Workstation Recognition using a Constrained Edge Based Hough Transform for Mobile Robot Navigation

David L. Vaughn and Ronald C. Arkin
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280

Abstract

Landmark recognition is a task required of many robotic systems. In this work, we examine the use of a constrained Hough transform used by a mobile robot to locate a docking workstation. This algorithm deals with the uncertainty inherent in a mobile robot by making use of a spatial uncertainty map maintained by the robot. Several iterations of the Hough transform are run with transformed models of the dock. Votes are accumulated in a collapsed Hough space which, although unable to recover range and orientation information, simplifies locating the dock within the image.

1 Introduction

The goal for this research has been to create a system capable of finding a docking station within an image taken by a mobile robot which is operating in a manufacturing environment. This is a common example of a class of problems faced by mobile robots. In this work, the recognition is accomplished by using a constrained Hough transform algorithm (Htrans) with edge points as the salient feature.

It is intended that the robot calls the algorithm after it has maneuvered itself into a position where the dock is in view and is between 10 and 20 feet distant (it has been decided to rely both on ultrasonic sensing and other visual techniques for the final close range maneuvering). Htrans then determines the position of the dock within the image. This information is then used to further guide the robot towards the dock. See [4] for more information on the overall docking strategy for the robot.

2 Previous Work

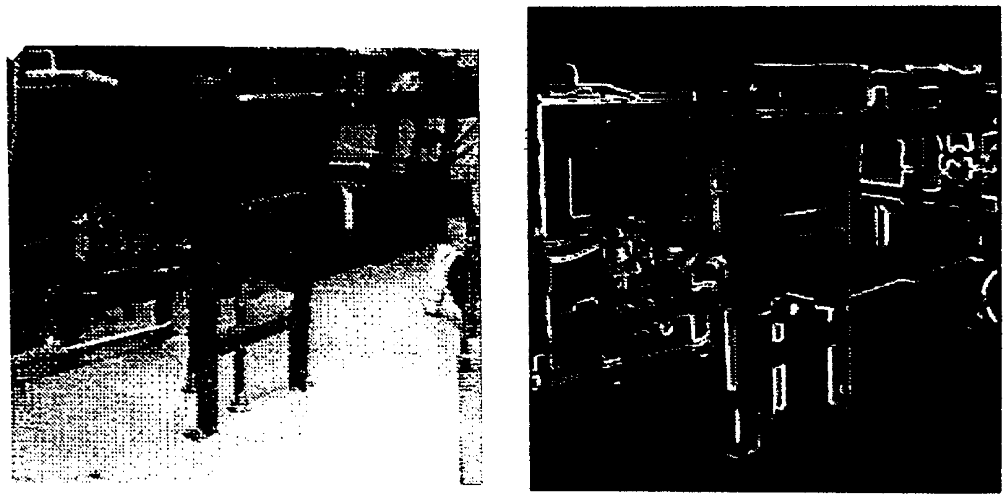
The Hough transform is a general algorithm for the recognition of arbitrary shapes within an image [8]. This approach has been used widely for the location of such landmarks as navigation lines [6] and more complex objects [1,7,9]. The Hough transform has proven to be successful in locating a wide number of objects in a variety of domains.

A number of variations on the basic Hough transform have been made to either tailor the algorithm towards a specific domain or application, or to improve its performance. Examples of application-tailored Hough transforms would be those used to find geometric shapes such as lines or circles. In these cases, it is possible to replace the usual object model with an equation. Common performance modifications include the use of features other than object border points (edge points) or to place restrictions on the features allowed to vote. The goals of the last two methods are to reduce the number of votes which must be cast so as to effect a speed-up of the algorithm and to restrict the votes cast to being only the most informative possible.

3 Operating Environment

The manufacturing environment chosen is the Materials and Handling Research Center Laboratory (MHRC) located at Georgia Tech. This environment contains a number of conveyor belts, stationary robots, work benches, docks (loading sites), etc. The lights are unevenly spaced, the walls contain strong horizontal and vertical edges and are not all painted in the same color, and there are occasional navigation lines along the floor used by resident autonomous guided vehicles (AGVs). This all makes for a highly cluttered environment. See Figure 1a for an example image of a dock. Note that there are a number of structures in this image (i.e., the conveyor belt and table) which vaguely resemble docks at different ranges and orientations.

The images taken have relatively little noise and no additional preprocessing was required other than to reduce them from 512x512 to 256x256. This was done solely to reduce computation. Although the images presented here



(a) (b)

Figure 1: Images generated for run of Htrans at position (6,7)
(a) Example image of a dock taken from 7 feet in front and 6 feet to the right of the dock.
(b) Sobel magnitude image resulting from (a).

were taken using a video camera mounted on a tripod, and not our actual robot, it is believed that this will not pose a problem when this system is actually utilized by our mobile robot, George (a Denning vehicle). The only significant difference a robot would present (other than some camera variations) would be the possibility of some minor camera roll.

