, *nMap***

A map of the noise levels in the room. It does not matter how this map was created (e.g. direct field, ray-tracing, sampled data, etc.)

- **Maximum Grid Cell Size, *gSize***

This specifies the maximum allowable grid cell size (1.8-m). This value multiplied by the square root of 2 indicates the maximum allowable distance between a waypoint and a clear, reachable grid cell that might contain a sound source.

- **Output List of Waypoints, *w***

The resulting list of waypoints, in real coordinates, for the robot to move to and sample at.

### Pseudocode

1.  $step\_size_x = ceiling\left(\frac{MaxX - MinX}{gSize}\right)$
2.  $step\_size_y = ceiling\left(\frac{MaxY - MinY}{gSize}\right)$
3. **for** a = MinX:step\_size\_x:MaxX
4.     **for** b = MinY:step\_size\_y:MaxY
5.         rangex = [a to a+step\_size\_x];
6.         rangey = [b to b+step\_size\_y];
7.         cell\_center = [a+step\_size\_x/2, b+step\_size\_y/2];
8.         let L be the set of all REACHABLE grid cells in CLEAR\_MAP within rangex and rangey
9.         Find the cell k in L with the lowest value in nMap... choosing the cell closest to cell\_center in the event of a tie.
10.         w.add([L[k].x,L[k].y]); /\*\* add real coordinates of cell center to list \*\*/
11.     **end for**
12. **end for**

If the provided noise map is empty or of uniform value, then the set of resulting waypoints defaults to the center of each range. Such a list is the same list as used in Section 5.2.2 for an uninformed waypoint selection process.

### **C.2.2. PATH ORDERING BY DISTANCE**

After obtaining a set of waypoints, the first step in building the FSA controller was to identify the order in which the waypoints should be visited. Assuming that all of the waypoints were reachable by the mobile robot, we used the Clear-Space Map described in the previous section to identify the all-pairs shortest path between waypoints. Given the small numbers of waypoints typically involved in this scenario, our chosen approach was to repeatedly apply Dijkstra's algorithm to all waypoints. For significantly larger numbers of waypoints, there are a number of alternative algorithms that may run faster[Cormen et al. 1990].

Using this graph representation, our heuristic for estimating the shortest path is to pick an arbitrary ordering of the nodes, and greedily swap nodes in the path which minimize the distance traveled by the robot.

### Variables

- **The graph representation  $G$**

$G(a,b)$  is the path length between waypoints  $a$  and  $b$ . It was calculated by repeatedly applying Dijkstra's algorithm for determining the single-source shortest path to each waypoint in the waypoint list. The map used in determining shortest path was a clear space map derived from an evidence grid representation of obstacles in the environment.

- **An initial ordering  $I_0$  of  $n$  waypoints  $w_n$**

- **An alternative ordering  $I_1(a,b)$**

$I_1(a,b)$  is identical to  $I_0$ , except that the positions of nodes  $a$  and  $b$  are swapped.

- **Path length difference  $\Delta(a,b)$**

Represents the total difference in path length between  $I_0$  and  $I_1(a,b)$

### Pseudocode

1.  $I_0 = [w_1, w_2, \dots, w_n]$
2.  $a = b = 1$ ;
3. **Repeat**
4.      $I_0 = I_1(a,b)$
5.     Find  $a,b$  such that  $\Delta(a,b)$  is maximized.
6. **Until**  $\Delta(a,b) < 0.3\text{-m}$

In the worst-case scenario, this algorithm could be less efficient than simply searching every possible ordering of nodes. However, in our case, the automatically generated waypoints were usually close to being ordered already, as the waypoints were

estimated in straight lines along the clear-space map. Using a greedy-node swap on a mostly ordered path meant that significant improvements in path length could be quickly achieved in only a few swaps.

### C.2.3. PATH ORDERING BY NOISE LEVELS

This section details how to order a set of waypoints so as to minimize the ambient noise exposure of a robot. The algorithm uses essentially the same pseudocode as in Appendix C.2.2, but substitutes the sum of the noise between nodes for the distance traveled. The following pseudocode describes how to build the graph representation  $G$ , so that the cost of traveling between waypoints reflects ambient noise levels instead of distance. It still uses Dijkstra's algorithm [Cormen et al. 1990] to calculate this cost, but incorporates the values of a noise map into the cost estimation process.

#### Variables

- **The set of waypoints,  $w$**
- **Clear space map,  $CLEAR\_MAP$**
- **Noise map,  $nMap$**
- **Intermediate Path Map,  $pMap$**

$pMap$  stores the intermediate results. For each cell  $[i,j]$  in  $pMap$ , the following information is stored:

*cost* – the cost of travelling from the path start to grid cell  $[i,j]$

*previous\_cell* – the previous cell  $[a,b]$  along the path to grid cell  $[i,j]$

- **Sorted List,  $Q$**

$Q$  is a list of cells  $[i,j]$ , ordered so that the head of the list is the cell in  $Q$  with the smallest value of  $pMap[i][j].cost$ . The pop operation removes the head of  $Q$ , leaving the cell with the next smallest cost at the head of the list.

- **The output graph representation  $G$**

$G(a,b)$  is the resulting cost of traveling between waypoints  $a$  and  $b$ .

**Pseudocode**

```

1. for each waypoint  $k$  in  $w$ 
    /*** initialize the path map ***/
2.   let  $[a,b]$  be the grid cell in  $CLEAR\_MAP$  that contains  $[w[k].x,w[k].y]$ 
3.   initialize  $pMap$  so that  $cost$  is infinity for every cell
4.    $pMap[a][b].cost = 0;$ 
5.    $pMap[a][b].previous\_cell = NULL;$ 
6.   add cell  $[a,b]$  to list  $Q$ 
7.   while  $Q$  is not empty
      /*** retrieve the node with shortest cost still in  $Q$  ***/
8.      $[a,b] = Q.head();$ 
9.      $Q.pop();$ 
10.    for each neighboring cell  $[i,j]$  of  $[a,b]$ 
    /*** add neighboring cells to  $Q$  when a new, less costly path has been identified ***/
11.      if  $CLEAR\_MAP[i][j]$  is REACHABLE
12.         $alt = pMap[a][b].cost + nMap[i][j]$ 
13.        if  $alt < pMap[i][j].cost$ 
14.           $pMap[i][j].cost = alt;$ 
15.           $pMap[i][j].previous\_cell = [a,b]$ 
16.        end if
17.      end if
18.    end for
19.  end while
    /***  $Q$  being empty means that the shortest path to all grid cells has been identified...
    now extract the cost to reach each of the waypoint cells from the starting waypoint ***/
21.  for each waypoint  $l$  not equal to  $k$ 
22.    let  $[i,j]$  be the grid cell in  $CLEAR\_MAP$  that contains  $[w[l].x,w[l].y]$ 
23.     $G(k,l) = pMap[i][j].cost$ 
24.  end for
25. end for

```

#### C.2.4. PATH FOLLOWING

After ordering the waypoints, following the path was simple. Starting with the first waypoint in a non-circular path (or the closest waypoint in a circular patrol route), the Task Manager software would send the target coordinates to the player interface, which in turned passed them to the wavefront planner in Player, causing the robot to move to that target. The Task Manager then constantly maintained track of the robot's position while it was moving. When the robot was within an acceptable distance (0.5-m) of the target, the next waypoint in the list was selected and the taskmanager would pass the new target to Player. The resulting controller is a finite state automaton, where the state of the controller is the waypoint being moved towards by the mobile robot.

