Perceptual support for ballistic motion in docking for a mobile robot

Douglas C. MacKenzie
Ronald C. Arkin
Mobile Robot Laboratory
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, 30332

Abstract

This paper describes ongoing research into methods to allow a mobile robot to effectively function in a manufacturing environment; specifically, generation of the ballistic motion phase of the docking behavior. This overall docking behavior causes the robot to move to a workstation and park in an appropriate position. The docking behavior consists of two distinct types of motion. Ballistic motion rapidly moves the robot to an area near the dock where recognition of the dock triggers the slower, more accurate orienting motion for the final positioning. The ballistic motion is supported with two simple low-level behaviors: a phototropic (light seeking) behavior and a temporal (motion) detection behavior. The phototropic or temporal activity perceptual strategy maneuvers the vehicle toward the bright light or the abundant motion usually associated with workstations. These vision algorithms have been selected because strict knowledge of the initial position of the dock is not needed and each requires limited computational resources. This system has been implemented and shown to successfully generate ballistic motion in support of docking in typical manufacturing environments.

1. Introduction

Designing a mobile robot to effectively function in a manufacturing environment is a difficult task. By their very definition, natural environments are constantly in a state of flux. Any intelligent agent attempting to function in such a world must be able to deal with this change in a rapid and robust manner.

Control strategies for an autonomous agent must cope with a changing environment, moving obstacles and potentially significant amounts of electrical, acoustical and visual noise. A route that was unobstructed yesterday may no longer be passable today. Landmarks that the vehicle has previously used for navigational aids may now be occluded, have changed position, or been removed completely. These very real problems add orders of magnitude to the complexity of the task.

The task can be simplified by partitioning the vehicle's motion into two distinct phases, ballistic motion and controlled motion. This partitioning has a strong biological basis [10, 14]. A system begins moving in a rapid, low accuracy mode toward the general area of the destination. This is called the *ballistic* phase of motion and is characterized by coarse external perception feedback requirements. At some point during the motion, a transition is made to a slower, more refined motion termed *controlled* motion. The transition can be triggered by an exteroceptive cue, internal estimates of position, elapsed time or other related events. Candidate exteroceptive cues include model-based visual recognition of the target, unexpected changes in the environment, etc. The controlled motion phase places a much higher demand on perception. The motions now must be carefully controlled since the system is in close proximity to the target. There typically is a coercive component to this motion that orients the system into the proper position for final positioning and motion termination [3].

This work concentrates on the perceptual aspects of the ballistic component of docking motion. The research on docking motor control is presented in detail elsewhere [3, 5]. In this paper, perceptual algorithms capable of supporting this motion in a Flexible Manufacturing System (FMS) environment are described and presented. Their usefulness is then explored in experiments designed to highlight their applicability and robustness in this domain.

2. Approach

The Autonomous Robot Architecture (AuRA) [6] provides the framework for our experimental implementations. Our approach to these problems is to use a reactive execution mechanism that is subservient to a hierarchical planner [4]. This reactive component copes with dynamic changes in the world. The system uses motor schemas (behaviors) that are selected and parameterized by a high level planner. These motor agents are single-minded in that they each only have one goal, such as "stay at least three feet away from that wall" or "stay in the middle of the path". Each agent generates a velocity vector which encodes its requirements for the vehicle's motion. The vectors from all active agents are summed to obtain the combined velocity acting on the vehicle. This velocity is then normalized for the robot, taking into account the physical constraints of the vehicle.

Single purpose perceptual schemas are also utilized to encapsulate perceptual processing. These schemas allow the system to sense only the information that will actually be used. This philosophy of action-oriented perception [2] reduces the computational demand required for perception dramatically. Incremental system building is facilitated as new perceptual strategies can be added easily as they are required.

The interactions between these algorithms as well as the invocation and termination of specific schemes can be managed by a finite state controller [1]. This partitioning of the control from the processing is one more step in constructing more task-oriented algorithms. The search for methods to decompose the problem and thus make it tractable through the application of biological and psychological principles is a good characterization of our approach to robotics.