4 Edge-based Hough Transform

It has been decided to use edge points as the basis for the Hough transform used in the first stage of this work. The model is described in terms of the complete set of the object's edge points. Because of the large number of points within the model and image, the algorithm runs slower when using edges than if some other feature (e.g., corners or lines) were used. On the other hand, an edge point Hough transform is more tolerant of variations (including partial occlusion) of the object within the image.

In its basic form, the Hough transform only recognizes those objects within the image corresponding to the model as given. It is possible, however, to take into account parameters such as scaling and rotation of the model by transforming the model and re-running the algorithm, placing the new votes in additional dimensions of the Hough space. If the correct scale and orientation of the object are completely unknown, then a large number of Hough transforms needs to be run in order to guarantee an accurate location of the object. If, however, some constraints can be placed on the object's location, as is done in this research, then it is possible to greatly reduce the number of applications of the transforms needed. In this paper, we examine a method to locate an object which requires the application of only five Hough transforms, given certain constraints on the object's position.

4.1 Assumptions

It has been assumed that the robot will only call Htrans when a portion the dock's face is in the field of view. If for some reason the dock were not in view, Htrans would return a location for the dock with a relatively low weight which could be recognized by a higher level reasoning mechanism.

The robot passes to the algorithm an estimate of its location relative to the dock (Fig. 2). This location is accurate to within plus or minus 4 feet of the dock's true location, creating a circle of uncertainty with a diameter of

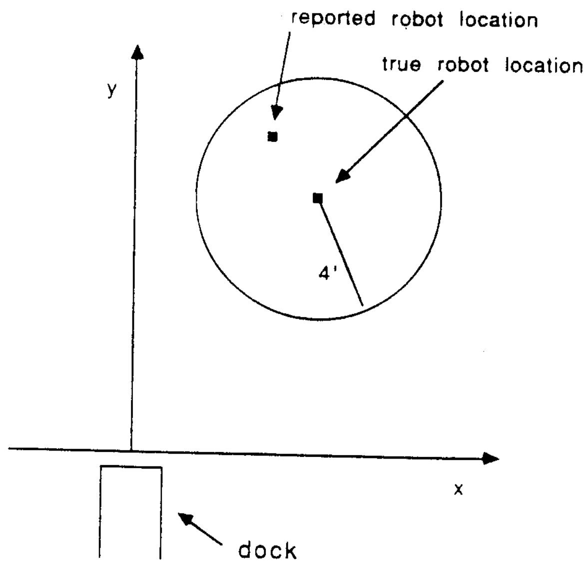


Figure 2: Example positioning of robot relative to dock. Note the reported robot location is within the 4 foot radius circle of uncertainty centered about the true robot location.

8 feet about the location passed to the Hough. Note the coordinate system used in Figure 2: the dock is considered to be at the origin with the positive y-axis going away from the dock and the positive x-axis to the right. This system will be used throughout the remainder of this paper.

The robot's motion is constrained to three degrees of freedom (two translation and one rotation). It is assumed that the floor is relatively level and that the robot has no significant tilt. It is also assumed that the dock is stationary.

Although constant lighting is not assumed, it is necessary to have sufficient illumination to distinguish objects. Shadows can be tolerated as long as it is still possible to extract strong edge points from within the region.

4.2 Spatial Uncertainty

The Autonomous Robot Architecture [2] maintains not only the robot's most likely position and orientation, but also measures of the uncertainty of this information. This information is represented by two structures: a convex polygon for the positional uncertainty, and a wedge for the orientation range (Fig. 3). Space prevents a thorough discussion of the details of uncertainty management, so the reader is referred to [3] for more information. What is important to remember is that the explicit representation of uncertainty affords constraints on the Hough Transform by providing bounds on the positional and orientational uncertainty of the vehicle relative to the docking workstation.

4.3 The Htrans Algorithm

The Htrans algorithm has four steps: edge point extraction, model transformation, vote casting, and maxima location. Edge point extraction is performed to recover low-level features from the image. Model transformation is based on the reported location of the robot relative to the dock and is intended to bring the appearance of the model into correlation with the possible object appearance within the image. Next, a series of Hough transforms are run using the transformed model and extracted edge points. Finally, the location of the object within the image is determined during maxima location. As discussed later in this paper, the model transformation and vote casting steps are repeated five times each before maxima location.

4.3.1 Model Representation

There exist a variety of methods for the internal representation of 3D models which allow a transformation into two dimensions. The method chosen for this research utilizes a variant of the aspect graph.

