CAR-TR-100
CSC-TR-1458

November 1984

Visual Position Determination for Autonomous Vehicle Navigation

Frederick P. Andresen
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, MD 20742

COMPUTER VISION LABORATORY

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

2

CAR-TR-100
CSC-TR-1458

November 1984

Visual Position Determination for Autonomous Vehicle Navigation

Frederick P. Andresen
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

This report describes a system by which an autonomous land vehicle might improve its estimate of its current position. This system selects visible landmarks from a database of knowledge about its environment and controls a camera's direction and focal length to obtain images of these landmarks. The landmarks are then located in the images using a modified version of the generalized Hough transform and their locations are used to triangulate to obtain the new estimate of vehicle position and position uncertainty.

-A-

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAK70-83-K-0018 (DARPA Order 3206) is gratefully acknowledged.

1. Introduction

The research described in this report is part of a larger project which has as its final goal the demonstration of an autonomous, visually guided, land vehicle [Davi85, Waxm85]. The vehicle will reach a desired destination by planning and following dynamically chosen and revised paths. As planning proceeds, the vehicle is commanded to move from point to point. However, errors in this movement create a positional uncertainty proportional to the distance travelled. It is therefore desirable to have a sub-system which can re-calculate the current position and reduce the uncertainty to within acceptable limits. A collection of algorithms for such a system has been designed and partially implemented in a research environment. The system uses the knowledge of the vehicle's approximate position to visually locate known landmarks. It then triangulates using the bearings of the known landmarks to acquire a new position with a reduced uncertainty.

The system is composed of three modules, called the MATCHER, the FINDER, and the SELECTOR, that interact to establish the vehicle's position with a new level of uncertainty.

1) The MATCHER locates likely positions for one or more landmarks in an image, and rates these locations according to some measure of confidence.

2) The FINDER controls the pointing direction and focal length of the camera to acquire specified images for a set of landmarks and directs the MATCHER to find possible locations for these landmarks in the images. It then eliminates possible locations for individual landmarks which are not consistent with the possible locations found for other landmarks. The FINDER then evaluates the remaining possible locations to determine the actual locations of the given landmarks.

3) The SELECTOR identifies a set of landmarks whose recognition in images of appropriate angular resolution would improve the position estimate of the vehicle by the desired amount. It then directs the FINDER to establish likely locations in such images for subsets of those landmarks. With these locations, the SELECTOR then computes new estimates of the vehicle position and position uncertainty and directs the FINDER, if necessary, to locate additional subsets of landmarks.

We assume the vehicle's camera is mounted on a computer controlled pan and tilt mechanism and has a computer adjustable focal length. We also assume estimates are available for the heading of the vehicle, as well as the current settings of the pan, tilt, and focal length of the camera. A database of landmarks exists that includes all pertinent landmark qualities, such as size and position, and at least one representation of each landmark from which it could be recognized in an image.

Chapter 2, 3, and 4 describe the MATCHER, FINDER, and SELECTOR, repectively. In Chapter 5, we describe an implementation of the algorithms.

Related literature is discussed in Chapter 6.

2. The MATCHER

2.1. Overview

A generalized Hough transform is employed to locate landmarks of known image orientation and scale. The landmarks are represented by lists of boundary points which are individually matched to edge points in the image. The algorithm consists of three main phases: edge point detection, matching of the template to the edge points, and interpreting the results of the matching. Edges are detected as points where the Laplacian changes sign and the local grey levels have a high symmetric difference. Matching is done using the generalized Hough transform and is restricted in two ways. First, template points match only points having close to the same gradient direction. Second, only those template points are used whose gradient directions have a high measure of informativeness; this measure is defined in Section 2.3.3. Throughout this chapter, the description of the algorithm will be supplemented by references to the two examples in Figures 2.1 and 2.2.

2.2. Getting the edge image

The image given is a grey-level picture (Figures 2.1a and 2.2a). It is first smoothed using a local edge-preserving smoothing operator. This operator, the symmetric nearest-neighbor algorithm, can be computed on any size neighbor-

hood, but is simplest to describe for a 3×3. A brief description of the 3×3 algorithm is provided below; for a full explanation see Harwood, Subbarao, and Davis [Harw84].

For each pixel **p** in the image, consider a 3×3 neighborhood. For each pair of symmetrically opposed neighbors, choose the neighbor whose grey level is nearest in value to the center pixel's. In the case of a 3×3, there will be four pairs and thus four points chosen. Replace **p** with the mean of these points. One could also use the median of the four points, which results in better preservation of corners; however it is slightly slower than using the mean and often the improvement is not enough to warrant the extra time.

