Marco Polo Localization

Eric Beowulf Martinson and Frank Dellaert

Georgia Institute of Technology
College of Computing
Atlanta, GA 30332-0280
{ebeowulf, dellaert}@cc.gatech.edu

Abstract

We introduce the Marco Polo Localization approach, where we apply sound as a tool for gathering range measurements between robots, and use those to solve a range-only Simultaneous Localization and Mapping problem. Range is calculated by correlating two recordings of the same sound, recorded on a pair of robots, after which the resulting time delay estimate is converted to a range measurement. The algorithmic approach we use is a straightforward application of the Bayesian estimation framework. We also present two complementary views on the associated optimization problem that provide insight into the problem and allows one to devise initialization strategies, indispensable in a range-only scenario. We illustrate the approach with both simulated and experimental results.

1 Introduction

In this paper, we apply sound as a tool for gathering the range measurements between robots, and use those to solve a range-only Simultaneous Localization and Mapping (SLAM) [1] problem. In many mobile robot applications it is essential to obtain an accurate metric map of a previously unknown environment, and to be able to accurately localize the robot(s) within it. The process of reconstructing such a map from odometry and sensor measurements collected by one or more robots is known as SLAM. By applying sound as a sensor for gathering range estimates, we hope to demonstrate the feasibility of using sound for localization and to increase the generality of the SLAM problem.

The name "Marco Polo Localization" is inspired by a children's game, commonly played pool-side, where a blindfolded person who is "it" shouts "Marco", and everybody else has to shout "Polo" to reveal where they are. From these audio cues, the person who is "it" has to catch and tag someone else.

However, the inspiration for using sound as a localization mechanism comes from biology. People as well as animals are able to estimate both bearing and range information to a sound source using a combination of inter-aural time differences (ITDs) and intensity. An ITD is a measure of the difference in the time of arrival of a sound at each ear, allowing us to recognize bearing, while range is correlated with the intensity and spectral characteristics of the sound. Exactly how people acquire ITDs is still being debated by the psychological community[2, 3], but they have become the standard model for sound localization work in digital signal processing (DSP).

The motivation for using sound in stead of or complementing an alternative sensing modality is because microphones and speakers can be included cheaply and compactly on small robots, in contrast to more bulky ultrasound, vision, or laser-based sensors. In addition, the approach is insensitive to lighting conditions, but is of course prone to interference by noise. The latter disadvantage can be alleviated by working in the ultrasound spectrum and filtering the sound as part of the pre-processing. Hence, it has potential applications as a sensor for 24 hour operations, where a variety of very different lighting conditions would arise. One obvious application is underwater robotics, where sound-waves have long been used by man and animal alike to perform localization and mapping.

The next section (Section 2) gives an overview of how sound has been used in the DSP community and in robotics, and discussed other works related to our approach. Section 3 discusses the technical details of the approach, Section 4 describes the simulation results for stationary and moving robot formations, and Section 5 describes implementation details on real robots and experimental results.

2 Related Work

Within the signal processing community, ITDs are relatively easy to acquire [4] and have long been used as a localization tool. Typically, some number of microphones are statically located somewhere in an environment, and used to identify the location of a moving sound source such as speech. Because sound travels at a finite speed, the sound arrives at each microphone at a different moment in time depending on how far away it is from the sound source in addition to some other properties of the environment. Using a cross-correlation algorithm [4] on the sound stream, one can determine the difference in the time of arrival at each microphone, the ITD, and thereby the range to the sound source. With enough microphones, it is possible to identify coordinates in 3-dimensions relative to the known positions of each microphone.

Humans have always used sound as a tool for localization, but its application to robotics has been limited. The traditional application of sound in robotics has been to find bearing measurements to to a sound source [5, 6]. Typically, this is only used to guide the robot when other sensors cannot detect the desired target. Noise from the robots, and from the environment tends to obscure sounds and make accurate range information from a microphone pair difficult to obtain. This work however, focuses on using those range measurements by thresholding physically impossible results, and combining what remains with odometry to reduce noise.