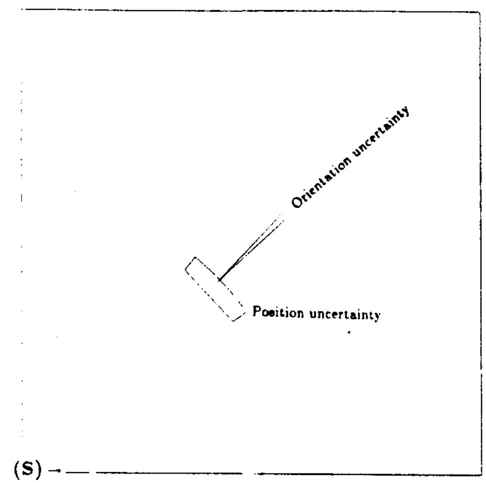


Figure 3: Spatial uncertainty map.
The convex region represents the limits of positional uncertainty while the wedge represents the bounds on orientational uncertainty [3].

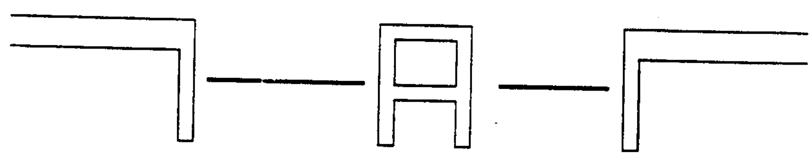


Figure 4: Orthographic projection aspect graph for dock object.

An aspect graph for an object is a graph where each node is a "topologically distinct stable view of the object and the cell viewing space from which that view is seen" [10]. The arcs within this graph connect a stable view with all other stable views which may be reached immediately. For the purposes of this work, it has been decided to use an orthographic projection aspect graph [10,11] using only the front and side faces of the dock. Figure 4 shows such a graph for a dock. Note that the back, top and bottom of the dock are not represented. The robot will never be able to see the top or bottom and we are guaranteed by restrictions on the problem that the robot will not be located behind the dock.

The advantage of the aspect graph lies in the ease with which it may be transformed. Since each node within the graph is a stable view, any transformations of the model within the constraints of a given node can be accomplished without having to determine which of the object's faces will be visible and which will be obscured. This model compares favorably to a 3D wire frame where, in order to translate into two dimensions, it must be determined which parts will be visible.

The 3D models used in this work were generated manually after examining the output of a Sobel operator on a series of three images taken of a dock from 10 feet away (one image for each face). These models are somewhat generalized in that only the the most salient features of the dock are included (not each rivet, for example). A separate model is used for each docking station in the MHRC, although a number of the docks are sufficiently similar that a single model could have been used to represent the group. Each dock is described with respect to the top center of its front face.

The three faces for the dock model are stored separately so as to allow independent manipulation. The points for the front face are stored in terms of their x and y offsets from the top center point. The points for the side faces are stored in terms of their x and y offsets from the top corner adjacent to the front face. All model transformations are performed using the original model to avoid error propagation. The model is initially described according to the dock's appearance from a distance of 10 feet.

To transform a model when two faces are visible, each face is transformed independently and the two are then joined. This joining is accomplished by removing from the side face the vertical line adjacent to the front face (this edge is not visible in the images) and then replacing each point in the side face with that point's x and y offset from the top center of the front face.

The rear legs of the dock are not included in the model since these rear legs are generally not visible within the image. For this reason, it is never necessary to use more than two of the model's faces at a time.

Based on the robot's presumed location, the model is transformed to match the dock's predicted appearance based on a pinhole camera model. Assume, for example, that the robot reports its position as (10,7) (this is 10 feet forward and 7 feet to the left from the dock's point of view). This location is 12.2 feet distant from the dock and at an orientation of 55 degrees. The model will therefore be scaled by a factor of 0.82 (10 / 12.2 = 0.82), since the model is based on the dock's appearance at 10 feet, and rotated accordingly.

The model points are grouped in 45 degree increments labeled 1 through 4 (1 being vertical, 3 horizontal, and 2 and 4 diagonal) based on the orientation of the line along which they lie. Such wide categories are chosen since the accuracy of edge direction information extracted from these images is not particularly accurate (a small amount of noise can result in large direction variations) and, since this work is within a man-made environment, nearly all the strong edges fall into the horizontal or vertical categories. Since it is not possible to guarantee a uniform background, it has been decided to combine all vertical edges into one category (similarly for horizontal and diagonal model points). For example, all vertical points (up and down) are grouped together since the difference between an 'up' edge point and a 'down' edge point is dependent upon whether the pixel's contrast is greater or lower than its neighbor's. If the background were brighter than a part of the dock on the left side and darker on the right side then the edge points on the opposing sides would register as being at the same orientation even though opposites would normally be expected. Although grouping these edge points makes running Htrans somewhat more computationally expensive (each edge point extracted from the image now casts votes for its true direction and its opposite), it does yield better results than not doing so.