The implementation used in this thesis is a 5×5 version of this algorithm with the mean computed for the nearest neigbors. Also, since the algorithm's results improve with iteration, two iterations are done on the image (Figures 2.1b and 2.2b).

A neighborhood size for the Laplacian, which is appropriate for the size of the object being sought, is then selected and the Laplacian is convolved with the smoothed image. At present, this selection is done manually and usually the size is very small, such as 3×5 or 3×7. However, the selection could be done automatically using such criteria as image size of the object or density of edge points, or average local standard deviation of edge direction.

At any point where the Laplacian crosses zero (a positive pixel with a negative neighbor), the local symmetric contrast is computed. This is done by taking

the maximum absolute difference in grey level of any two symmetrically opposite points in a 3×3 neighborhood. This is a fast and isotropically smooth edge strength measure and is used to eliminate all the false or weak edge points given by the zero-crossing of the Laplacian. Using this measure of edge strength, the weakest 75 percent of the zero-crossing points are eliminated. The result is an image of thin contours (due to the zero-crossing operator) whose edge strength is significant (Figures 2.1c-e and 2.2c-e). Many of the contours are often broken by single pixels of low contrast, but this does not affect the matching procedure which matches patterns of individual points and not patterns of extended contours.

2.3. Matching

A generalized Hough transform (GHT), incorporating the gradient direction at points, is used to perform the matching. The GHT and the specific implementation used are described in Section 2.3.1. Certain assumptions made regarding the orientation and scale of the object in the image are explained in Section 2.3.2. Finally, a discussion of the gradient direction informativeness measure introduced in Section 2.3.1 is given in Section 2.3.3.

2.3.1. The generalized Hough transform

The generalized Hough transform is a fast point pattern matching algorithm that can be used to detect arbitrary specific shapes in images (see [Ball81] and [Davi82]). The general problem solved is to find the function that best transforms a set of object points (i.e., the shape) into a set of image points. This

function will be in terms of the parameters allowed to vary in the transformation. The result of the generalized Hough transform is a "Hough space" which indicates the likelihood of any particular transformation being the correct one for the given object and image. In our application, we are only concerned with finding a transformation function that performs translation in a plane. All other transformation parameters (such as image scale and orientation) are assumed to be fixed and known.

In our implementation, the object points are the boundaries of a landmark being viewed from an (approximately) known direction and distance. The image points are those found using the edge finding algorithm described in the previous section. The gradient direction is calculated at both object points and image points. Figures 2.1f and 2.2f show the edges used in the image with the gradient directions represented by grey levels. Figures 2.1g and 2.2g show the edges and directions of the object boundaries. The object points are organized into an "offset table" indexed on the gradient direction with a set of (x,y) pairs for each gradient direction. The (x,y) pairs indicate the offsets to some arbitrary reference point. We have used the centroid of the boundary points as the reference point. For n gradient directions, the table is of the form

Offset Table	
Gradient Direction	Offsets to reference point
ϕ_1	$(s_1, t_1)_1, (s_2, t_2)_1, \ldots, (s_m, t_m)_1$
ϕ_2	$(s_1, t_1)_2, (s_2, t_2)_2, \ldots, (s_p, t_p)_2$
ϕ_n	$(s_1, t_1)_n, (s_2, t_2)_n, \ldots, (s_q, t_q)_n$

Each edge point in the image with gradient direction ϕ_i may correspond to one or several object points with gradient direction ϕ_i. For each possible correspondence, there will be a potential reference point. The positions of the reference points, relative to the edge point, are given by the (x,y) pairs for ϕ_i in the offset table. Therefore, at each edge point **e** in the image with gradient direction ϕ_i, the set of possible reference points can be obtained by adding each offset for gradient direction ϕ_i to the position of **e**. The generalized Hough transform operates by incrementing, in an accumulator array, all the possible reference points for each edge point in the image. The accumulator array is the Hough space mentioned above.

In the general case, the maxima in the array A represent the most likely parameter values which would characterize the best transformation functions from the object to the image. In our case, the maxima in A represent possible locations for the object in the image.

2.3.2. Orientation and scale

As mentioned above, we assume we know the orientation and scale of the object in the image within a given tolerance. The template can therefore be scaled and rotated to match the appearance of the object in the image. This eliminates the need for the generalized Hough transform to include scale and orientation parameters; however, the implementation used should allow for errors in both parameters. This section describes how our implementation allows for

small errors.

Orientation errors effect the algorithm through mismatches of gradient direction between image edge points and boundary points in the template. A mismatch can occur because of errors in the measurement of the gradient direction, noise near the edge point in the image, actual local differences between the object's silhouette in the image and the template, or a grossly inaccurate assumption of the orientation of the object in the image.