To the best of our knowledge, the application of sound to the SLAM problem has not been done. Unlike the localization problem in DSP, we know that the sound source is co-located with a microphone, but all of the microphone locations remain unknown. Given enough robots, the resulting range measurements can be transformed into the relative positions of all robots in the room. Range-only SLAM itself is a newly emerging field and has only been explored recently. Work by Kantor and Singh [7] at CMU uses radio beacons in the environment with known positions. A set of beacons measures the distance to the robot, and transmit their measurements and unique id to the robot. The robot then determines where it was using a variant of evidence grids. Some strongly related work was performed by LeMaster and Rock [8] for self-calibrating pseudolite arrays. A more sophisticated approach, based on multivariable optimization, was proposed by Newman and Leonard [9] in the area of sub-sea mapping, although the details are as yet unpublished.

The approach taken in this paper follows the same strategy, by casting the problem as a multi-variable Bayesian estimation problem, but one where the robots themselves are used as landmarks. As such it is an application of the more general Intrinsic Mapping and Localization (IML) framework that we have put forward in [10]. Recent work at USC [11] takes a similar optimization approach for the case when robots can sense the orientation of other robots in addition to range *and* bearing. The range-only case discussed here is more difficult, because of the ambiguities and initialization problems involved. The idea of using *stationary* robots as landmarks has been around for a while, and dates back to Kurazume [12, 13] and later work by Rekleitis et al. [14].

3 Marco Polo Localization

The problem at hand is to estimate the *poses* X of all robots at all times given *odometry data* O and range measurements Z. We will assume that the identity of each robot can be established by the sound emitted. In terms of notation, we will refer to the entire set of sought poses as X, whereas the poses of one robot only are denoted as X_i, with $i \in 1..m$, and the poses of all robots at a specific time t as X^t, with $t \in 1..T$. The pose of robot i at time t is denoted as x_i^t. Similar conventions are used for the odometry O and the range measurements Z. The range measurement between robots i and j, if available, is written as z_{ij}. We define $|e|^2_\Sigma \triangleq e^T \Sigma^{-1} e$ to be the squared Mahalanobis distance with covariance matrix Σ.

3.1 Odometry Measurements

Given no other information, the maximum a posteriori (MAP) trajectory $\hat{X}_i$ of a single robot i given odometry O_i is obtained simply by integrating the odometry over time. If no prior on the initial pose x_i^1 is available, the trajectory can be determined *up to a 2D displacement only*, i.e., an arbitrary translation and rotation in the plane. If a prior is available, there is no remaining ambiguity. In detail, the MAP trajectory is found by maximizing the posterior probability

$$P(X_i|O_i) \propto P(X_i)P(O_i|X_i) = P(x_i^1)\prod_t P(o_i^t|x_i^t, x_i^{t+1}) \qquad (1)$$

where we make the usual conditional independence assumptions, and the only prior knowledge available is a guess $\vec{x}_i^1$ for the initial pose x_i^1. In the case of normally distributed measurement noise, the associated error to be minimized is equal (up to a constant) to the negative log-posterior, given by

$$E_{oi} \triangleq |x_i^1 - \vec{x}_i^1|^2_Q + \sum_t |o_i^t - g(x_i^t, x_i^{t+1})|^2_R \qquad (2)$$

where $g(x,y)$ is the odometry measurement function between two poses x and y, and Q and R are the covariances for the prior on x_i^1 and odometry measurements o_i^t, respectively[1]. The error (2) will be exactly zero for the MAP trajectory, *iff* only odometry information is available, regardless of whether a prior $P(x_i^1)$ is available or not.