Using this simple FSA for robotic control is described in the following pseudocode for a non-circular path:

##### Variables

- **Current Location of the Robot – (*Loc*)**

*Loc* was estimated by the amcl driver in Player.

- **The  $i^{\text{th}}$  waypoint in the ordered path -  $I_0(i)$**

##### Pseudocode

1.  $i=1$ ;
2. **while**  $i < n+1$
3.     Send Goal  $I_0(i)$  to Player
4.     Update *Loc*
5.     **while**  $\text{distance}(Loc - I_0(i)) < 0.5\text{-m}$
6.         Acquire Audio Sample
7.         Update *Loc*
8.     **end while** loop
9.      $i = i+1$ ;
10. **end while** loop

### C.3. INVESTIGATION OF A SOUND SOURCE

After the location of a potential sound source had been identified, the next stage of the sound source discovery process (as described in Chapter 4) was to collect a large number of samples in the vicinity of the source, preferably from as many angles as possible. This was accomplished using the Clear-Space map (Appendix B.1) to identify a number of waypoints in the vicinity of the sound source, and then moving the robot to each of those waypoints and collecting a sample. The remainder of this section discusses the implementation details of these two parts of the algorithm: (1) identifying waypoints in the vicinity of the source, and (2) moving to and sampling at each of the waypoints.

#### C.3.1. IDENTIFYING WAYPOINTS IN THE SOURCE VICINITY

##### Variables

- **Clear Space Map, *CLEAR\_MAP***
- **Map gridcell size, *stepsize***

All gridcells were assumed to be square. Most maps were of 0.3-m to each side.

- **Map Range, [*MinX-MaxX,MinY-MaxY*]**

The range of area described by the auditory evidence grid.

- **Suspected Source Location, [*Sx, Sy*]**

This is the grid cell containing the location to be investigated by the robot. Typically, the location's global coordinates are determined by applying an auditory evidence grid to patrol data.

- **Sampling radius, *radius***

The investigation needs to sample at locations up to 2-m away from the source centroid. As the robot's movement trajectory is not precise, the result still contains a number of samples collected outside the 2-m radius which can be used for estimating the reverberant volume of the room.

- **Clear Map gridcell size, *stepsize***

All gridcells were assumed to be square. Most maps were of 0.3-m to each side.

- **Output - Sampling Locations, *w***

The resulting locations at which the robot should sample to investigate the sound source.

### **Pseudocode**

1.  $rad = radius/stepsize;$
2. **for**  $i=(a-rad):(a+rad)$
3.     **for**  $j=(b-rad):(b+rad)$
4.         **if** (cell  $[i,j]$  is in the map) **and** (CLEAR\_MAP $[i][j]$  is REACHABLE)
5.              $[x,y] =$  global coordinates of the center of cell  $[i,j]$
6.              $w.add([x,y])$
7.         **end if**
8.     **end for**
9. **end for**

In our implementation, Clear-Space maps were always of 0.3-m resolution, meaning that each cell in the grid was 0.3-m x 0.3-m in size. Therefore, this choice of waypoints typically resulted in 100-250 sampling locations, varying with the obstacle density in the vicinity of the target. The time it would take for our robot to sample at each of these locations was generally between 20-40 minutes. With a different map resolution, these numbers would have changed because the resolution is exponentially related to the number of waypoints. Therefore, given these numbers, which may already

be too large for many practical applications, a different algorithm may be necessary for extracting target locations if the map resolution is much smaller.

### C.3.2. DYNAMIC PATH PLANNING

The resulting waypoint distribution from this strategy is very widely but densely distributed by distance and angle about the suspected sound source location. Given this scattering of sampling targets we opted for a less informed approach to the path-planning problem. If robotic localization had been completely accurate, then a path-planner would have been the most appropriate choice, as a pre-ordered path could minimize the sampling time required to complete the investigation. Given, however, the density of the sampling targets and the size of the error in localizing the robot, often as large as 0.5-m, a pre-ordered path could actually be detrimental to performance as the robot overshoot points in the path and then tried to return to them. Therefore, instead we opted for a dynamic planner that picked the next waypoint in the path based on the robots current position and angle. After the robot successfully sampled at a given target, the next waypoint selected should be close to the robot, and ideally, straight ahead so that the robot moves in lines across the sampling space.

#### Variables

- **Set of remaining sampling locations -  $w$**

When initialized,  $w$  stores the entire set of waypoints  $(x,y)$ . As each waypoint is moved to and sampled at, however, it is removed from the list. Therefore, the list decreases in size as time progresses, emptying completely by the time the investigation finishes.

- **Distance to each waypoint -  $Dist_w(i)$**   
This variable (or function) returns the distance from the robot's estimated current location to the  $i^{\text{th}}$  waypoint in  $w$
- **Angular difference to each waypoint -  $Ang_w(i)$**   
This variable (or function) returns the difference in angle between the robot's current orientation and the vector from the robot's current position to the  $i^{\text{th}}$  waypoint in  $w$
- **Turning radius of the robot –  $turn\_rad$**   
Assuming that the robot must first rotate in place to reach the target (an assumption that is generally true for short distances), the turning radius (0.3-m) allowed us to account for extra distance the robot needs to move in order to reach waypoints that are off to one side or behind it.