4.3.2 Edge Point Extraction

Edge point extraction is performed by applying a Sobel operator with a 3x3 filter to the input image. This technique extracts edge points along with their associated magnitudes and directions. Directions are grouped in 45 degree increments just as with the model points. All edge points with magnitudes below the average edge strength are discarded as are edge points along diagonals. There are very few diagonal edges within the model, even when viewed at a 45 degree angle, since the camera is at the same height as the top of the dock and because of the long ranges at which the dock is viewed. Weak edges are removed to reduce the effect of noise and to reduce computation. The edge point array for the image in Figure 1a is displayed in Figure 1b.

After the Sobel has been run, the only edge points remaining consists of points corresponding to areas of high contrast within the image. Each edge produced is typically several points thick, making the magnitude array appear 'fuzzy' when displayed. If a larger filter (5x5, for example) were used then the edges would be even thicker, making the Hough more tolerant of slight variations in object appearance. However, a larger filter would also take substantially longer to process, and produce far more points which would result in the casting of more votes.

In some cases, it is possible to further prune the edge points by eliminating the less informative points [1] from the image. Informative edge points have an orientation (or other attribute) which is common in the model but rare in the image. It has been found that, for the images used in this work, the dock object is not significantly statistically different from the rest of the image in terms of edge orientation or magnitude (the only two attributes considered). In other words, compared to the rest of the image, the dock has a typical intensity and has a typical distribution of horizontal, vertical, and diagonal edge points.

4.3.3 Voting

Each edge point casts one vote for each model point with a corresponding direction. Each vote is placed within a compressed Hough space (Hspace) at the location corresponding to what the center of the dock's front face would

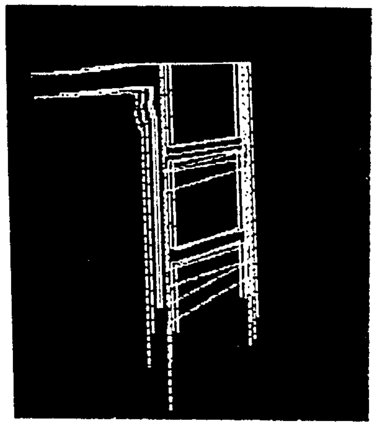


Figure 5: The cluster transformations for the model centered about point (6,7) and overlapped on top of one another.

be if the edge point were the point in the model currently under consideration. The Hspace is represented as a two dimensional array of the same size as the input image.

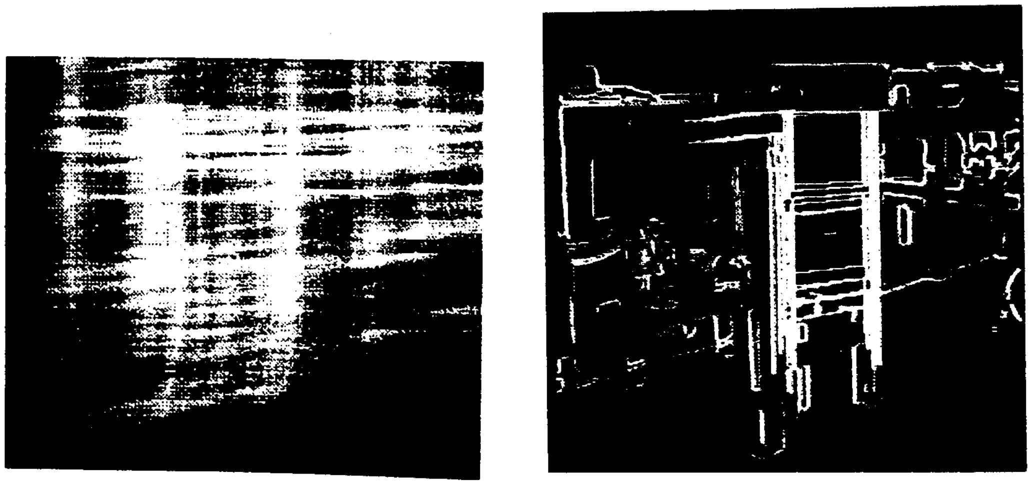
If the robot were passing in the correct coordinates for its location (but not necessarily orientation) then this is all we would need to do. We would be reasonably guaranteed that our transformed model corresponded to the appearance of the dock within the image and, assuming a sufficiently unobscured dock appeared within the image, the Htrans would succeed. However, as has been previously mentioned, the robot may be off in its position estimate by several feet, thus eliminating any guarantee of success.

To cope with this uncertainty, a series of five Hough transforms are run. The initial Htrans is run using the location for the robot which was originally passed in. The other four are run using locations one foot distant from the initial location (either one foot left, right, forward, or back). See Figure 2 for the layout of this clustering. This group of Hough transforms is referred to as the 'cluster' for the remainder of the paper. Figure 5 shows the model transformed for each position within the cluster centered about position (6,7) [1] and overlapped on top of one another. Note that although some parts of the dock vary substantially with each transformation, the top edge and the two legs remain fairly constant. Empirical examinations have shown a spacing of one foot between model transformations to be best.