3.2 Range Measurements

Similarly, assuming no prior for now, at each time-step t we can obtain a maximum likelihood estimate $\hat{X}^t$ for the *configuration* of poses X^t given only the range measurements Z^t at time t, by maximizing

$$P(Z^t|X^t) = \prod_{ij} P(z_{ij}|x_i^t, x_j^t) \qquad (3)$$

or, alternatively, minimizing the associated error:

$$E_z^t = \sum_{ij} |z_{ij} - h(x_i^t, x_j^t)|^2_S \qquad (4)$$

where $h(x,y)$ is the *range measurement function* associated with the ordered pair of poses x and y, S is the noise covariance for each set of measurements, and the summation is over all pairs (i,j) where a range measurement is available. Since the orientation of the robots is not observable using only range, the number of unknown parameters is $2m$, i.e., x and y position for each robot. However, any solution can only be determined up to a 2D displacement and an orientation flip. In addition, the problem can be solved only if the number of measurements K is larger than $2m-3$, the degrees of freedom (DOF) of the system. Since in the best case, the maximum number of (non-redundant) range measurements is equal to $K_{max} = \binom{m}{2}$, is is easily checked that at least 4 robots need to be available in order to converge to a single solution (up to the stated ambiguity), as in that case there are potentially 6 measurements to constrain 8 unknowns, up to a 3DOF displacement (and a flip).

3.3 The Global Optimization problem

The global problem can now be seen as combining these two estimation problems, i.e., obtaining the MAP estimate for

[1] Note that in general these can be different for each measurement, but explicit time and robot indices are omitted to unburden the notation.

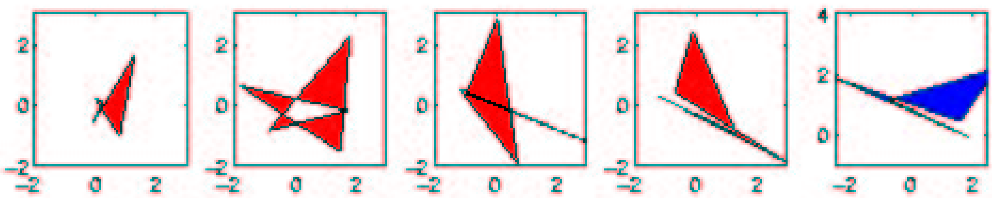

Figure 1: Given a set of range estimates with no odometry, these are the intermediate steps of the Levenberg Marquardt algorithm in determining 2D coordinates for all robots. The resulting configuration is an inverted transformation of the true values (last frame).

the poses X for all robots $1..m$ and times $1..T$, given the odometry O and the range measurements Z:

$$\hat{X} = \underset{X}{\operatorname{argmax}}\ P(X|O,Z) = \underset{X}{\operatorname{argmax}}\ P(X)P(O|X)P(Z|X)$$

or, alternatively, minimizing the following error function:

$$E \triangleq \textstyle\sum_i E_{oi} + \sum_t E_z^t =$$
$$\textstyle\sum_i |x_i^1 - \bar{x}_i^1|_Q^2 + \sum_{it} |o_i^t - g(x_i^t, x_i^{t+1})|_R^2 + \sum_{tij} |z_{ij} - h(x_i^t, x_j^t)|_S^2$$

We use a non-linear optimization method, Levenberg-Marquardt with a sparse QR solver, to obtain the MAP estimate in a batch optimization procedure. To compute the (sparse) Jacobian $\frac{\partial E}{\partial X}$ we have implemented an automatic differentiation (AD) framework. AD is neither symbolic nor numerical differentiation, and calculates the Jacobian at any given value exactly, efficiently, and free of numerical instabilities. See [15] for more details.

4 Simulation results

The scenario we use is as follows: a group of robots are scattered about the environment in unknown starting positions. They each generate a sound, and time differences are gathered between each of the robots, allowing us to generate range estimates. We call this a *slice*, as it reveals the position of the robots at one point in time. After sounding off, the robots move about the room, recording their odometry. At some time later, the robots are stopped, and another slice is gathered. This procedure is repeated throughout the experiment.