3. Perceptual Algorithms

Within AuRA, perceptual algorithms are developed to fulfil a specific behavioral requirement. For instance, it is important that a mobile vehicle can recognize and avoid moving objects that may pose a collision threat. A perceptual algorithm is then developed specifically to perform this task. It would be responsible for enabling the appropriate sensors and processing the sensor input data to locate such objects. For each threat found, a motor control schema would be instantiated to monitor its trajectory and generate appropriate velocity vectors to guide the vehicle away from potential collisions. The perceptual algorithms therefore are implementations of specific behavioral requirements.

The ballistic component of motion represents the initial rapid movement of the vehicle through the environment to a position near the intended destination where the target workstation is recognizable. To perform this transport, ballistic motion requires the support of long range perceptual schemas for guidance and rapid obstacle avoidance schemas to allow safe passage. The guidance schemas must extract salient features from the world that will allow the vehicle to be directed toward the destination. The obstacle avoidance schemas must watch for objects that the vehicle may encounter and then guide the vehicle away from these potential collisions.

A significant portion of the complexity of the guidance algorithms is devoted to handling failures. Each perceptual schema must be capable of quantifying its performance so that it can notify the controller if it is no longer useful. If, for example, a schema that is tracking a bright light no longer detects the light then it must report this failure so that some other tracking method can be deployed.

3.1 Long-range Ballistic Cues

Long-range perceptual algorithms are used to attract the robot to the general area of the target destination. These algorithms are not expected to be accurate and will fail under certain conditions. The algorithms must be capable of detecting when they have failed and report that to the planner/controller.

Each of the long-range ballistic cueing algorithms processes video images in a manner consitent with by its perceptual requirements. The algorithms presented here require little precision or resolution. These schemas support a servo control mechanism that drives the feature being tracked to the horizontal center of the image as the robot moves. A calibrated optic system thus is not required. A rough approximation of the camera field of view is used to decide when the feature should be visible as well as to determine vehicle rotations necessary to servo the feature to the center of the image. The system currently uses 256x256 pixel images with 256 levels of gray scale.

3.2 Phototropic Perception

The phototropic (light seeking) perceptual algorithm is used to attract the robot to areas of high light intensity. In [12] it is suggested that sea turtles find their way to the ocean after hatching using a phototropic behavior. In [11] a robot is presented that navigates using a photovoric behavior. In a FMS environment, the bright light source used for navigation would arise from naturally occurring workstation lights near the intended destination or by using artificial beacons. The phototropic perceptual strategy has been chosen because it is simple, fast and robust over a relatively large range of environmental conditions. The environmental constraint requires that the desired target light must be the brightest light in the field of view of the camera. This condition is reasonably easy to meet.

3.2.1 Phototropic Perception Algorithms

The camera mounted on our Denning robot is constrained to track with the wheels. This lack of independent camera control causes some difficulty since the light may pass out of view as the robot maneuvers. When tracking of the light is thus interrupted, it is necessary for the phototropic algorithm to reacquire the target light. This reacquisition uses the camera field of view along with heading information to determine when the light should be visible. If the camera is pointing towards where the light was last seen, the image is examined for a light with an intensity similar (including a small drift factor) to the previously tracked light. An allowable drift factor of 10% works well in our applications when the lights are much brighter than the background and widely spaced geographically.

It would be possible to implement more robust matching of the target light from image to image, but it is preferable to retain a simple and computationally inexpensive implementation. Tracking the wrong light when it is in close proximity to the goal is not a serious problem. The vehicle still ends up generally near the desired destination, which satisfies the basic requirement for ballistic motion.