Two measures have been taken in an effort to increase the likelihood of the correct reference point being incremented for each of the above cases. First, when the gradient direction is calculated, it is rounded to the nearest 10 degrees. Second, during the matching process, instead of matching points only when their gradient directions are equal, edge points with gradient directions within ± 15 degrees of a template point's gradient direction are also matched.

Small errors in scale effect the algorithm by incrementing points in the Hough space which fall just short or just long of the actual reference point. This will create a faint inverse silhouette of the object in the Hough space whose size indicates the magnitude of the scaling error. When the error is very small (one or two pixels), the inverse silhouette is just a small diffuse dot. Post-processing of the Hough space can therefore find the center of the dot by local averaging.

2.3.3. Gradient direction informativeness

It can often occur that one or several gradient directions are so prevalent in the image that they produce strong voting clusters in Hough space at incorrect

locations. If a gradient direction occurs at N edge points in the image and at M template boundary points, then MxN increments are made for that gradient direction in the Hough space. Since usually only a small fraction of the edge points in the image are part of the object's boundary, the remainder of the matches can potentially contribute to false peaks in the Hough space.

Also, if a gradient direction is prevalent in the template, then, as a group, points with that gradient direction will contribute more correct votes than would points with an infrequent gradient direction. This is, of course, because there will be more boundary points with the prevalent gradient direction incrementing potential reference points. Therefore, when the reference point happens to be the correct location for the object, more of the votes contributing to its peak will come from points with the prevalent gradient direction than from points with an infrequent gradient direction.

To use these observations to best advantage, a measure of gradient direction informativeness (GDI) was developed to rate the gradient directions. Then, only those points whose gradient directions rate highly are used in the matching. In this way, we can eliminate the uninformative sources of spurious patterns in the Hough space and make best use of the most informative points. The measure used is $\frac{P[\,G\,]_t}{P[\,G\,]_i^2}$ where $P[\,G\,]_t$ is the probability that gradient direction G occurs in the template and $P[\,G\,]_i$ is the probability that gradient direction G occurs in the image. The actual probabilities are extracted from histograms of the template and the image. Based on this measure, only the most informative 15 percent of

the edge points in the image are used in the matching. Consequently, boundary points in the template whose gradient directions are not selected will not be used.

It can be seen that gradient directions that occur often in the template but infrequently in the image would rate very high on this scale. Also, gradient directions with few occurences in the template but many in the image would rate very low. Points with such gradient directions would yield a high number of unrelated votes, cluttering the Hough space and creating false peaks.

This measure has proved very successful in several tests. Figures 2.3 and 2.4 show the Hough space for two pictures with varying degrees of filtering using the gradient informativeness measure. It is clearly most useful when a few gradient directions, which are not essential to locating the object, dominate the image.

Figure 2.5 shows the object points actually used in the matching for our two examples. In each case, the most informative 15 percent were used.

2.4. Finding the peaks in the Hough space

The matching algorithm described in Section 2.3 produces a two-dimensional Hough space the same size as the image being searched (Figs. 2.1h and 2.2h). The local intensity peaks in this image represent possible locations for the object in the image. To avoid making false conclusions when near ties occur, we produce a list of the possible peaks with their respective confidences. This allows the decision about which peak represents the actual location to be passed to higher level decision-making systems. The process of producing the list from the Hough space can be thought of as several passes of simple neighborhood

operators:

(1) Sum the votes in a K×K neighborhood.
(2) Perform non-maximum suppression on a J×J neighborhood, i.e., eliminate all points having a neighbor in a J×J neighborhood with a higher sum.
(3) Compute confidences of the remaining points.

This process could be quite inefficient if it were computed on array-formatted pictures of any significant size; therefore, some reasonable limits were imposed on the numbers and values of points that were of importance at each step in the process. The resulting algorithm, using these limits and lists of sorted points, is as follows.

(1) The N points having highest vote counts are selected and sorted into a list on vote count. (N is selected to ensure not eliminating the correct object location. Fifty has proved to be a sufficient number for all cases tried.)
(2) All points whose vote count is below M percent of the highest value are eliminated.
(3) For the remaining points, the vote count is replaced by a sum of the vote counts in a $K \times K$ neighborhood and a new list sorted on this value.
(4) In this new list, those points are again eliminated which are below M percent of the highest in the list.
(5) Non-maximum suppression is used to identify the local maxima. Starting at the low end of the sorted list of summed points, each point, call it **s**, is compared to each of the points below it. If a point, **t**, below point **s** is in a $J \times J$ neighborhood centered on **s**, then eliminate **t**. The algorithm must start at the bottom of the sorted list so that points are not eliminated before they have a chance of eliminating others.