### Pseudocode

1. **while** length(W)>1
2.     Update  $Dist_w, Ang_w$
3.     Find  $i$ , such that  $(Dist_w(i) + Ang_w(i)*turn\_rad)$  is minimized
4.     Send goal  $w(i)$  to Player
5.     **while**  $Dist_w(i)<0.5\text{-m}$      /\*loop until the robot reaches the target\*/
6.         Update  $Dist_w(i)$
7.     end while loop
8.     Stop Robotic Movement
9.     Acquire Audio Sample     /\*investigations used pause and sample methodology\*/
10.     Remove  $i^{\text{th}}$  waypoint from list  $w$
11. **end while** loop

## Appendix D - HRI APPLICATION

In Chapter 7, a Human-Robot Interaction application was presented which combined both reactive and deliberative elements of acoustical awareness together into a single implementation. This appendix provides the implementation details for this application. In the immediately following code, the Finite-State-Machine is implemented with a series of case statements. Then in the following sections, the acoustically-aware actions are described, including: (1) selecting speech volume, (2) pausing for interruptions, (3) rotating to face the listener, and (4) relocating the robot. Of these actions, the first three actions are designed to handle short duration disturbances to the auditory scene, while the final action makes use of the sound fields framework to respond to medium-to-long duration disturbances.

### Variables

- **Volume Threshold, *threshold***

This is the sound pressure level threshold at which ambient noise is considered too loud to speak over. This value actually depends on the type of noise the robot is working with, as some types of noise are easier for a human listener to ignore than others.

### Pseudocode

1. **do** `/** loop **/`
2. **switch** (state):
3. **case** WAIT\_STATE:  
`/** waiting for an interaction to happen, just sample the auditory scene ***/`
4. sample auditory scene
5. determine volume of the sample
6. **if** speech is detected
7.     Rotate the robot to face the speaker `/** Appendix D.3 **/`
8.     transition to LISTEN\_STATE

```

9.   else if volume remains > threshold for more than 10-seconds
10.       Relocate the Robot /** Appendix D.4 **/
11.   end if
12.   case LISTEN_STATE:
13.       sample the auditory scene
14.       determine the volume of the sample /** Appendix B.7 **/
15.       if speech is detected and the speech corresponds to a report title
16.           Rotate the robot to face the speaker /** Appendix D.3 **/
17.           transition to READ_STATE
18.       else if volume remains > threshold for more than 10-seconds
19.           Relocate the Robot /** Appendix D.4 **/
20.       end if
21.   case READ_STATE:
    /** read the report sentence by sentence, sampling in between to determine if the
    robot has been interrupted, and set the speech output volume **/
22.       for each sentence in the report
23.           sample the auditory scene
24.           Check for Interruptions /** Appendix D.2 **/
25.           Set the volume of the output /** Appendix D.1 **/
26.           Read the sentence
27.           save the current location in the report
28.       end for
29.   case WAIT_FOR_COMMAND_STATE:
    /** the robot was interrupted by speech, or the user decided not to move after a
    loud sound source **/
30.       Wait for a command from the speech detection engine.
31.       switch command:
32.           case 'continue where you stopped':
33.               transition to READ_STATE
34.           case 'repeat last line'
35.               move the current report location back one sentence
36.               transition to READ_STATE
37.           case 'repeat from the beginning'
38.               move the current report location back to the beginning
39.               transition to READ_STATE
40.           case 'Change to a new subject'
41.               transition to LISTEN_STATE
42.       end switch
43.       if no command occurs within 2-minutes
    /** assume interaction has ended **/
44.           transition to WAIT_STATE
45.       end if
46.   end switch
47. until application is turned off

```

## D.1. SELECTING SPEECH VOLUME

The volume of the robot is set every time the robot speaks. It is determined as follows:

### Variables

- **Noise Range, [ $MinN - MaxN$ ]**

This is the range of noise (in dB) during which the robot adjusts its speaking volume. If the ambient noise volume is less than  $MinV$  then the robot speaks at its softest. If the ambient noise volume is greater than  $MaxV$  then the robot speaks at its loudest.

- **Distance Range, [ $MinD - MaxD$ ]**

This is the range of distances over which a robot can communicate with a human partner. If the distance to the person is less than  $MinD$  then the robot speaks at its softest. If the distance to the person is greater than  $MaxD$  then the robot speaks at its loudest.

- **Output Volume Range, [ $MinV - MaxV$ ]**

This is the range of volumes over which the robot can talk. The SAPI 5.1 software had a range of 0-1. Our volume adjustment code needed to choose a sub-range of 0-1 over which the robot could react to ambient noise volume and distance. Our  $MinV$  was 0.4, so that the robot speech was still audible. Our  $MaxV$  was 1.0.

- **Current Volume,  $currentV$**

$currentV$  is the current ambient noise volume in dB, as detected by the microphone array mounted on the robot.

- **Current Distance, *currentD***

*currentD* is the current distance to the user, as determined by the stereo-vision system mounted on the robot.

Pseudocode

/\*\* Need to make sure that V and D are within the necessary range by checking minimums and maximums\*\*/

1.  $V = \text{currentV}$

2. **if**  $\text{currentV} < \text{MinN}$

3.      $V = \text{MinN}$

4. **else if**  $\text{currentV} > \text{MaxN}$

5.      $V = \text{MaxN}$

6. **end if**

7.  $D = \text{currentD}$

8. **if**  $\text{currentD} < \text{MinD}$

9.      $D = \text{MinD}$

10. **else if**  $\text{currentD} > \text{MaxD}$

11.      $D = \text{MaxD}$

12. **end if**

    /\*\* identify a scaling factor using an ellipse to combine the 2 dimensions \*\*/

13.     
$$SF = \left( \frac{V - \text{MinN}}{\text{MaxN} - \text{MinN}} \right)^2 + \left( \frac{D - \text{MinD}}{\text{MaxD} - \text{MinD}} \right)^2$$

    /\*\* generate an output volume based on a linear progression \*\*/

14.      $\text{output\_volume} = \text{ceil}(SF * (\text{MaxV} - \text{MinV}) + \text{MinV})$

## D.2. PAUSING FOR INTERRUPTIONS

When speech, or otherwise significant sound, was preventing communication, the robot would recognize this and pause its speech output. Depending upon the type of the sound disturbing the interaction, the length of the pause would vary.

### Variables

- **Volume Threshold, *threshold***

This is the sound pressure level threshold at which ambient noise is considered too loud to speak over. This value actually depends on the

type of noise the robot is working with, as some types of noise are easier for a human listener to ignore than others.

### **Pseudocode**

1. Between every sentence, sample the environment (250-msec/sample)
2. **if** speech was detected in that sample
3.     stop all interaction
4.     */\*\* wait until the user directs the robot to continue using voice commands \*\*/*
5.     **transition** to WAIT\_FOR\_COMMANDS\_STATE
6.     **else if** the ambient noise volume > *threshold*
7.         **repeat** */\*\* start of loop \*\*/*
8.             Check the last 3-sec of samples.
9.             **if** less than 10% of those samples were greater than *threshold*
10.                 **continue** speaking
11.             **end**
12.             pause for 1-sec.
13.         **until** 10-seconds have elapsed, or the robot started speaking again
14.         **if** the robot did **not** continue speaking
15.             Relocate the Robot (Appendix D.4)
16.         **end if**
17.     **end if**

### **D.3. ROTATING TO FACE THE LISTENER**

Besides changing its volume in response to ambient noise, and pausing when noise levels interrupt the conversation, the robot also needed to rotate to keep the user in view. This is accomplished through 2 sensory modalities: (1) audition, and (2) vision. The microphone array mounted on the robot is needed to localize the human participant before they are detected by the vision system. Then, once the human has been localized, the robot relies upon its vision system to return regular angular measurements as it tracks the human through the surrounding environment.