Figure 1 presents pseudocode for the algorithm used to calculate a velocity vector that moves the robot toward the light in feedforward mode. The bootstrapping process is not detailed, but consists merely of directing the camera at the target light and initializing the variables used to remember its initial intensity and direction. Figure 2 presents in pseudocode the algorithm used to scan the image plane and locate the brightest light. This is invoked by the main algorithm when the target light is predicted to be visible.

3.2.2 Experimental Data

The example run using the phototropic perceptual algorithms is shown in Figures 3-5. Figure 3 is the camera view after initially acquiring the light to be tracked. The cross shown in the image indicates the location that the algorithm is tracking towards. In Figure 4, the robot has moved around an obstacle and lost sight of the light during the transit. It has just reacquired the light again and has resumed tracking. The final image shown in Figure 5 presents the view at the termination of the ballistic motion phase of the robots movements.

3.3 Activity Detection

The activity detection (motion seeking) algorithm attracts the robot to areas of high motion [13]. The motion is expected to be produced by pick and place robots operating near the intended destination. This

```
Input: Image plane
Output: Velocity vector
BEGIN
    /* Check if light should be visible */
    IF (abs(robot_heading - last_heading) < lens_fov) THEN
        /* Light should be visible, invoke the algorithm in Figure 2 */
        findlight(image, intensity, light_loc)
        /* Check if the target light was detected */
        IF (intensity >= (last_intensity - drift) AND (intensity > min)) THEN
            /* Light is visible, so update internal representation */
            last_heading = robot_heading + light_loc
            last_intensity = last_intensity + (intensity - last_intensity)/2
        ENDIF
    ENDIF
    vector_direction = last_heading
    /* The base_gain_level value sets the maximum vehicle velocity */
    /* The cur_light_gain value sets the contribution of this schema */
    vector_magnitude = base_gain_level * cur_light_gain
END
```

Figure 1: Phototropic Perceptual Schema

```
Input: Image plane
Output: Row and column of light in the image
Note: For efficiency, only every other row and column are scanned
BEGIN
    max = 0
    FOR row = 2 TO max_row - 3 BY 2 DO
        FOR col = 2 TO max_col - 3 BY 2 DO
            IF (pixel(row, col) > max) THEN
                /* Check that not random noise */
                IF (pixel(row, col - 2) > max AND
                    pixel(row, col + 2) > max AND
                    pixel(row - 2, col) > max AND
                    pixel(row + 2, col) > max) THEN
                    /* remember the new maximum */
                    max = pixel(row, col)
                    loc_row = row
                    loc_col = col
                ENDIF
            ENDIF
        ENDDO
    ENDDO
END
```

Figure 2: Computation of location of light in image plane

typically occurs when the vehicle is delivering cargo to be processed by one of these robots. Tracking is initiated by pointing the camera at the intended destination. Several frames are then sampled and compared to determine areas of high activity. Related research can be found in [8, 15, 16].

3.3.1 Motivation

One characteristic of a FMS environment is motion. Product assembly areas with robots and conveyors are constantly moving. This motion is regular since the equipment is typically repeating the same movements over and over. This constant motion can be exploited to provide perceptual cues using simple