A measure of confidence is now calculated for each remaining point. It is designed to indicate both the strength of the peak and how it relates to other surviving peaks. If it is necessary to compare confidence measures for peaks in several images of different size and edge content, then this confidence measure should be normalized.

If V_x is the number of votes for location **x** (summed over a $K \times K$ neighborhood) and there are **n** possible positions, then the confidence measure is computed as follows:

$$C_x = \frac{V_x}{\sum_{i=0}^{n} V_i} \times 100 \quad .$$

In words, the confidence measure C_x is the number of votes that location **x** received (summed over a $K \times K$ neighborhood centered on **x**) expressed as a percentage of the total votes given to all points (summed over $K \times K$ neighborhoods) in the list of possible positions.

Figures 2.1i-j and 2.2i-j show the most confident peak found using this algorithm. The template boundary points are overlayed on both the Hough space and original image.

2.5. Suggestions for improvements

Certain simplifying assumptions are made in the MATCHER. Although some are reasonable for the intended application, in general the system is less robust because of them. One such assumption, that the scale and orientation of the landmark in the image are known, can be eliminated by including the scale and image orientation of the landmark as parameters in the Hough transform. Although this would result in a four dimensional Hough space, the range of values in each dimension could be limited and therefore make computation time reasonable.

A possible improvement to the MATCHER is, after a set of possible positions is found, to try to match the landmark model at these points without the GDI filter that was applied in the first matching process. This would strengthen and better delineate the actual location of the landmark.

3. The FINDER

3.1. Overview

This chapter describes a strategy (the FINDER) for determining bearings to a given set of landmarks. The FINDER is also given specifications for images in which it can expect to find these landmarks. It then controls the camera to obtain these images and uses the MATCHER to establish likely locations for the landmarks in their respective images. Since the search for any specific landmark may result in several possible locations for that landmark (at most one of which, of course, can be correct), we employ a simple geometric constraint propagation algorithm to eliminate many of the false locations.

The geometric constraint propagation algorithm considers possible locations for a pair of landmarks and determines if they could both be the correct location for their respective landmarks. Two possible locations (or, more briefly, peaks) are called *consistent* if they meet this criterion. The details of this consistency computation are described below. With consistency determined for all pairs of peaks, a graph is then constructed in which nodes are peaks, and arcs represent the mutual consistency between two peaks. Analysis of this graph can determine consistency among groups of more than two peaks and therefore eliminate peaks based on more than just pairwise inconsistency.

3.2. Consistency between peaks

To determine consistency between two peaks p_1 and p_2 for landmarks L_1 and L_2, we first calculate a range of possible angular differences between L_1 and L_2 based on the vehicle's position uncertainty. We then extend this range by the pointing error and check that the measured angular difference between p_1 and p_2 falls within this range.[1] See Figure 3.1.

The angular difference between L_1 and L_2 is determined by simply taking the difference of their bearings. The range of angular differences is then obtained by letting the value for the current position vary according to the position uncertainty of the vehicle. For the purposes of this analysis, we assume that the position uncertainty can be represented by a solid disc on the local ground plane. If we make the reasonable assumption that neither landmark lies inside the disc, then it is easy to show that the positions which give the maximum and minimum angular differences will always lie on the circumference of the disc. To prove this, we give the following informal argument.

Assume the contrary, that **p** is a point inside the disk for which the angle L_1pL_2 is maximum. Consider a line which bisects this angle. Since **p** is inside the disc, there must exist a point **p'** on that line which is also inside the disk and is closer to both L_1 and L_2. See Figure 3.2. Clearly the angle $L_1p'L_2$ would be greater than L_1pL_2. This contradicts our assumption.[2] A similar argument can be applied if L_1pL_2 is assumed to be a minimum.

[1] We could also account for the error due to pixel size, but do not since it is negligible.

[2] Note that this does not imply that the solution points will be at the intersection of the bisector and the circumference of the disk. We could find no such simple intuitive solution to this problem.

From these positions we can then calculate directly the maximum and minimum angular difference between L_1 and L_2. An abbreviated derivation of an analytic solution for these points follows.

We define the positions of the two landmarks L_1 and L_2 to be $(x_1,y_1)_M$ and $(x_2,y_2)_M$, in map coordinates. The current vehicle is at $(x_0,y_0)_M$, and the disk of uncertainty is centered on this location and has radius r. We then transform all locations from map coordinates into a coordinate system centered on the vehicle position. All coordinates from now on will be in this new coordinate system.

Two lines with slopes m_1 and m_2 meet at an angle ψ (measured from line 1 to line 2 counter-clockwise) given by

$$\psi = atan\left(\frac{m_2 - m_1}{1 + m_1 m_2}\right) . \qquad (1)$$

If $m_1 = \frac{y_1 - y}{x_1 - x}$, $m_2 = \frac{y_2 - y}{x_2 - x}$, (x,y) is the vehicle's actual position, and (x_1,y_1) and (x_2,y_2) are the locations of two landmarks, then equation (1) would determine the angular difference between the two landmarks.