Unlike all other work on the human-robot interface, the rotation process is performed in parallel with the main controller. In other words, once speech initializes the

interaction, the rotation of the robot is performed at regular intervals regardless of the state of the interaction.

### Variables

- **Recent sample, *sample***

This is the most recent sample collected by the microphone array which has been determined to contain speech.

### Pseudocode

1. *SpLikelihood* = spatial likelihood of *sample* /\*\* Appendix B.1 \*\*/
2. *BestA* = most likely angle to the sound source in *SpLikelihood* /\*\* Appendix B.1.1 \*\*/
3. Rotate the robot to face *BestA*
4. **repeat**
5.     Get the angle  $\alpha$  to the human partner, using the vision system
6.      $dA = \alpha - \text{pose}.\theta$ ;
7.     Convert  $dA$  to between  $[-\pi, \pi]$
8.     **if**  $|dA| < \pi/6$
9.         Rotate camera to face human /\*\* maintains visual tracking \*\*/
10.    **else**
11.         Rotate the robot to face  $\alpha$
12.    **end else**
13. **until** interaction has ended

## **D.4. RELOCATING THE ROBOT**

Once the sound source has been identified as a medium-to-long duration noise source, the robot can try to relocate itself in the environment by localizing the sound source, building a direct field map of the environment, and moving to the quietest predicted location. This same work was used both in Chapter 7, and in Section 5.3.1 of Chapter 5.

### Variables

- **Robot position, *pose***

The last known location and orientation  $[x, y, \theta]$  of the robot.

- **Clear Space Map, *CLEAR\_MAP***

*CLEAR\_MAP* is a map of clear, reachable space. Appendix C.1 describes in more detail how to create such a map from an evidence grid representation of the obstacles in the environment.

- **Active Source List, *S***

This is the list of active sources in the environment. It is needed to build maps of the direct field, and is updated as new sound sources are discovered by the mobile robot.

### Pseudocode

1. *samples* = sample auditory scene for 20-seconds  
 /\*\*\* Determine the average sound pressure level, Appendix B.7 describes how to calculate the sound pressure level from a recorded sample \*\*\*/
2. *DetectedSource.vol* = average SPL at current location  
 /\*\*\* determine the direction to the sound source, Appendix B.1.1 describes how to estimate this value from spatial likelihood measurements \*\*\*/
3.  $Th_s$  = most likely direction to the sound source, averaged across *samples*
4.  $D_s = 1\text{-m}$  /\*\*\* assume 1-m away to start \*\*\*/
5. **if not** interacting with anyone  
     /\*\*\* with no current interactions, resort to algorithm described in Section 5.3.1 for relocating the robot \*\*\*/
6. Move robot in direction ( $Th_s + pose.\theta + 90^\circ$ ) for 10-sec  
     /\*\*\* Identify the position of the sound source by building an auditory evidence grid (Appendix B.2) and clustering the result (Appendix B.2.1) \*\*\*/
7. Build an auditory evidence grid *AEG* from collected samples
8. *DetectedSource.centroid* = largest cluster center in *AEG*
9. **else**  
     /\*\*\* The differences are : (1) the robot asks person if they want to move before moving, and (2) the robot does not try to localize the sound source first, instead assuming the source was located only 1-m away so as to speed up the relocalization process \*\*\*/
10. Ask the user if they want to relocate to another less noisy location
11. Wait for a response
12. **if** no response was returned within 2-min
13.      assume interaction has been canceled
14.      return to wait state
15. **else if** response is “no”

16. **exit** “Moving the Robot” state, and continue interaction at maximum volume
17. **else if** response is “yes”
18.     DetectedSource.centroid.x = pose.x+D<sub>s</sub>cos(Th<sub>s</sub> + pose.θ);
19.     DetectedSource.centroid.y = pose.y+D<sub>s</sub>sin(Th<sub>s</sub> + pose.θ);
20. **end if**
21. **end if**
22. Add DetectedSource to S
23. Build Direct Field map, *dMap*, of active sources */\*\* Appendix B.5 \*/*
24. Let [x,y] be the quietest location in *dMap* that is also REACHABLE in *CLEAR\_MAP*
25. Move the robot to [x,y].
26. **if** an interaction had been interrupted
27.     tell the user that the robot is ready to continue
28. **end if**
29. **exit** “Moving the Robot” state

## REFERENCES

- A. Krokstad, S. S., and S. Sorsdal (1968). "Calculating the Acoustical Room Response by the Use of a Ray-Tracing Technique." *Journal of Sound Vibration*, vol 8: 118-125.
- Alford, A., S. Northrup, K. Kawamura, and K-W. Chan (1999). "Music Playing Robot." *Proceedings of the International Conference on Field and Service Robotics (FSR '99)*: 174-178.
- Allen, J. B. and D. A. Berkely (1979). "Image method for efficiently simulating small-room acoustics." *Journal of the Acoustical Society of America*, vol 65: 943-950.
- Amsellem, A., O. Soldea, et al. (2006). "Function-Based Classification from 3D Data and Audio." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 336-341.
- Argentieri, S., P. Danes, et al. (2006). "Modal Analysis Based Beamforming for Nearfield or Farfield Speaker Localization in Robotics." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 866-871.
- Argentieri, S., P. Danes, et al. (2006). "Broadband Variations of the MUSIC High-Resolution Method for Sound Source Localization." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA: 866-871.
- Arkin, R. C. (1998). *Behavior-based Robotics*. Cambridge, MA, MIT Press.
- Atal, B. S. and L. R. Rabiner (1976). "A Pattern Recognition Approach to Voiced-Unvoiced Silence Classification with Applications to Speech Recognition." *IEEE Trans. Acoust., Speech, Signal Processing*, vol 24: 201-212.
- Balch, T., R. C. Arkin (1993). "Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation." *Proceedings of the Int. Conf. on Robotics and Automation (ICRA)*: 678--685.
- Bard, E. G. and M. P. Aylett (2000). "Accessibility, Duration, and Modeling the Listener in Spoken Dialogue." *Proceedings of the GötaLog 2000 Fourth Workshop on the Semantics and Pragmatics of Dialogue*, Göteborg, Sweden.
- Bian, X., G. Abowd, et al. (2005). "Using Sound Source Localization in a Home Environment." *Proceedings of the Third Int Conf on Pervasive Computing*, Munich, Germany, LNCS vol 3660: 19-36.