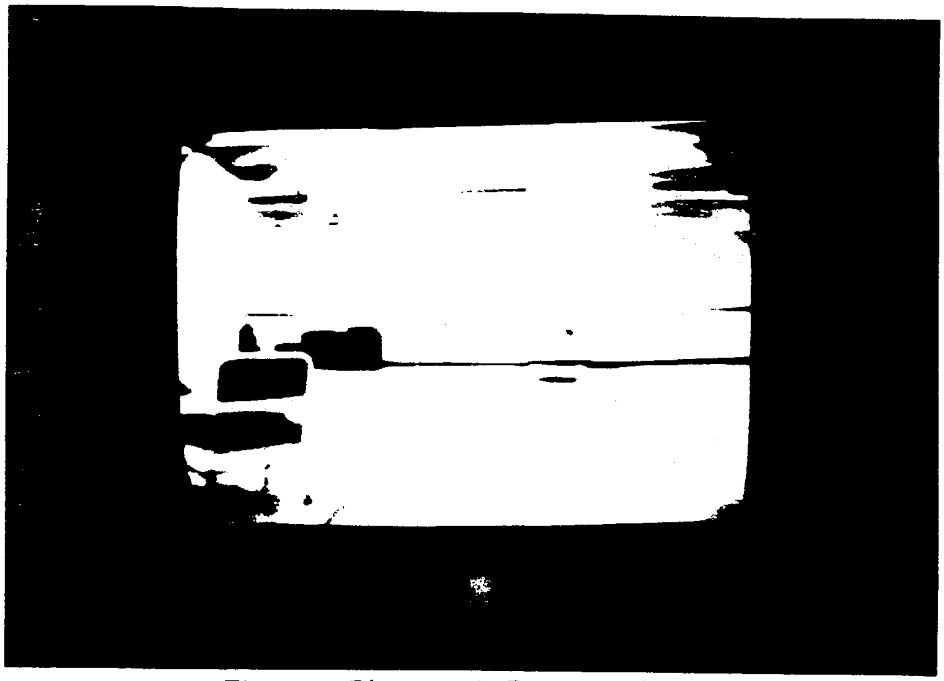

Figure 3: Phototropic Detection Image

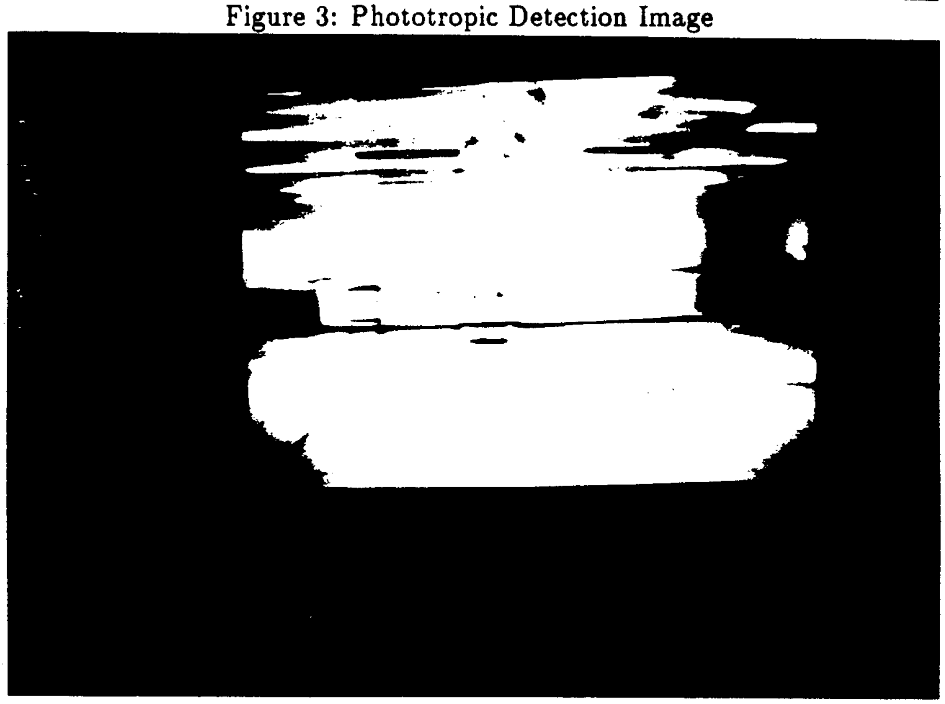

Figure 4: Phototropic Detection Image

vision processing algorithms to aid the ballistic phase of motion [5, 7].