We can find the extrema of ψ by differentiating equation (1) and setting it equal to zero. Since (x,y) is constrained to lie on the disk's circumference, we can represent (x,y) as $(r\cos\theta, r\sin\theta)$. Note that there is only one variable, θ, since r is a constant. This makes $m_1 = \frac{y_1 - r\sin\theta}{x_1 - r\cos\theta}$ and $m_2 = \frac{y_2 - r\sin\theta}{x_2 - r\cos\theta}$. Differentiating equation (1) and simplifying, we obtain

$$\frac{d\psi}{d\theta} = \frac{1}{1+\frac{s^2}{t^2}} \cdot \frac{\frac{ds}{d\theta}\cdot t - s\cdot\frac{dt}{d\theta}}{t^2} = \frac{1}{\frac{t^2+s^2}{t^2}} \cdot \frac{\frac{ds}{d\theta}\cdot t - s\cdot\frac{dt}{d\theta}}{t^2} = \frac{\frac{ds}{d\theta}\cdot t - s\cdot\frac{dt}{d\theta}}{t^2+s^2} \qquad (2)$$

with

$$u = \frac{s}{t} = \frac{-r\cos\theta(y_2 - y_1) + x_1y_2 - x_2y_1 - r\sin\theta(x_1 - x_2)}{-r\sin\theta(y_2 + y_1) + y_1y_2 + x_1x_2 - r\cos\theta(x_1 + x_2) + r^2} .$$

Setting equation (2) to zero, we obtain

$$0 = \frac{ds}{d\theta}\cdot t - s\cdot\frac{dt}{d\theta} = -N - M\sin\theta - P\cos\theta \qquad (3)$$

where

$$N = r\,(y_2^2 - y_1^2 + x_2^2 - x_1^2)\,,$$

$$M = -y_1y_2^2 + y_1^2y_2 + x_1^2y_2 + r^2(y_1 - y_2) - x_2^2y_1 \text{ , and}$$

$$P = x_1(-y_2^2 - x_2^2) + x_2y_1^2 + x_1^2x_2 + r^2(x_1 - x_2) \text{ .}$$

Since $\cos\theta = \frac{x}{r}$, $\sin\theta = \frac{y}{r}$, and $y = \sqrt{r^2 - x^2}$, we can rearrange equation (3) to get the quadratic

$$0 = (P^2 + M^2)x^2 + 2NMrx + (N^2 - P^2)r^2. \qquad (4)$$

At the solution points to (4), we calculate the angle between the two individual landmark points and subtract using the equation

$$\phi = \phi_2 - \phi_1 = atan\left(\frac{y_2 - y}{x_2 - x}\right) - atan\left(\frac{y_1 - y}{x_1 - x}\right) = atan(m_2) - atan(m_1).$$

Calculating ϕ for each solution, we get the maximum and minimum angular differences between the two landmarks L_1 and L_2 given a position uncertainty of $\pm r$.

At this point we extend this range to allow for the pointing error of the pan/tilt mechanism. This is done by simply adding the error to the range.

3.3. Consistency Graph Analysis

As mentioned above, the consistency graph represents consistency relations between the possible locations for different landmarks. Ideally, we would want to determine the maximal complete subgraphs (MCS's) of this graph because they would represent the largest sets of landmark locations that are all mutually consistent. For small graphs this is not impractical, but for large graphs we might be forced, due to time constraints, to perform a simpler analysis.

We can, for example, apply certain simple iterative tests to the graph that would eliminate any landmark location not part of at least a k-clique. In what follows, we identify two simple tests for eliminating nodes not part of k-cliques. These processes are similar to so-called "discrete relaxation" algorithms - see, e.g., Haralick and Shapiro [Hara79].

First, we can iteratively eliminate all nodes which do not have arcs to nodes representing at least k other distinct landmarks. After this process is complete, we can then eliminate all nodes which are not the center of what we refer to as a **k-fan**. A node **n** is the center of a k-fan if there exists a connected chain of nodes of distinct landmarks of length k-1 in which each element of the chain is connected to **n**. Figure 3.4 contains an example of applying both node deletion processes to a graph of landmark locations. Finally, we find all MCSs for this pruned graph.

Since we could end up with several MCSs, we now need a way to determine which is the actual set of landmark locations. To do this, we define an evaluation function to operate on the MCSs and then pick the MCS which responds best to the evaluation function. In our current system, we use a simple summation of the confidences for each of the possible locations.