- Biber, P., H. Andreasson, et al. (2004). "3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera." *Proceedings of the Int Conf. on Intelligent Robots and Systems*, Sendai, Japan, vol 4: 3430-3435.
- Birgersson, E., A. Howard, et al. (2003). "Towards Stealthy Behaviors." *Proceedings of the Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV: 1703-1708.
- Birgersson, E., A. Howard, et al. (2003). "Towards Stealthy Behaviors." *Proceedings of the Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, vol 2: 1703-1708.
- Bischoff, R. (2000). "Towards the Development of 'Plug-and-Play' Personal Robots." *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Cambridge, MA.
- Blisard, S., B. Fransen, et al. (2007). "Using Vision, Acoustics, and Natural Language for Disambiguation." *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, Arlington, VA: 73-80.
- Bloothoof, G. and E. d. Os (1997). "'The Elsned Olympics: Testing Spoken Dialogue Systems at Eurospeech'97'." *ELSNEWS*, vol 6.5: 1-3.
- Borenstein, J. and Y. Koren (1989). "Real-time Obstacle Avoidance for Fast Mobile Robots." *IEEE Trans. on Systems, Man, and Cybernetics*, vol 19(5): 1179-1187.
- Boril, H., P. Fousek, et al. (2006). "Lombard Speech Recognition: A Comparative Study." *Proceedings of the 16th Czech-German Workshop on Speech Processing*, Prague, Czech Republic, vol 1: 141-148.
- BOSE. (2007). "AudioPilot Noise Compensation Technology." Retrieved Aug 25, 2007, from [http://www.bose.com/controller?event=VIEW\\_STATIC\\_PAGE\\_EVENT&url=/automotive/innovations/audiopilot.jsp&ck=0](http://www.bose.com/controller?event=VIEW_STATIC_PAGE_EVENT&url=/automotive/innovations/audiopilot.jsp&ck=0).
- Botteldooren, D. (1995). "Finite-Difference Time-Domain Simulation of Low-Frequency Room Acoustic Problems." *Journal of the Acoustical Society of America*, vol 98: 3302-3308.
- Bou-Ghazale, S. E. and J. H. L. Hansen (2000). "A comparative study of traditional and newly proposed features for recognition of speech under stress." *IEEE Trans. on Speech and Audio Processing*, vol 8(4): 429 - 442.
- Bradski, G., A. Kaehler, et al. (2005). "Learning-based computer vision with intel's open source computer vision library." *Intel Technology Journal*, vol 9(1).
- Breazeal, C. (2001). "Emotive Qualities in Robot Speech." *Proceedings of the Int. Conf. on Intelligent Robots and Systems (IROS)*, Maui, Hawaii, vol 3: 1388-1394.

- Bregman, A. S. (1990). *Auditory Scene Analysis: The Perceptual Organization of sound*. Cambridge, MA, MIT Press.
- Carrol, D., G. A. Gilbreath, et al. (2002). "Extending Mobile Security Robots to Force Protection Missions." *Proceedings of the AUVSI Unmanned Systems*, Lake Buena Vista, FL.
- Cavanaugh, W. (1999). "Introduction to Architectural Acoustics and Basic Principles." *Architectural Acoustics: Principles and Practice*. W. Cavanaugh and J. Wilkes. New York, John Wiley & Sons: 1-54.
- Chien, T., K. Su, et al. (2005). "The Multiple Interface Security Robot - WFSR-II." *Proceedings of the IEEE Int. Workshop on Safety, Security, and Rescue Robotics*, Kobe, Japan: 69-74.
- Clarkson, B., N. Sawhney, et al. (1998). "Auditory Context Awareness via Wearable Computing." *Proceedings of the Workshop on Perceptual User Interfaces*, San Francisco.
- Claudio, E. and R. Parisi (2001). "Multi-Source Localization Strategies." *Microphone Arrays*. eds. M. Brandstein, and D. Ward. Berlin, Germany, Springer: 181-201.
- Cormen, T., C. Leiserson, et al. (1990). *Introduction to Algorithms*. Cambridge, MA, MIT Press.
- Couvreur, C., V. Fontaine, and C.G. Mubikangiey (1998). "Automatic Classification of Environmental Noise Events by Hidden Markov Models." *Applied Acoustics*, vol 54(3): 187.
- Dellaert, F., F. Alegre, et al. (2003). "Intrinsic Localization and Mapping with 2 Applications: Diffusion Mapping and Marco Polo Localization." *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, TW, vol 2: 2344-2349.
- DiBiase, J, H. Silverman, and M. Brandstein (2001). "Robust Localization in Reverberant Rooms." *Microphone Arrays*. eds. M. Brandstein, and D. Ward. Berlin, Germany, Springer: 181-201.
- Dourish, P. and V. Bellotti (1992). "Awareness and coordination in shared workspaces." *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, Toronto, Canada: 107-114.
- Drager, K. and J. E. Reichle (2001). "Effects of Discourse Context on the Intelligibility of Synthesized Speech for Young Adult and Older Adult Listeners." *J. of Speech, Language, and Hearing Research*, vol 44: 1052-1057.

- Duda, R., P. Hart, et al. (2001). *Pattern Classification*. New York, NY, John Wiley & Sons.
- Dusan, S. and J. Flanagan (2002). "Adaptive Interface for Spoken Dialog." *Journal of the Acoustical Society of America*, vol 111(5): 2481.
- Egan, J. P. and H. W. Hake (1950). "On the Masking Pattern of a Single Auditory Stimulus." *Journal of the Acoustical Society of America*, vol 22: 622-630.
- Elfes, A. (1992). "Multi-source spatial data fusion using Bayesian reasoning." *Data Fusion in Robotics and Machine Intelligence*. M. A. Abidi and R. C. Gonzales. New York, Academic Press.
- Elorza, D. O. (2005). *Room Acoustics Modeling Using the Raytracing Method: Implementation and Evaluation*, Licentiate Thesis, Dept of Physics, University of Turku, Turku, Finland
- Endsley, M. (1988). "Design and evaluation for situation awareness enhancement." *Proceedings of the Human Factors Society 32nd Annual Meeting*, Santa Monica, CA: 97-101.
- Estrin, D., W. Michener, et al. (2003). "Environmental Cyberinfrastructure Needs for Distributed Sensor Networks: A Report from a National Science Foundation Sponsored Workshop." 12-14 August, 2003, Scripps Institute of Oceanography Retrieved Sept. 30, 2007, from [www.lternet.edu/sensor\\_report](http://www.lternet.edu/sensor_report).
- Femmam, S., N. K. M'Sirdi, et al. (2001). "Perception and characterization of materials using signal processing techniques." *IEEE Trans. on Instrumentation and Measurement*, vol 50(5): 1203-1211.
- Fong, T. W., I. Nourbakhsh, et al. (2003). "A survey of socially interactive robots." *Robotics and Autonomous Systems, Special issue on Socially Interactive Robots*, vol 42: 143-166.
- Funkhauser, T., N. Tsingos, et al. (2004). "A beam tracing method for interactive architectural acoustics." *Journal of the Acoustical Society of America*, vol 115(2): 739-756.
- Gat, E. (1991). "Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots." *SIGART*, vol Bulletin 2: 17-74.
- Gerkey, B., R. Vaughan, et al. (2003). "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems." *Proceedings of the 11th International Conference on Advanced Robotics*, Coimbra, Portugal: 317-323.