Motion detection algorithms must deal with many problems, the most critical being the matching problem and visual noise. It has been suggested that expectation driven methods coupled with environmental knowledge can overcome these problems [9]. Within a FMS, knowledge of the motion characteristics and limits of machines near the destination dock coupled with positional models of the stationary features can support simple motion detection algorithms. These algorithms are then used to aid the ballistic motion

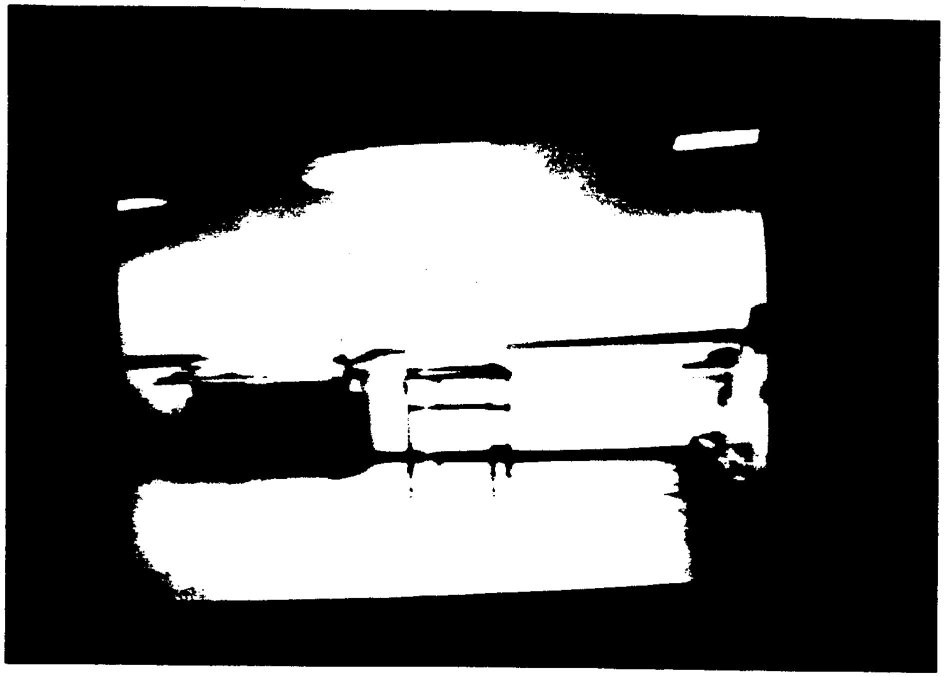

Figure 5: Phototropic Detection Image

phase of travel to the workstation.

3.3.2 Activity Detection Algorithms

Two different methods are currently available for extracting the location of motion sources from images. The first algorithm simply accumulates the absolute difference in intensities of the image planes. The input is a sequence of images I_1 to I_n. Image I_j is compared with image I_{j+1} and the absolute value of the intensity variations or each pixel is accumulated in the difference plane. This algorithm appears in Figure 6.

Input: Image sequence set, S = $(I_1, ..., I_n)$.
Output: Cumulative Absolute Difference Plane.
BEGIN
    Initialize accumulation plane $(pixel_{cumulative})$ *to* 0*/
    FOR *each sequential image pair of the set*$(i = 1, ..., I_n)$ DO
        FOR every pixel DO
            /* Calculate absolute difference of image pair $(i, i+1)$ */
            $pixel_{diff}(row, col) = abs(pixel_i(row, col) - pixel_{i+1}(row, col))$
            /* Accumulate normalized total activity */
            $pixel_{cumulative}(row, col) = [pixel_{cumulative}(row, col) * (i-1)/i] + [pixel_{diff}(row, col) * (1/i)]$
        ENDDO
    ENDDO
END

Figure 6: Computation of Cumulative Absolute Difference Plane

Using knowledge of the ratio of the number of pixels that should be in motion versus the image size, the difference plane can be processed into a binary motion plane. This processing is necessary to reduce the effects of noise on the results. Changes in contrast caused by lighting variations as well as the vehicles motion through the environment will introduce a random motion component into the difference plane.