The simulation results can be divided into two types of scenarios: the slice view, and the track view. In the slice view, every slice which can be calculated is first calculated from the range measurements. Then the odometry measurements are used to orient the slices in space. In the track view, the odometry of each robot is first held to be correct, i.e., it is known how the robot travels through space from an initial position. The range measurements are used to align these rigid tracks translationally and rotationally with each other.

4.1 Slice View

In the slice view, we look at each individual slice in which range measurements were acquired. If we do not need to recover the angle of the robot for a slice, then each slice only requires $2m-3$ measurements, up to a rotationally, translationally, and inverted transform of the true value. Figure 1 shows how an initial estimate for a slice is transformed into the correct geometry using the Levenberg-Marquardt algorithm. At each step, the points of a geometric shape represent the positions of each robot in that estimate. The last shape in the series is the shape of the true geometry.

However, the configuration displayed in Figure 1 is "flipped" or inverted, which demonstrates a fundamental difficulty with range-only measurements. From an algorithmic standpoint, this is a correct answer unless prior information is included. From the robot localization standpoint, this inverted geometry needs to be corrected in order to properly map the progress of each robot.

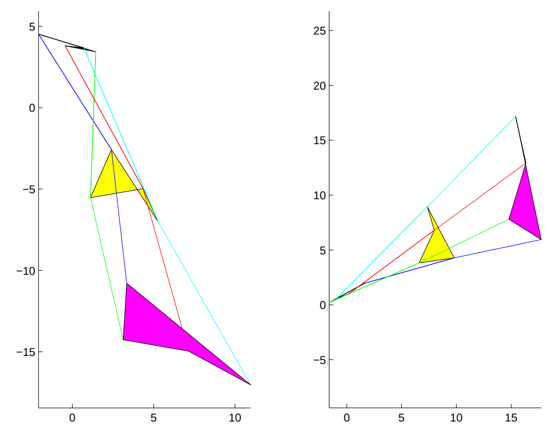

Figure 2: When using odometry to align a set of slices, the system regularly falls into configurations with inverted intermediate steps.

With a single slice, the flipping is wrong for mapping, but algorithmically correct. With odometry included however, "flipped" solutions are no longer correct because the angles would not align with those measured by the robot. Unfortunately, the "flipping" of individual slices introduces a large number of local minima to the system. So much so that the equation solver cannot reach the correct solution. Figure 2 (left) displays a local minima solution obtained with the slice view, where an intermediate slice has been inverted.

Although the slice view regularly converges (94% of 1000, using 4 robots) to a correct, albeit transformed, solution with individual slices, it rarely converges to the correct solution using multiple slices and odometry (4% of 1000 trials).