It is common when running multiple Hough transforms to place the results of each in a different dimension of the Hough space - creating a multi-dimensional Hough space [5]. It is then often possible to scan this Hough space for the strongest peak and recover not only the location of the object in the image but also its range and orientation. To do this, however, it is necessary to have run the Htrans using a model transformed to a very close approximation of the object within the image. For our purposes, there is a fairly high degree of uncertainty as to the object's appearance within the image due to robot uncertainty in positioning error. In order to guarantee such a close approximation, it would be necessary to run an extensive number of Hough transforms taking far more time and resources (memory for the Hspaces) than practical.

To overcome this problem, only one collapsed Hspace is used, with each Htrans accumulating its votes within the same space. Although this makes it impossible to recover depth and orientation information, it also requires fewer runs of the Hough. Even when none of the transformations are made from the actual location of the robot, the Htrans will still succeed as long as there are sufficient runs using model transformations which are reasonably close to the correct transformation. This is because each run, even those which are far removed from the correct position of the vehicle, votes for the correct location with some portion of the model. Enough sufficiently varied runs

[1] This notation, used henceforth in this paper, refers to the following: the first number indicates the number of feet in front of the dock, and the second number indicates the number of feet to the right of the center of the dock (from the dock's point of view). A negative number for the second value indicates the number of feet to the left.



(a) (b)

Figure 6: Model Transformations
(a) The Hspace generated from Figure 1a.
(b) The set of models overlapped on top of the Sobel magnitude image at the maxima located within the Hspace.

increases the likelihood that each part of the dock is be represented.

Since each transformation of the model contains a different number of points, some transformations are more heavily weighted than others. To compensate for this, normalizing the vote weights so that each model transformation is considered evenly was examined. It was found that the Htrans was only slightly more accurate in certain cases (generally providing results not more than a few pixels closer then otherwise obtained) making model weighting only marginally beneficial.

The location in the Hspace corresponding to the dock's location grows fastest when running those Htrans' in the cluster which most closely approximate the robot's true location. Since the cluster is uniformly distributed, using a larger cluster only helps if the Htrans' most distant from the robot's correct location do not vote strongly for any false docks. It has been observed that strong false docks are most likely in this environment when running Htrans' which are greater than 5 feet away from the true location. A cluster radius of 1 foot results in one or more of the Htrans' begin run at a greater distance than 5 feet from the true location when the position error is more than 4 feet. A cluster radius of 3 feet results in Htrans' being run at distances greater than 5 feet from the true location when the position error is more than 2 feet. Therefore, it is generally not helpful to increase the cluster radius beyond 1 or 2 feet due to the increased probability of running a Htrans at a distance of greater than 5 feet from the true position of the robot.

4.3.4 Maxima Location

After the Htrans has been completely run, it is possible to recover the position of the dock within the image by scanning the collapsed Hspace for the largest value. This returns the location which was most frequently voted as being the position within the image of the dock.

Figure 6a is an example Hspace. This Hspace was generated using the image from Figure 1a (taken from location (6,7)) as input and running a cluster with the initial coordinates (8,8). Note that the brightest peak within this space corresponds to the center of the dock's front face. Figure 6b shows the overlapped transformed models placed over the edge point image at the location of the peak in the Hspace. This is a correct match of the dock's location within the image.

5 Experiments

Figures 9, 10, and 11 show the results of running clusters offset from the correct robot location at 0-4 feet on four different images taken of the same dock at varying locations. The images and results are presented in Figures 1, 7, and 8. Note that the same dock model is used in all cases. The percentage error listed is the offset of the dock recognition location from the dock's actual position within the image expressed as a percentage of the dock's width as it appears in the image. In each case, as expected, the performance degrades as the robot location error increases. Thus an error greater than 100% indicates a complete miss. Anything less means that the recovered location lies somewhere along the front of the dock. As long as the coordinates given for the robot are within 3 feet of its true location, the algorithm is highly successful at locating the correct position of the dock within the image. At 4 feet, the dock is still located but the match is not as accurate. At greater distances, the results are less predictable.

Although there were no actual failures to detect the workstation, there are a number of runs which come close to failing when the robot's presumed location is offset by 4 feet from its true location (see Fig. 9). It is not unexpected that the results presented in Figure 11 appear better than those obtained from the other images since an error of four feet is more visible in an image taken from 9 feet away (for example) than in an image taken from 18 feet.

5.1 Computational Cost

The Htrans algorithm takes approximately 10 minutes to run the cluster of Houghs on a MicroVAX II. Although this is nowhere near real time, it is not surprising. Consider that the typical image has on the order of 10000 edge points (after thresholding) and the model has about 700 edge points. This means that approximately 7 million votes are being cast for each run of the Hough and 35 million votes for the entire cluster. Certainly this algorithm could be made significantly faster on a parallel machine. Using lines or corners for features, instead of edges, would take substantially less time as the model would then be composed of only a few dozen features with only several hundred lines or corners being extracted from the image (resulting in only several thousand votes).