An estimate of the percentage of active pixels in the image that can be attributed to the motion of the feature being tracked can be expressed as threshold t_1. The difference plane is scanned beginning with the pixels exhibiting the highest activity. Pixels in the binary motion plane are set, marking each of these high activity pixels, until the ratio of the active to inactive pixels exceeds t_1. This algorithm appears in Figure 7.

```
Input: Accumulation Plane and threshold t_1, the active pixel ratio
Output: Binary Motion Plane
BEGIN
    Initialize binary motion plane (pixel_binary) to 0
    total_pixels = max_row * max_column
    marked_pixels = 0
    location_activity = n - 1
    WHILE (marked_pixels/total_pixels < t_1) DO
        /* Mark pixels which equal location_activity */
        FOR every pixel DO
            IF (pixel_cumulative = location_activity) THEN
                pixel_binary(row, col) = 1
                marked_pixels = marked_pixels + 1
            ENDIF
        ENDDO
        location_activity = location_activity - 1
    ENDDO
END
```

Figure 7: Creation of Binary Motion Plane

A variation of this algorithm has been constructed using a threshold to remove noise at the accumulation stage. The input is a sequence of images I_1 to I_n of a scene believed to contain the workstation and the activity threshold t_2. Image I_j is compared to image I_{j+1} pixel by pixel. If the absolute value of the difference in intensities between the two images exceeds the threshold t_2, then the bucket for that pixel in the accumulation plane is incremented. The result is a scene motion accumulation plane. This algorithm is shown in Figure 8.

```
Input: Image sequence set, S=(I_1, ..., I_n) and t_2 the minimum activity threshold.
Output: Scene Motion Accumulation Plane
BEGIN
    initialize accumulation plane (pixel_accumulate) to 0
    FOR each sequential image pair of the set(i = 1, ..., n - 1) DO
        /* calculate absolute difference of image pair (i,i+1) */
        pixel_diff(row, col) = abs(pixel_i(row, col) - pixel_i+1(row, col))
        /* Threshold the pixel value */
        IF pixel_diff(row, col) > t_2 THEN
            pixel_accumulate(row, col) = pixel_accumulate(row, col) + 1
        ENDIF
    ENDDO
END
```

Figure 8: Computation of Scene Motion Accumulation Plane

3.3.3 Activity Detection Experimental Results

Figure 9 shows a scene of a conveyor with totes moving along it with a flashing warning light in the upper left corner of the image. A sequence of 4 such images taken at different times were used to construct the scene motion accumulation plane shown in Figure 10. A threshold was then applied to this plane using the algorithm in Figure 7 that retained only the pixels representing the top 2% activity level to produce the binary motion plane shown in Figure 11. These high activity regions in the image can then be used to guide the robot to its desired destination.

Figure 12 shows the activity detection algorithm being used on an image with an assembly robot and a conveyor system. A sequence of 14 images taken at different times were used to construct the scene motion accumulation plane shown in Figure 13. Figure 14 shows the resultant binary motion plane.