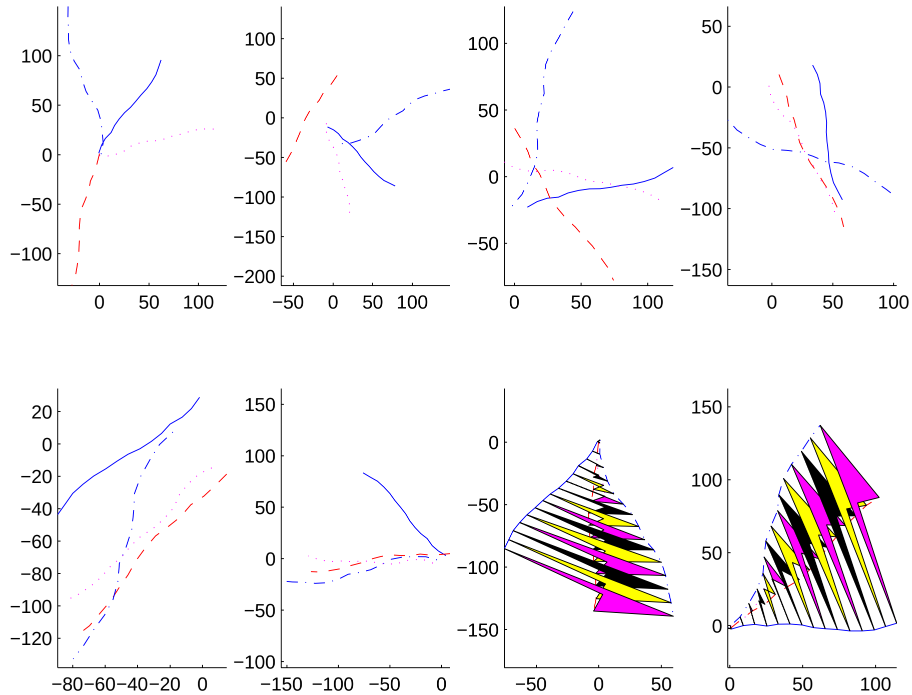

Figure 3: If we hold the tracks to be rigid, then finding the global solution is a matter of rotating and aligning all of the individual tracks. This figure shows the intermediate steps generated by the Levenberg Marquardt algorithm, using the track view method, in left-to-right reading order. For clarity, the last two sub-figures also show the slice shapes, respectively for the final solution and the ground truth, which only differ up to a 2D displacement.

4.2 Track View

In the track view, the odometry of each robot is held inviolate, creating a set of tracks through space. These rigid tracks are then moved and rotated into the correct relative position, through the range measurements gathered over the duration of the experiments. The track view method requires $3m-2$ range measurements, in order to identify starting positions and angles of each the robots. Figure 3 shows how the tracks are transformed through space into the correct configuration in the last frame, using the Levenberg Marquardt algorithm. Note that by choosing the track method, we are holding the odometry to be more reliable than the range measurements generated through time difference estimates.

Like the slice view, the track view still has a local minima problem due to an inverted first figure. However, if we detect that the system has fallen into a local minima, we can invert the initial guess $[M = -M]$ passed into the system and pass that instead. With this trick, if we generate a set of values for 4 microphones and 15 slices, then in 18.2 out of 20 (91%) runs on average, the system of equations settled in the correct answer. These numbers were generated over 100 simulated configurations with no error in the measurements.

We can improve performance even further by bootstrapping the track view algorithm with a known slice. If we have enough range measurements to determine the x,y coordinates without angle information for any one slice during testing, then in 19.5 out of 20 (97.5%) runs on average, the system settles in the correct solution

5 Experimental Results

Figure 4: Robots used for the experiments. The wires lead from the microphones to a desktop machine off screen.

Experiments were run on four Nomad 150 robots, shown in Figure 4, equipped with laptops and a wireless connection. Each Nomad had a speaker mounted on one side of the laptop, and a microphone mounted on the other. In order to synchronize the recordings between channels, microphones were plugged into a single desktop computer with a 16bit sound card. The digital sampling quality of all recording was performed at 16 bit quality, and 22,050 Hz. While the microphones used were wired, the system can in principle be implemented wirelessly without difficulty.

5.1 Experimental Procedures

Distance estimates were gathered from the robots in pairs. One robot played a sound while recording data, while a second robot was also recording data. These two files were then saved. Then the robot started recording again and repeated the sound while another robot was listening. This process repeated until every pair of robots had recorded a time-delay between them. We were limited to two robots at a time, because the standard sound we used was limited to recording two channels at a time. In principle, one sound could be used to generate $n-1$ readings using the appropriate hardware to synchronize n channels.

Once all the data was recorded, each pair of sound files is compared using a cross-correlation algorithm to find the time-delay between the two channels. The sound used was a clicking noise that was experimentally determined to provide the best time-delay estimates using cross-correlation. The software used for the actual cross-correlation was Ishmael 1.0, developed by the Office of Naval Research[16]. If the cross-correlation algorithm returned an estimated time-delay greater than 15 ms., then that measurement was discarded. This threshold value was experimentally determined to be the physical limit of the microphones/amplifier used.