5.2 Analysis

The problem with relatively distant images is that, in the images used to test this work, the edge contrast of the dock begins to fade since the resolution of the images used is not very high (256x256) and the dock has little significant difference in contrast from the rest of the image. Distant images have the advantage of being relatively tolerant of robot position error. Closer images have the problem of being highly susceptible to robot position error, and the advantage of being able to pull out a clearly defined dock from the image by feature extraction.

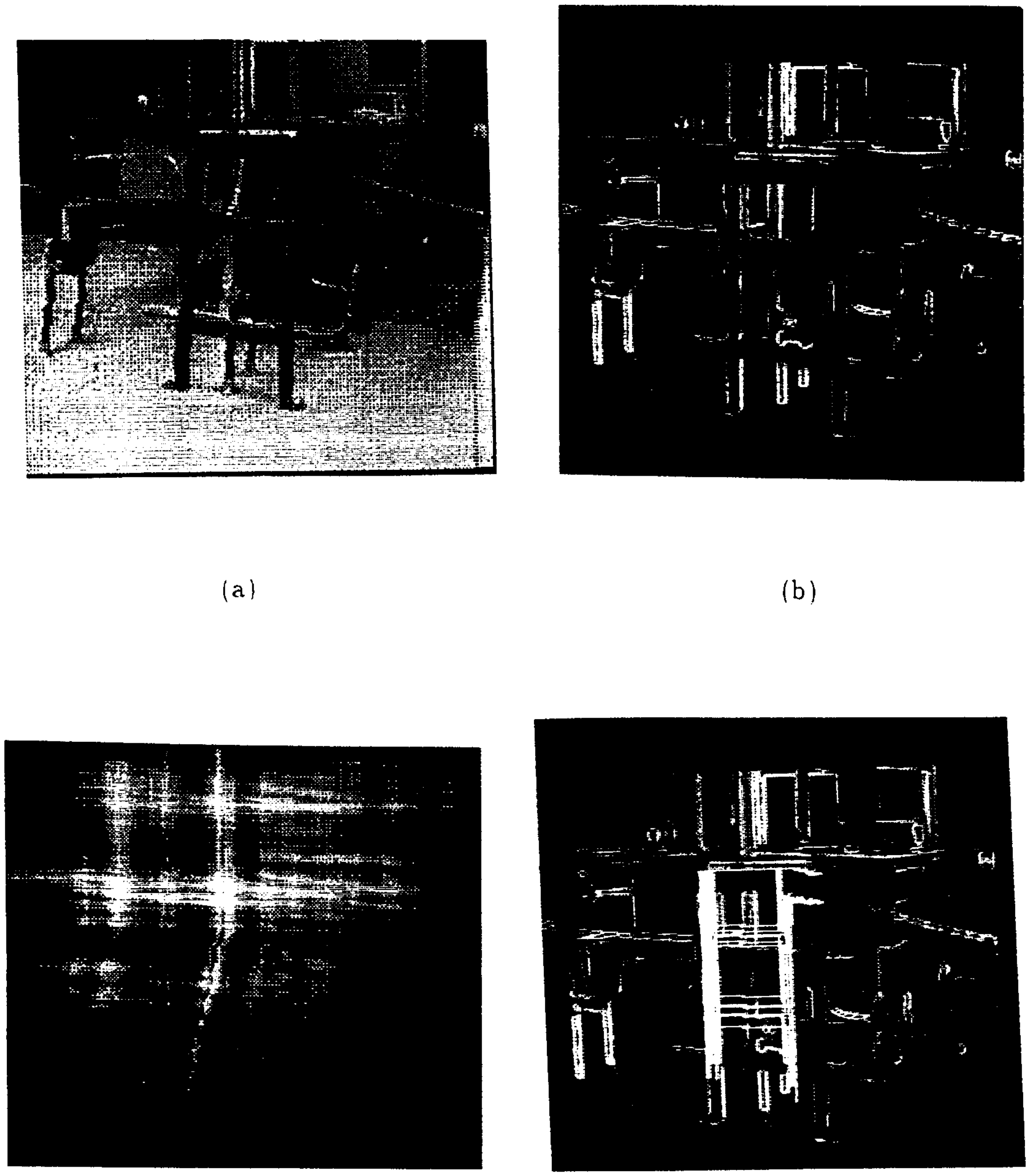
Therefore, there is an effective range for Htrans in this environment using these conditions. It is estimated that this range is somewhere between 10 and 20 feet. Images taken from less than 10 feet distant are too susceptible to robot position error due to scaling error (although they perform very well when this error is low). Images taken from more than 20 feet distant will probably not yield good results from feature extraction.

6 Summary

In this work, we have presented an implementation of a constrained Hough transform which has been successful in locating a docking work station accurately enough to guide a robot towards the dock in real manufacturing environments. It has been shown that this method is capable of performing even with a positional error of up to 4 feet and, therefore, meets the requirements of the docking application for this research.