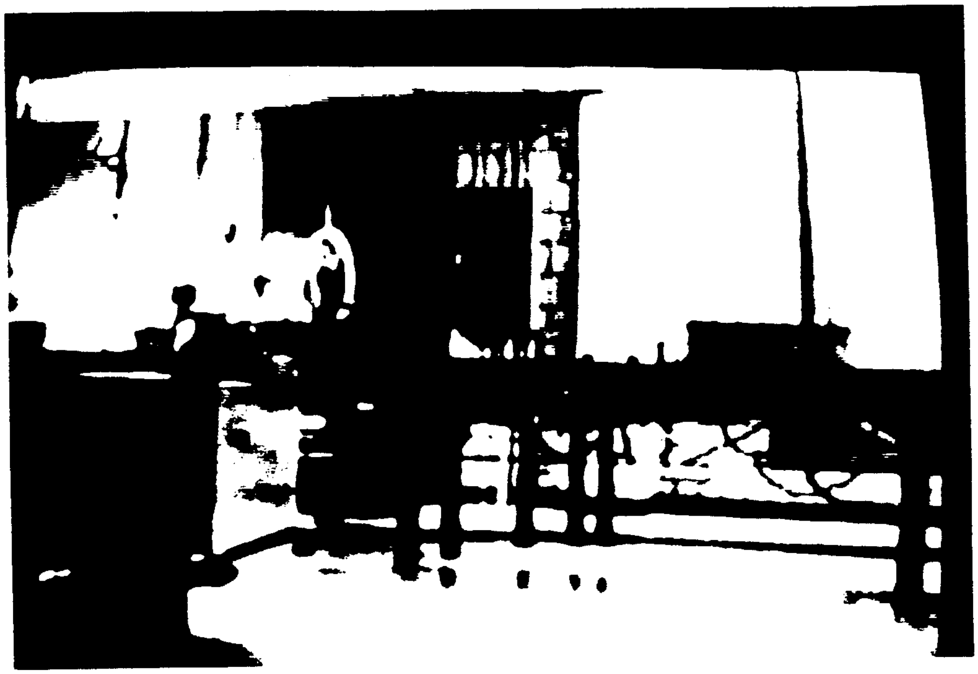

Figure 9: One image from sequence used for activity detection for localization

4. Summary and Conclusions

The reactive control paradigm used within AuRA encourages the development of a library of perceptual schemas, each optimized to support a single goal. The two perceptual algorithms presented here are useful additions to this library of perceptual algorithms. This incremental system building decomposes the perceptual task into manageable portions thereby allowing significant progress to be made in constructing a mobile robot capable of functioning similar to an AGV in a FMS environment.

The algorithms presented have been shown to be useful in manufacturing environments but they also could find applicability in other domains. Their low computational complexity and minimal environmental constraints makes them enticing candidates for supporting ballistic motion.

5. Acknowledgements

This research is supported in part by the CIMS program and Materials Handling Research Center of Georgia Tech. The authors would like to thank Robin Murphy, David Vaughn and Mark Pearson for their role in developing computer vision algorithms used within this research.