5.2 Stationary Robots

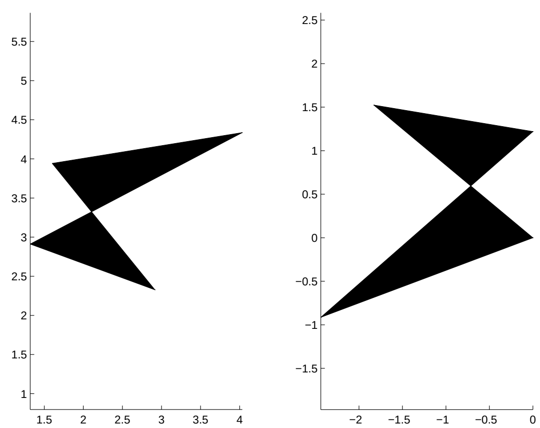

Figure 5: (left) Reconstructed slice for real robots. (right) Actual configuration in meters.

Using 4 Nomad 150 robots equipped with a microphone and a speaker, we could measure the relative position of all robots in the room. The left of Figure 5 shows the resulting configuration from a set of 6 time-delay measurements. On the right, we see the actual configuration of robots in the room. The points of each geometric figure are the positions of the robots.

From a stationary position with only 4 robots, we can experience lots of noise in the recordings. If a robot was located in a corner, or some other acoustically difficult location, then sounds could be obscured by echoes or not heard at all. In those cases, we received bad time-delay estimates that make accurate reconstruction of the robot positions difficult. During testing, out of 5 sets of recorded data for individual slices, one was not recognizable at all, while two had obvious discrepancies from the correct position.

5.3 Results with Odometry

Test	# Bad TD	# Good TD	Average Error
1	4	14	0.313 m
2	8	10	0.416 m
3	8	10	0.288 m

Table 1: Statistics on the time-difference estimates calculated during each test.

In practice, a large number of the actual time-delay measurements will not be available to help align the tracks. Especially as robots start to move around in the environment, it occurs more often that they are located in positions where a good time-delay estimate is difficult to obtain. On the experiments with 4 robots and 3 stops, three tests recorded 14, 10, and 10 useful time-delay estimates out of a possible 18. The rest were removed by thresholding.

The statistics revealed in table 1 for each of the tests demonstrate an average error from the true values for the range measurements. Actual bad measurements which were not removed by thresholding were rare, and did not influence the data much in small tests. The error here is mostly due to the off centered position of the microphones on each robot. Microphones could not be placed exactly center because of existing equipment on the robots.

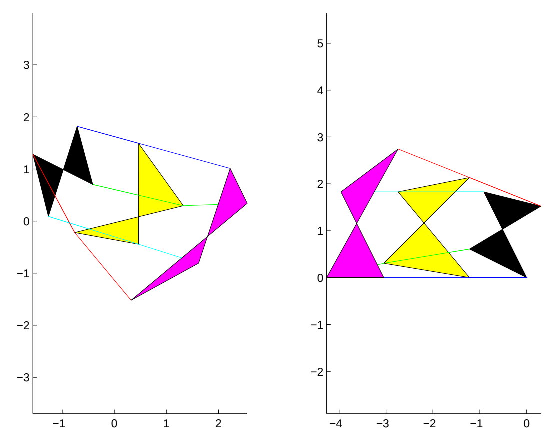

Figure 6: (left) Experimental results for 4 robots, and 3 slices using the track view method. (right) True positions. In this experimental run, 14 range measurements were used to align the 4 rigid tracks.

Unfortunately, with that amount of average error, we did get some skew in the output. Figure 6 displays the reconstructed results from test 1, using the track view method. Two changes to procedure would help correct this error. First, locating the microphones as close to the center as possible or including their position on the robot into the model. Second, obtaining more range estimates. With 10 measurements, there is only one extra measurement for removing noise.

6 Conclusion

The range-only Simultaneous Localization and Mapping method we presented is applicable to any multi-robot scenario where sound or any other range measurement between robots is available. In this paper, we demonstrated this both with large scale simulation experiments as well as with preliminary experimental results, using sound as a measurement device. The method is applicable to determining stationary configurations, as well as recovering trajectories and relative positions of a robot team over time.