There are several unusual features in this implementation of the Hough transform. First, an uncertainty map is used to constrain the possible locations of the robot relative to the dock. Second, a small cluster of Htrans' are run about the given location of the robot. Third, each Htrans casts its votes within the same two dimensional space, thus creating an overlapped Hspace.

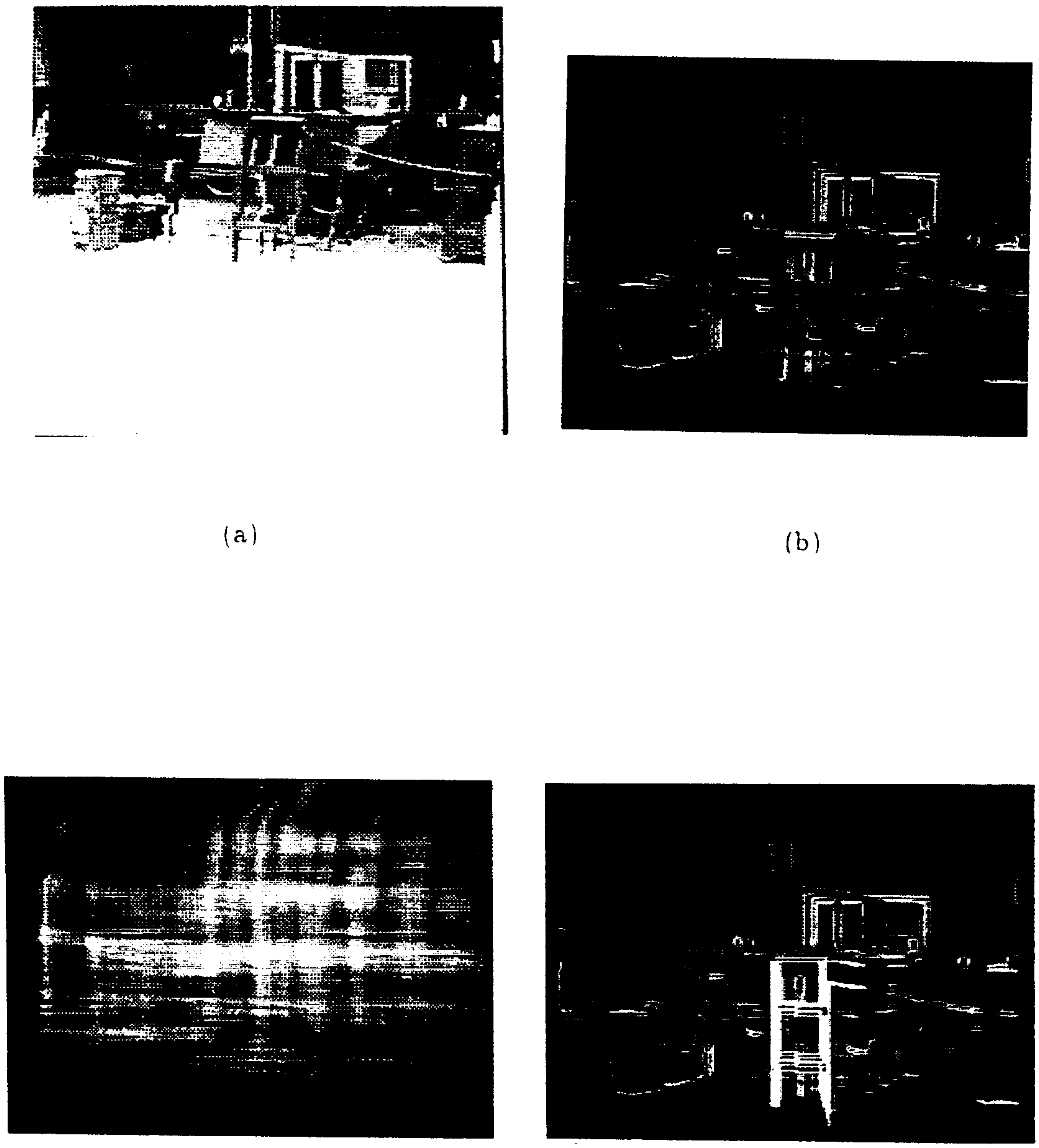
In future work, the algorithm will be expanded to utilize lines and corners instead of edge points. It is expected that these implementations will require less time to execute but may not be as robust.



(a) (b)

(c) (d)

Figure 7: Images generated for run of Htrans at position (-4,10).
(a) Raw image taken from position (-4,10).
(b) Sobel magnitude image.
(c) Collapsed Hspace image.
(d) Model cluster overlapped on top of the Sobel output at the point indicated by the peak in the Hspace.