Figure 10: Resulting motion accumulation plane

Figure 11: Resulting binary motion plane

Figure 12: One image from sequence used for activity detection

Figure 13: Resulting motion accumulation plane

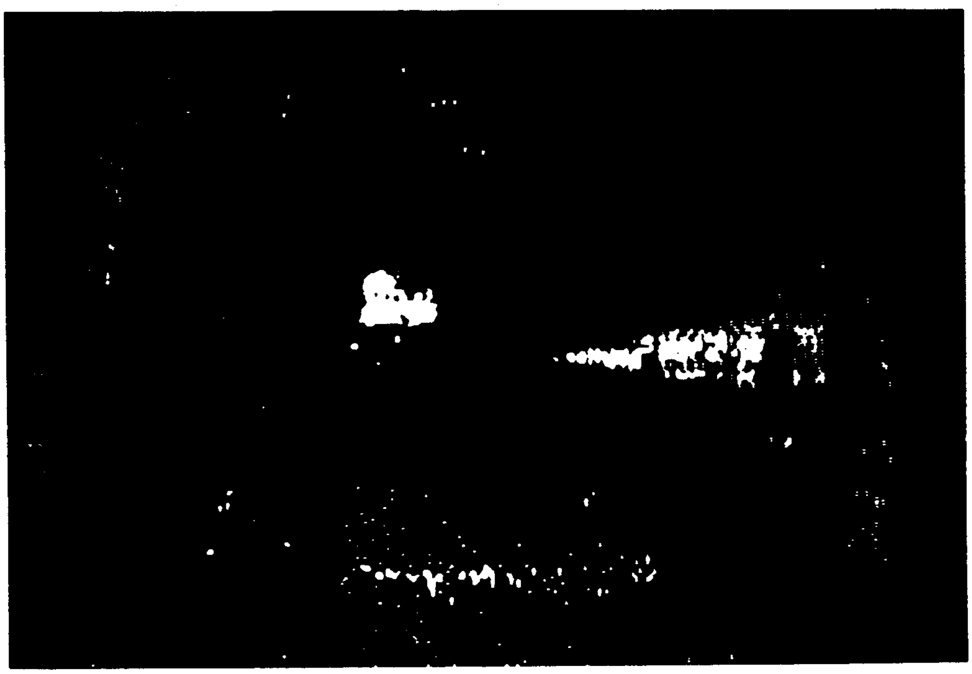

Figure 14: Resulting binary motion plane

MacKenzie

6. References

1. Arkin, R.C., MacKenzie, D.C., "Temporal Coordination of Perceptual Algorithms for Mobile Robot Navigation", Submitted to *IEEE International Conference on Robotics and Automation,* Nice, France, May 1992.

2. Arkin, R.C., "The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study", *IEEE Transactions on Systems, Man and Cybernetics,* Vol 20, No. 6, Nov/Dec 1990, pp. 1245-1257.

3. Arkin, R.C. and Murphy, R.R., "Autonomous Navigation in a Manufacturing Environment", *IEEE Transactions on Robotics and Automation,* Vol. 6, No. 4, August 1990, pp. 445-454.

4. Arkin, R.C., "Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation", *Robotics and Autonomous Systems,* 6 (1990) pp. 105-122.

5. Arkin, R.C., Murphy, R.R., Pearson, M. and Vaughn, D., "Mobile Robot Docking Operations in a Manufacturing Environment: Progress in Visual Perceptual Strategies", *Proc. IEEE International Workshop on Intelligent Robots and Systems '89,* Tsukuba, Japan, pp. 147-154, 1989.

6. Arkin, R.C., "Towards Cosmopolitan Robots: Intelligent Navigation of a Mobile Robot in Extended Man-made Environments", *Ph.D. Dissertation, COINS Technical Report 87-80,* Department of Computer and Information Science, University of Massachusetts, 1987.

7. Ballard, D.H., Brown, C.M. *Computer Vision,* Prentice-Hall, 1982, pp. 195-199.

8. Bhanu, B., Symosek, P., Ming, J., Burger, W., Nasr, H., Kim, J. "Qualitative Target Motion Detection and Tracking," *Proc. DARPA Image Understanding Workshop,* May 1989, pp. 370-398.

9. Bhanu, B., Roberts B., and Ming, J. "Inertial Navigation Sensor Integrated Motion Analysis," *Proc. DARPA Image Understanding Workshop,* May 1989, pp.747-763.

10. Brooks, V.V., Cooke, J.D., and Thomas, J.S. "The Continuity of Movements," *Control of Posture and Locomotion,* edited by R.B. Stein, et al, Plenum Press, pp. 257, 1973.

11. Connell, J., "The Omni photovore; how to build a robot that thinks like a roach", *Omni,* pp. 201-212, October 1988.

12. Hayes, W.N., Ireland, L.C. "Visually Guided Behavior of Turtles", *The Behavior of Fish and other Aquatic Animals,* edited by D.I. Mostofsky, Academic Press, pp. 300-302, 1978.

13. Horswill, I.D., Brooks, R.A. "Situated vision in a Dynamic World: Chasing Objects", *Proc. AAAI 88, Seventh National Conference on Artificial Intelligence,* vol. 2, 1988, pp. 796-800.

14. Howarth, C.I., Begg, W.D.A., and Bowden, J.M., "The Relationship between Speed and Accuracy Aimed at a Target," *ACTA Psychological Review,* vol. 35, 1971, pp. 207-218.

15. Liou, S.P., Jain, R.C., "Motion Detection in Spatio-Temporal Space," *Proc. IEEE Computer Society Workshop on Computer Vision,* November 1987, pp. 352-354.

16. Pavlin, I., "Motion From a Sequence of Images," *Proc. DARPA Image Understanding Workshop,* April 1988, pp. 930-937.