- Gerkey, B., R. Vaughan, et al. (February 2006). "Player User Manual v2.1." Player/Stage Project, <http://playerstage.sourceforge.net>, Retrieved September, 2005, from <http://playerstage.sourceforge.net>.
- Girod, L. and D. Estrin (2001). "Robust Range Estimation Using Acoustic And Multimodal Sensing." *Proceedings of the IEEE/RSI Int. Conf. on Intelligent Robots and Systems(IROS)*, Wailea, Hawaii: 1312-1320.
- Goldstein, E. B. (2007). *Sensation and Perception*. Belmont, CA, Thomson Wadsworth.
- Guo, Y., L. Parker, et al. (2004). "Towards Collaborative Robots for Infrastructure Security Applications." *Proceedings of the International Symposium on Collaborative Technologies and Systems*, San Diego, CA: 235-240.
- Heckmann, M., T. Rodemann, et al. (2006). "Auditory Inspired Binaural Robust Sound Source Localization in Echoic and Noisy Environments." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 368 - 373.
- Hetek. (2004, 07/22/03). "Gmic - Acoustic Leak Sounding System." Retrieved July 26, 2004, 2004, from <http://www.hetek.com/instruments/waterleak/gmic.shtml>.
- Hornstein, J., M. Lopes, et al. (2006). "Sound Localization for Humanoid Robots - Building Audio-Motor Maps based on the HRTF." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 1170-1176.
- Howard, A. (2004). "Simple mapping utilities (pmap)." Retrieved 3/19/2007, from <http://www-robotics.usc.edu/~ahoward/pmap/index.html>.
- Hu, J., W. Liu, et al. (2006). "Location and Orientation Detection of Mobile Robots Using Sound Field Features under Complex Environments." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 1151-1156.
- Huang, J., N. Ohnishi and N. Sugie (1997). "Sound Localization in Reverberant Environment based on the Model of the Precedence Effect." *IEEE Trans. Instrumentation and Measurement*, vol 46(4): 842-846.
- Huang, J., N. Ohnishi, et al. (1997). "Mobile Robot and Sound Localization." *Proceedings of the Intelligent Robotics and Systems (IROS)*: 683-689.
- Hughes, K., A. Tokuta, et al. (1992). "Trulla: An algorithm for path planning among weighted regions by localized propogations." *Proceedings of the Int. Conf. on Intelligent Robots and Systems (IROS)*, Raleigh, NC: 469-476.
- Hyde, P., and Knudsen, E.I (2000). "A topographic projection from the optic tectum to the auditory space map in the inferior colliculus of the barn owl." *Journal of Computational Neurology*, vol 421: 146-160.

- Hyde, P. S., and Knudsen, E.I. (2000). "The optic tectum controls visually guided adaptive plasticity in the owl auditory space map." *Nature*, vol 415: 73-76.
- Ingham, N. J., S. K. Thornton, et al. (1998). "Age-related changes in auditory spatial properties of the guinea pig superior colliculus." *Brain Research*, vol 788(1): 60-68(9).
- Isoda, S., M. Maeda, et al. (2003). "Realization of an Autonomous Search for Sound Blowing Parameters for an Anthropomorphic Flutist Robot." *Proceedings of the Int. Conf. on Robotics and Automation (ICRA)*, Taipei, TW: 3582-3587.
- Jeffress, L. A. (1948). "A place theory of sound localization." *J. Comp Physiol. Psychol.*, vol 41: 34-39.
- Johansson, R. (2003). *Information Acquisition in Data Fusion Systems*, Ph.D., Dept of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden
- Jones, J., A. Flynn, et al. (1999). *Mobile Robots: Inspiration to Implementation*. Natick, MA, A.K. Peters, Ltd.
- Junqua, J. C. (1993). "The Lombard Reflex and its Role on Human Listeners and Automatic Speech Recognizers." *Journal of the Acoustical Society of America*, vol 93(1): 510-524.
- Kaelbling, L. and S. Rosenschein (1991). "Action and Planning in Embedded Agents." *Designing Autonomous Agents*. P. Maes. Cambridge, MA, MIT Press: 35-48.
- Kaess, M., R. C. Arkin, et al. (2003). "Compact Encoding of Robot-Generated 3D Maps for Efficient Wireless Transmission." *Proceedings of the IEEE Intl. Conf. on Advanced Robotics*, Coimbra, Portugal: 324-331.
- Katijani, M. (1989). "Development of musician robots." *Journal of Robotics & Mechatronics*, vol 1: 254-255.
- Keller, C. H., K. Hartung, et al. (1998). "Head related transfer functions of the barn owl: Measurement and neuronal responses." *Hearing Res*, vol 118: 13-34.
- Kennedy, W. G., M. Bugajska, et al. (2007). "Spatial Representation and Reasoning for Human-Robot Collaboration." *Proceedings of the AAI-2007*, Vancouver, Canada: 1554-1559.
- Kettebekov, S., Yeasin, M., Sharma, R (2002). "Based Co-analysis for Continuous Recognition of Coverbal Gestures." *Proceedings of the International Conference on Multimodal Interfaces (ICMI'02)*, Pittsburgh, USA: 161-166.