(a)

(b)

(c)

(d)

Figure 8: Images generated for run of Htrans at position (-8,16).
(a) Raw image taken from position (-8,16).
(b) Sobel magnitude image.
(c) Collapsed Hspace image.
(d) Model cluster overlapped on top of the Sobel output at the point indicated by the peak in the Hspace.

True robot location: (6,7) Distance: 9.2 ft. Orientation: 42°
True dock location in image: (147,57)
Width of dock's front in image: 62 pixels

Reported robot location	Robot offset	Reported dock location in image	Dock offset from true location	Dock offset as % of dock width in image
(6,7)	1 foot	(147,58)	1.0 pixels	1.3
(7,7)	1	(147,58)	1.0	1.3
(6,8)	2	(147,57)	0.0	0.0
(8,7)	2	(147,60)	3.0	4.1
(9,7)	3	(155,57)	8.0	11.1
(6,10)	3	(149,57)	2.2	3.1
(6,11)	4	(191,57)	44.0	61.1
(10,7)	4	(190,57)	43.0	59.7

Figure 9: Results for a series of runs with image taken from position (6,7).

True robot location: (-4,10) Distance: 10.8 ft. Orientation: 111.8°
True dock location in image: (153,84)
Width of dock's front in image: 71 pixels

Reported robot location	Robot offset	Reported dock location in image	Dock offset from true location	Dock offset as % of dock width in image
(-5,10)	1 foot	(152,84)	1.0 pixels	1.4
(-4,11)	1	(150,82)	3.6	5.1
(-6,10)	2	(147,84)	6.0	8.5
(-4,12)	2	(149,82)	4.5	6.3
(-7,10)	3	(143,84)	10.0	14.1
(-4,13)	3	(142,81)	11.4	16.1
(-8,10)	4	(137,82)	16.1	22.7
(-4,14)	4	(140,85)	13.0	18.3

Figure 10: Results for a series of runs with image taken from position (-4,10).

True robot location: (-8,16) Distance: 17.9 ft. Orientation: 63.4°
True dock location in image: (152,93)
Width of dock's front in image: 40 pixels

Reported robot location	Robot offset	Reported dock location in image	Dock offset from true location	Dock offset as % of dock width in image
(-9,16)	1 foot	(152,95)	2.0 pixels	5.0
(-8,17)	1	(152,97)	4.0	9.9
(-10,16)	2	(148,97)	5.7	14.2
(-8,18)	2	(152,97)	4.0	9.9
(-11,16)	3	(148,97)	5.7	14.2
(-8,19)	3	(148,97)	5.7	14.2
(-12,16)	4	(144,97)	8.9	22.1
(-8,20)	4	(145,95)	7.3	18.1

Figure 11: Results for a series of runs with image taken from position (-8,16).

References

[1] Andresen, F.P.; Davis, L.S., "Visual Position for Autonomous Vehicle Navigation", *1985 IEEE International Conference on Robotics and Automation*, pp. 856-861, Nov. 1984.

[2] Arkin, R.C.; Riseman, E.; Hanson, A.,"AuRA: An Architecture for Vision-based Robot Navigation", *Proc. DARPA Image Understanding Workshop*, pp. 417-431, 1987.

[3] Arkin, R.C.,"Spatial Uncertainty Management for a Mobile Robot and its Role in Expectation Perception", *Robot Control 1988*, pp. 279-284, October 1988.

[4] Arkin, R.C.; Murphy, Robin R., "Autonomous Navigation in a Manufacturing Environment", *IEEE Transactions on Robotics and Automation*, pp. 445-454, August 1990.

[5] Ballard, D.H., "Viewer Independent Shape Recognition", *Pattern Analysis and Machine Intelligence, 5, No. 6*, Nov. 1983, pp. 653-660.

[6] Drake, K.C.; McVey E.S., "Experimental Position and Ranging Results for a Mobile Robot", *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, Feb. 1987, pp. 31-42.

[7] Hakalahti, H.; Moring, I., "Method of 2D and 3D Object Recognition in Nonoptimal Conditions", *SPIE Intelligent Robots and Computer Vision*, vol. 521, 1984, pp. 24-33.

[8] Hough, P.V.C., "Method and Means for Recognizing Complex Patterns", U.S. Patent 3,069,654, Dec. 18, 1962.

[9] Koch, M.W.; Kashyap, R.L., "A Vision System to Identify Occluded Industrial Parts", *1985 IEEE International Conference on Robotics and Automation*, pp. 55-60, 1985.

[10] Stark, L.; Eggert, D.; Bowyer, K., "Aspect Graphs and Nonlinear Optimization in 3-D Object Recognition", *Second International Conference on Computer Vision*, pp. 501-507, Tampa, December 1988.

[11] Stewman, J.; Bowyer, K., "Creating the Perspective Projection Aspect Graph of Polyhedral Objects", *Second International Conference on Computer Vision*, pp. 494-500, Tampa, December 1988.