4. The SELECTOR

4.1. Overview

This chapter describes a strategy (the SELECTOR) for selecting a set of landmarks whose identification in appropriate images would improve the current estimate of the vehicle's position. The SELECTOR supplies subsets of these landmarks, with appropriate image specifications, to the FINDER which returns the most likely relative positions for each landmark in each subset. The SELECTOR then computes the vehicle's actual location and the new uncertainty associated with it. If this new uncertainty is insufficient, then the SELECTOR can either simply accept the new uncertainty as the best achievable result, or try to further improve the position estimate using other landmarks.

Given a database of visual landmarks, a variety of strategies can be employed to select a subset of those landmarks for identification. The implementation of any of these strategies requires the ability to determine both the ease of identification of any given landmark and the effect of its identification on the vehicle's position uncertainty. The development of these abilities is described in Sections 4.2 and 4.3. Further discussion of the SELECTOR module continues in Section 4.4.

4.2. Determining ease of identification

Many factors effect the ease of identification of a landmark. Some examples are the size of the landmark in the image being matched, the stability and geometric complexity of the landmark's model from the current vantage point, and the position of the sun. In a more sophisticated system, we may have information about the visual surroundings of the landmark and be able to consider the landmark's relative uniqueness in the image as a factor. In this section we consider two factors: 1) the ability to obtain an image that will allow us to accurately locate a particular landmark, and 2) the suitability of the landmark's model for use with the MATCHER.

4.2.1. Suitable image verification

The two quantities that determine an image are direction and focal length (or field of view). The direction is constrained by the fact that extremely bright scenes (i.e., those containing light sources) will most likely saturate the camera and result in a pure white or washed out image. It would clearly be futile to search for a landmark in such an image. The directions in which these scenes might occur could be predicted by using special instruments or by analyzing failures of the MATCHER. To determine the camera direction, we simply use the bearing of the landmark with respect to the current presumed vehicle position. This can be calculated straightforwardly from the coordinates of the landmark and vehicle. We can then verify that the camera does not point towards a "bright scene".

There at least four constraints on the appropriate focal length for a suitable image.

(1) The focal length must be short enough for the image to contain the landmark after accounting for all the errors in our estimation of its position.
(2) The focal length must be long enough to insure that locating the landmark would improve the position estimate.
(3) The focal length must be long enough to guarantee that the landmark will appear in the image with a size (in pixels) large enough for it to be reliably located.
(4) The above three constraints must be satisfied by at least one focal length within the available range of focal lengths for our imaging system.

The first three constraints are discussed in the next three sections. The last constraint and the verification process are addressed in the following section.

4.2.1.1. Determining the minimum field of view

In order to insure that we obtain an image large enough to contain a particular landmark, we need to know the physical size of the landmark, its position, the vehicle's current position, the vehicle's current position uncertainty, and the pointing error of the camera control system. These parameters determine the field of view (fov) necessary to include the landmark.

To determine the minimum field of view needed, we use a method illustrated in Figure 4.1. The factors considered are the size of the landmark and our ability to point the camera at the landmark. Our ability to point the camera in the correct direction depends on how well the landmark's bearing can be approximated from our current position, and how precisely the camera can be pointed in that direction.

Our approximation of a landmark's bearing can only be as accurate as our approximation of the current vehicle location. If a set of halflines, emanating from a landmark, are extended through all the possible vehicle locations, then they form a wedge with an angular width θ. This angle θ is the amount of angular uncertainty with which either the landmark or the vehicle could locate each other.

Now, a landmark L with width W will subtend an angle ϕ in the field of view which depends on the distance D to the landmark. Since the focal length will be very small compared to the distance, D, to the landmark, we can approximate D - focal length, the distance from the landmark to the center of focus, by D. Therefore, L subtends approximately $\phi = \arctan(\frac{W}{D})$.

The camera control mechanisms will certainly have an inherent angular uncertainty error (pointing error) which we will denote by ψ. The same will be true of the heading feedback equipment and we will denote this orientation uncertainty by α.

We can now simply add these four angular uncertainties together to arrive at a total field of view which, if centered on our best approximation of the landmarks location, will be guaranteed to contain the landmark. Therefore, the minimum fov for L is,

$$\text{minimum-FOV} = \theta + \phi + \psi + \alpha \ .$$

For a camera with a digitizing surface of size f_s (in millimeters), the minimum-FOV can be obtained by using a focal length max-fl given by

$$\text{max-fl} = \frac{fs}{\tan(\text{minimum-FOV})} .$$

4.2.1.2. Satisfying the accuracy requirement

The objective of the SELECTOR is to reduce the vehicle's position uncertainty to a certain amount. Since the new uncertainty of the vehicle location is determined by the bearings for landmarks of known positions, the accuracy of those bearings determine the accuracy of the new vehicle position. We now describe how to express this accuracy requirement as a minimum focal length for the image in which we identify that landmark.