However, for sound-based range estimation, we found that the success of the method depends crucially on the quality and the robustness by which the time differences can be recovered. In our current setup, we are plagued by occasional outliers due to erroneous correlations, which we cannot identify other than comparing them to the ground truth values. Success on a larger scale will depend on better identifying such outliers or avoiding them altogether, for example by using more structured sounds that afford less opportunity for error.

A number of additional improvements are possible. The robots are currently required to stop and sound off to gather time-delay estimates, because the sound emitted is obscured by the noise the robot itself generates while moving. If another sound which is not obscured by noise can be located, then the robots could potentially be gathering range estimates while moving around in the environment.

Our current implementation uses the stereo sound-card of a nearby desktop computer to establish the range between two robots. Clearly this is not the way such a system would be used outside the lab. We would like to investigate whether we can reliably correlate the sound recorded by one robot with the waveform file, given that this can be distributed among all robots. This scheme would still require a synchronized clock, which is not always obvious in a wirelessly networked environment. Thus, an obvious question is whether we can synchronize the clocks of the different robots using the sounds themselves, as part of the optimization process.

Finally, can we get relax the requirement that robots generate the sounds themselves? With enough robots, the algorithm could potentially solve for the problem with both unknown source and unknown sink locations. However, which sounds in the environment can we use for localization, how do we recognize them as such, and if those problems are solved, can we keep the algorithm from settling in a local minimum ?

Acknowledgments

Gabriel Brostow suggested the name Marco Polo localization. Gregory Abowd and Ron Arkin graciously provided hardware for the robot experiments.

References

[1] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE Int. Workshop on Intelligent Robots and Systems*, pp. 1442–1447, 1991.

[2] L. Jeffress, "A place theory of sound localization," *Journal of Comp. Physiol. Psychol.*, vol. 41, pp. 34–39, 1948.

[3] S. Shamma, N. Shen, and P. Gopalaswamy, "Stereausis: Binaural processing without neural delays," *Journal of The Acoustical Society of America*, vol. 86, pp. 999–1006, 1989.

[4] C. Knapp and G. Carter, "The generalized correlatoon method for estimation of time delay," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, 1976.

[5] J. Huang, T. Supaongprapa, I. Terakura, F. Wang, I. Ohnishi, and N. Sugie, "A model based sound localization system and its application to robot navigation"," *Journal of Robotics and Autonomous Systems*, vol. 27, pp. 199–209, 1999.

[6] K. Nakadai, K. Hidai, H. Okuno, and H. Kitano, "Epipolor geometry based sound localization and extraction for humanoid audition," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001.

[7] G. Kantor and S. Singh, "Preliminary results in range-only localization and mapping," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1818 –1823, 2002.

[8] E. LeMaster and S. Rock, "Field test results for a self-calibrating pseudolite array," in *Proceedings of Institue of Navigation GPS-2000 Conference*, 2000.

[9] P. Newman and J. Leonard, "Range-only subsea concurrent Mapping and Localization," in *ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots*, 2002.

[10] F. Dellaert, F. Alegre, and E. Martinson, "Intrinsic mapping and localization," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2003.

[11] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, September 2002.

[12] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2, pp. 1250–1257, 1994.

[13] R. Kurazume and S. Hirose, "An experimental study of a cooperative positioning system," *Autonomous Robots*, vol. 8, pp. 43–52, January 2000.

[14] I. M. Rekleitis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 7–40, 2001.

[15] A. Griewank, "On Automatic Differentiation," in *Mathematical Programming: Recent Developments and Applications* (M. Iri and K. Tanabe, eds.), pp. 83–108, Kluwer Academic Publishers, 1989.

[16] D. Mellinger, "Ishmael 1.0 user's guide. ishmael: Integrated system for holistic multi-channel acoustic exploration and localization," Tech. Rep. OAR PMEL-120, NOAA Technical Memorandum, 2002.