- Kogan, J. A. and D. Margoliash (1998). "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study." *Journal of the Acoustical Society of America*, vol 103(4): 2185.
- Koizumi, T., N. Tsujiuchi, et al. (2002). "Prediction of the Vibration in Buildings Using Statistical Energy Analysis." *Proceedings of the International Modal Analysis Conference (IMAC)*, Los Angeles, CA, vol 1: 7-13.
- Korany, N. O. (2000). *A model for the Simulation of Sound Fields in Enclosures Integrating the Geometrical and the Radiant Approaches*, Faculty of Engineering, Alexandria University, Alexandria, Egypt
- Krotkov, E. (1995). "Robotic Perception of Material." *Proceedings of the International Joint Conference Artificial Intelligence*: 88-94.
- Lamere, P., P. Kwok, et al. (2003). "Design of the CMU Sphinx-4 Decoder." *MITSUBISHI ELECTRIC RESEARCH LABORATORY Technical Report, TR-2003-110*.
- Langer, B. and A. Black (2005). "Using Speech in Noise to Improve Understandability for Elderly Listeners." *Proceedings of the ASRU*, San Juan, Puerto Rico: 112-116.
- Liu, J., M. Wang, et al. (2005). "iBotGuard: An Internet-Based Intelligent Robot Security System Using Invariant Face Recognition Against Intruder." *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol 35(1): 97-105.
- Lukowicz, P., J. Ward, et al. (2004). "Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers." *Proceedings of the Pervasive Computing*, Vienna, Austria: 18-22.
- Luo, R., C. Lai, et al. (2006). "Rapid Environment Identification for Intelligent Mobile Robot." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 1158-1163.
- Martinson, E. (2007). "Hiding the Acoustic Signature of a Mobile Robot." *Proceedings of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*. San Diego, CA.
- Martinson, E. and R. C. Arkin (2004). "Noise Maps for Acoustically Sensitive Navigation." *Proceedings of SPIE*, vol 5609.
- Martinson, E. and D. Brock (2007). "Improving Human-Robot Interaction through Adaptation to the Auditory Scene." *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, Washington, DC: 113-120.

- Martinson, E. and F. Dellaert (2003). "Marco-Polo Localization." *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, TW, vol 2: 1960-1965.
- Martinson, E. and A. Schultz (2006). "Auditory Evidence Grids." *Proceedings of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*. Beijing, China: 1139-1144.
- Martinson, E. and A. Schultz (2007). "Robotic Discovery of the Auditory Scene." *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Rome, Italy: 435-440.
- Marzouqi, M. and R. Jarvis (2005). "Fast Visibility Evaluation for Covert Robotics Path Planning." *Proceedings of the IEEE Int. Workshop on Safety, Security, and Rescue Robotics*, Kobe, Japan: 48-53.
- Maurer, U., A. Rowe, et al. (2006). "eWatch: A Wearable Sensor and Notification Platform." *Proceedings of the Int. Workshop on Wearable and Implantable Body Sensor Networks*, Cambridge, MA: 4pp. - posted online.
- Miles, D. (2006). "New Device Will Sense Through Concrete Walls." Retrieved Feb 21, 2007, from [http://www.defenselink.mil/news/Jan2006/20060103\\_3822.html](http://www.defenselink.mil/news/Jan2006/20060103_3822.html).
- Mungamuru, B. and P. Aarabi (2004). "Enhanced Sound Localization." *IEEE Trans. on Systems, Man, and Cybernetics*, vol 34(3).
- Nakadai, K., D. Matsuura, H.G. Okuno, and H. Kitano. (2003). "Applying Scattering Theory to Robot Audition System: Robust Sound Source Localization and Extraction." *Proceedings of the Int. Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, vol 2: 1147-1152.
- Nakadai, K., K. Hidai, H.G. Okuno, and H. Kitano. (2001). "Epipolar Geometry Based Sound Localization and Extraction for Humanoid Audition." *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Maui, Hawaii: 1395 - 1401.
- Nakadai, K., H. Nakajima, et al. (2006). "Real-Time Tracking of Multiple Sound Sources by Integration of In-Room and Robot-Embedded Microphone Arrays." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 852-859.
- Naylor, G. M. (1993). "ODEON - Another Hybrid Room Acoustical Model." *Applied Acoustics*, vol 38: 131-143.
- Noll, P. (1998). "MPEG Digital Audio Coding Standards." *The Digital Signal Processing Handbook*. V. K. Madisetti and D. B. Williams, IEEE Press/ CRC Press: 40-1 - 40-28.

- O'Keefe, J. a. G. S., and John Bradley (1998). "Acoustical renovation of The Orpheum Theatre." *Proceedings of the International Congress on Acoustics*, Vancouver, Canada.
- Painter, T. and A. Spanias (2000). "Perceptual Coding of Digital Audio." *Proceedings of the IEEE*, vol 88(4): 451-515.
- Parker, L., B. Kannan, et al. (2003). "Heterogeneous Mobile Sensor Net Deployment Using Robot Herding and Line-of-Sight Formations." *Proceedings of the Intelligent Robotics and Systems (IROS)*, Las Vegas, NV: 2488 - 2493.
- Peltonen, V., J. Tuomi, et al. (2002). "Computational Auditory Scene Recognition." *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Orlando, FL, vol 2: 1941-1944.
- Perzanowski, D., A. Schultz, et al. (2000). "Using a Natural Language and Gesture Interface for Unmanned Vehicles." *Proceedings of the Unmanned Ground Vehicles II, Aerosense 2000*, vol 4024: 341-347.
- Philips. (2004, March, 2004). "Audio Fingerprinting for Automatic Music Recognition." Retrieved Nov. 18, 2004, 2004, from [http://www.research.philips.com/initiatives/contentid/downloads/audio\\_fingerprinting\\_leaflet.pdf](http://www.research.philips.com/initiatives/contentid/downloads/audio_fingerprinting_leaflet.pdf).
- Pierce, A. (1989). *Acoustics, An Introduction to Its Physical Principles and Applications*. Woodbury, NY, Acoustical Society of America.
- Predko, M. (2003). *Programming Robot Controllers*. New York, NY, McGraw Hill.
- Quatiri, T. (2002). *Discrete Time Speech Signal Processing*. Dehli, India, Pearson Education.
- Rabiner, L. R. and R. W. Schafer (1978). *Digital Processing of Speech Signals*. Englewood Cliffs, NJ, Prentice Hall, Inc.
- Ragheb, H. and E. R. Hancock (2003). "Estimating surface characteristics using physical reflectance models." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, WI, vol 2: 177-184.
- Raichel, D. R. (2000). *The Science and Applications of Acoustics*. New York, NY, Springer-Verlag.
- Ravindran, S. (2006). *Physiologically Motivated Methods for Audio Pattern Classification*, Ph.D. Thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Ga, USA