The achieved angular accuracy is determined by the pointing error of the pan/tilt mechanism , the error in orientation estimation and the width of a pixel in the image where the landmark is located. A pixel will subtend a certain angle in the field of view (called the *pixel angle*[3]), that depends on the focal length of the lens, the width of the camera's imaging surface, the spatial resolution (number of pixels across) of the camera, and the position of the pixel in the image. Since the position of the landmark in the image can not be known at this point, we approximate it by the center pixel of the image.[4] Using this approximation, we have $\text{pixel angle} = \arctan(\frac{\text{film size}}{\text{focal length} \cdot \text{resolution}})$. The pointing error, ψ, and orientation error, α, are expressed as degrees in the pan and tilt direction. These errors in the pan direction are then added to the pixel angle to get the

[3] This is also known as the *instantaneous field of view* or IFOV, since the pixel is the result of one instantaneous sampling of the scene by the imaging device.

[4] This is not a critical assumption since even in the largest field of view the pixel angle would only vary by a very small amount across the entire image.

total angular error. That is,

$$\text{total angular error} = \psi + \alpha + \arctan\left(\frac{\text{film size}}{\text{focal length} \cdot \text{resolution}}\right). \quad (6)$$

To calculate the focal length needed to achieve the required accuracy, we simply solve for focal length in equation (6), as follows:

$$\text{minimum focal length} = \frac{\text{film size}}{\tan(\text{minimum angular error} - (\psi + \alpha)) \cdot \text{resolution}}.$$

With this minimum focal length established, we can then know when the bearing of a landmark, found in a particular image, meets the accuracy specifications given for that landmark.

4.2.1.3. Determining minimum size

We still need to insure that the landmark we are searching for will appear in this initial image with a size that maintains at least some of its unique identifying features. The minimum spatial resolution (number of pixels across the landmark), *min-pixels*, needed to insure this is determined a priori for each landmark. The minimum focal length, *min-fl*, is then given by

$$min\text{-}fl = \frac{\text{film size}}{\tan(min\text{-}pixels \cdot \text{pixel angle})}.$$

4.2.1.4. Constraint checking

To visualize the restrictions that these constraints put on our choice of focal length, refer to Figure 4.2 where they are displayed on a focal length axis. The thick part of the line axis represents the range of available focal lengths. Two of the constraints specify ranges bounded by minimum values and one specifies a range bounded by a maximum value. If these ranges overlap in a region that has

some part within the range of available focal lengths, then a suitable image is obtainable. In fact, the most desirable image, in light of these constraints, is most likely the one with the longest focal length in this acceptable region.

If the limiting factor for our choice of focal length is the maximum available focal length, then we can simply use a window from an image with a shorter focal length as long as this shorter focal length obeys the other constraints. This, in fact was done in the implementation of Chapter 5. Note that in a more sophisticated system, if the limiting factor is the minimum available focal length (i.e., the acceptable range is entirely below it), then several higher resolution images with smaller fields of view could be used to systematically cover the field of view that the original image would have covered.

4.2.2. Landmark model/MATCHER suitability

Recall that the MATCHER selects points to match based on gradient direction. Points are matched only if their gradient direction appears frequently in the model compared to its occurance in the image. Without future knowledge of the image, we can only assume that a model with a uniform distribution of gradient direction would be best. This is because if the model has a frequent gradient direction that also occurs frequently in the image, then this gradient direction will probably not be matched, causing a disproportionately large number of model points not to be matched. Therefore, we can use, for example, the standard deviation of the gradient direction counts as a measure of how uneven the distribution is for the model. This can then be compared to other landmarks to

help determine the best choice.

4.3. Determining the effect of identification

This section describes how to determine the extent to which finding a particular landmark's bearing would effect the vehicle's position uncertainty. To do this, we first explain how that landmark's bearing would be used (with other landmarks' bearings) to help determine the new vehicle location. Then, we show how to obtain an estimate of the uncertainty for this new vehicle location.

Given a pair of bearings (B_1,B_2) for two landmarks with known positions (x_1,y_1) and (x_2,y_2), we can find the actual vehicle location by intersecting the lines passing through (x_1,y_1) with angle B_1 and (x_2,y_2) with angle B_2. See Figure 4.3. If the bearing B_i to landmark L_i is only known to within $\pm\theta_i$, then the possible lines passing through (x_i,y_i) would sweep out a wedge W_i of angular width $2\theta_i$ on the ground plane. See Figure 4.4a. Since for each landmark, L_i, found the vehicle is constrained to lie in the planar wedge W_i, then the vehicle must lie in the convex polygon formed by the intersection of these wedges. See Figure 4.4b.