- Roy, N., G. Baltus, et al. (2000). "Towards personal service robots for the elderly." *Proceedings of the Proceedings of the Workshop on Interactive Robotics and Entertainment (WIRE)*, Pittsburgh, PA.
- Rucci, M., G. Tononi, et al. (1997). "Registration of neural maps through value-dependent learning: modeling the alignment of auditory and visual maps in the barn owl's optic tectum'." *Journal of Neuroscience*, vol 17(1): 334-352.
- Russell, S. and P. Norvig (1995). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, Prentice Hall.
- Sasaki, Y., S. Kagami, et al. (2006). "Multiple Sound Source Mapping for a Mobile Robot by Self-motion Triangulation." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 380-385.
- Savioja, L. (1999). *Modeling Techniques for Virtual Acoustics*, Ph.D., Helsinki University of Technology, Helsinki, Finland
- Scheutz, M., P. Schermerhorn, et al. (2006). "The Utility of Affect Expression in Natural Language Interactions in Joint Human-Robot Tasks." *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, Salt Lake City, UT: 226-233.
- Schiele, B. and S. Antifakos (2002). "Beyond Position Awareness." *Proceedings of the Personal and Ubiquitous Computing(2002)*, Springer-Verlag, vol 6: 313-317.
- Schulte, J., C. Rosenberg, et al. (1999). "Spontaneous Short-term Interaction with Mobile Robots in Public Places." *Proceedings of the Int. Conf. on Robotics and Automation (ICRA)*, vol 1: 658-663.
- Schultz, A. and W. Adams (1998). "Continuous localization using evidence grids." *Proceedings of the IEEE International Conf. on Robotics and Automation*, Leuven, Belgium, vol 4: 2833-2839.
- Shamma, S. A., N. Shen and P. Gopalswamy (1989). "Stereoausis: Binaural Processing without Neural Delays." *Journal of The Acoustical Society of America*, vol 86: 999-1006.
- Simmons, R., D. Goldberg, et al. (2003). *GRACE: An Autonomous Robot for the AAI Robot Challenge*. AAAI Magazine.vol 24: 51-72.
- Slaney, M. (1994). *The Auditory Toolbox*", Apple Computer Company, Apple Technical Report #45.
- Smaragdīs, P. (2001). *Redundancy Reduction for Computational Audition, a Unifying Approach*, Ph.D., Media Laboratory, Massachusetts Institute of Technology, Cambridge

- Sony. (2004). "QRIO's Technology." *QRIO* Retrieved Aug 29, 2004, 2004, from [http://www.sony.net/SonyInfo/QRIO/technology/index3\\_nf.html](http://www.sony.net/SonyInfo/QRIO/technology/index3_nf.html).
- Stager, M., P. Lukowicz, et al. (2003). "SoundButton: Design of a Low Power Wearable Audio Classification System." *Proceedings of the Proc. of the IEEE Int'l Symposium on Wearable Computing (ISWC)*, New York: 12-17.
- Strobel, N., S. Sascha, R. Rabenstein (2001). "Joint Audio-Video Signal Processing for Object Localization and Tracking." *Microphone Arrays*. D. W. M. Brandstein. Berlin, Germany, Springer.
- Svensson, P. (2002). "Modelling Acoustic Spaces for Audio Virtual Reality." *Proceedings of the IEEE Benelux Workshop on Model Based Processing and Coding of Audio*, Leuven, Belgium: 109-116.
- Takahashi, T. T. and C. H. Keller (1994). "Representation of multiple sound sources in the owl's midbrain." *Journal of Neuroscience*, vol 14: 4780-4793.
- Takeda, R., S. Yamamoto, et al. (2006). "Missing-Feature based Speech Recognition for Two Simultaneous Speech Signals Separated by ICA with a Pair of Humanoid Ears." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China: 878-885.
- Thrun, S. (2002). "Robotic Mapping: A Survey." *Exploring Artificial Intelligence in the New Millenium*. G. Lakemeyer and B. Nebel. San Francisco, CA, Morgan Kaufman.
- Thrun, S. (2005). "Affine Structure from Sound." *Proceedings of the Advances in Neural Information Processing Systems*, Whistler, Canada: 1353-1360.
- Thrun, S., W. Burgard, et al. (2005). *Probabilistic Robotics*. Cambridge, MA, MIT Press.
- Thrun, S., D. Fox, et al. (2001). "Particle Filters for Mobile Robot Localization." *Sequential Monte Carlo Methods in Practice*. A. Doucet, N. de Freitas and N. Gordon. New York, NY, Springer-Verlag.
- Tremain, T. E. (1982). *The Government Standard Linear Predictive Coding Algorithm: LPC-10*. *Speech Technology Magazine*: p. 40-49.
- Treptow, A., G. Cielniak, et al. (2005). "Active People Recognition Using Thermal and Grey Images on a Mobile Security Robot." *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Alberta, Canada: 2103-2108.
- Valin, J. M. (2005). *Auditory System for a Mobile Robot*, Ph.D., Faculte de genie, University of Sherbrooke, Sherbrooke, Canada

- Valin, J. M., J. Rouat, et al. (2004). "Enhanced Robot Audition Based on Microphone Array Source Separation with Post-Filter." *Proceedings of the IEEE/RSJ Int. Conf on Intelligent Robots and Systems*, Sendai, Japan, vol 3: 2123-2128.
- Venkatagiri, H. S. (2003). "Segmental intelligibility of four currently used text-to-speech synthesis methods." *Journal of the Acoustical Society of America*, vol 113(4): 2095-2104.
- Waseda. (2000). "Humanoid History, Booklet2000." Humanoid History Retrieved April 5, 2001, 2001, from [http://www.phys.waseda.ac.jp/humanoid/booklet/booklet2000\\_takanishi.html](http://www.phys.waseda.ac.jp/humanoid/booklet/booklet2000_takanishi.html).
- Webb, B. (1998). "Robots crickets and ants: models of neural control of chemotaxis and phonotaxis." *Neural Networks*, vol 11: 1479-1496.
- Williams, K. (2004). *Build Your Own Humanoid Robots: 6 Amazing and Affordable Projects*. New York, NY, McGraw-Hill.
- Wilson, C. E. (1994). *Noise Control: Measurement, Analysis, and Control of Sound and Vibration*. Malabar, FL, Krieger.
- Yamasaki, N., Y. Anzai (1995). "Active Interface for Human-Robot Interaction." *Proceedings of the Int. Conf. on Robotics and Automation (ICRA)*: 3103-3109.
- Young, S. H. and M. V. Scanlon (2001). "Robotic vehicle uses acoustic array for detection and localization in urban environments." *Proceedings of the SPIE*, vol 4364: 264-273.
- Zhang, B. and G. Sukhatme (2005). "Controlling Sensor Density using Mobility." *Proceedings of the Second IEEE Workshop on Embedded Networked Sensors*, Sydney, Australia: 141 - 149.
- Zheng, F., G. Zhang, et al. (2001). "Comparison of Different Implementations of MFCC." *Journal of Computer Science & Technology*, vol 16(6): 582-589.