The size and shape of this convex polygon is determined by the width of each wedge at their intersection and the angles at which they intersect. The width U_i of a wedge W_i at a distance d_i from L_i is given by $U_i = 2 \cdot d_i \cdot \tan\theta_i$, where $\pm\theta_i$ is the uncertainty of the landmark bearing as calculated in Equation (6). Therefore, the effect of finding L_i's bearing on the vehicle location uncertainty is proportional to the angular uncertainty θ_i of the bearing and the distance from L_i to the actual vehicle location. Since the actual vehicle location is not known at

this point, we approximate it by the assumed current position.

Now, since the vehicle is known to be within a certain distance r of the assumed current position, we can initially constrain the vehicle to lie within a convex region centered on the current position. For the purposes of this analysis, we shall assume that this region is a square of dimension $2r$. Therefore, initially the effect of finding landmark L_i is that of finding the intersection of the wedge W_i and the square region of uncertainty. We will now discuss briefly a simple method for finding this intersection and evaluating its size.

A natural representation for the wedges and the initial uncertainty region is the intersection of halfplanes, primarily because the wedge is unbounded in one direction and our initial uncertainty region is convex. The initial uncertainty region is represented by the intersection of four halfplanes. Then we add to this set the two half planes which represent the first wedge. We now have the intersection of six halfplanes defining a new convex polygon[5] (see Figure 4.5a). As wedges representing the subsequent landmarks are added to the set of halfplanes, the convex polygon (resulting from the intersection of the halfplanes in the set) will get smaller and smaller (see Figure 4.5b).

To express in one parameter the uncertainty represented by a convex polygon, we find the two vertices which are furthest apart. Half of the distance between these two vertices is a reasonable approximation of the "radius" of this polygon. This "radius" can be compared to the original r to determine to what extent the vehicle's position uncertainty has been reduced.

[5] The intersection of any number of halfplanes is a convex polygon.

4.4. The SELECTOR continues

By maximizing both the ease of detection and the effect of identification as discussed in the previous sections, the SELECTOR arrives at a set of landmarks. During this process, it calculates a direction and optimal focal length for each landmark that specify an image in which that landmark can be identified. The SELECTOR then directs the FINDER to find the subsets of landmarks in their respective images. (One might search for only a subset to limit the amount of effort devoted to a potentially fruitless search; if some critical subset cannot be identified, a completely new set of landmarks could be chosen.)

The FINDER returns its best estimate of the locations of those landmarks along with the bearings calculated for them. From these bearings and the locations of the landmarks, the SELECTOR then computes new estimates of actual location and uncertainty. These calculations were described in Section 4.3. If all the landmarks were found as expected, then this new uncertainty will meet the initial uncertainty requirement. However, if only a subset of the landmarks selected for the set were found, they could, in some circumstances, determine an acceptable uncertainty. If they do not, then the SELECTOR can choose another combination of landmarks for analysis.

5. An Implementation

This chapter describes a partial implementation of the system described in the first four chapters in an indoor environment. The FINDER and MATCHER are completely implemented to the extent possible with the available equipment. The SELECTOR is partially implemented. The parts not implemented on the computer were done by hand and results supplied to the system when needed.

5.1. Experimental Environment

The environment was a terminal room adequately lit from above by several rows of fluorescent lights. The camera used was a Cohu 2800 with a manually adjustable zoom lens having a range of 20mm to 80mm. The camera was mounted on a large tripod with manual adjustments for pan, tilt, and spin. Programming was done in both C and Franz Lisp on a VAX 11/780 running under the UNIX operating system (BSD 4.2). For information on the interface between Franz Lisp and C see [Andr84].

5.2. Programming

Top level control of both the SELECTOR and FINDER is done with the YAPS production rule system for flexibility and extensibility. YAPS integrates naturally with Franz Lisp, as the right hand side of YAPS rules can be arbitrary sequences of lisp expressions. Lisp functions comprise most of the FINDER and

SELECTOR and are called from the right hand side of simple YAPS rules or from other lisp functions.

The database is comprised of YAPS facts and knowledge stored as lisp data structures. The YAPS facts are used for high-level decision making and the firing of YAPS rules. The lisp data structures are generally symbols having property lists containing global information for the system. For instance, all the landmark information is stored in this manner.

The MATCHER is in two parts, the Hough transform and the Hough space analysis. Both were written completely in C, but are used as lisp functions. This is the case with many other functions which might have required a large amount of numerical computation, such as the calculations for the minimum and maximum angular difference between landmarks (in Section 3.1.1).

5.3. Operation of the system

As mentioned above, only parts of the SELECTOR module are currently implemented. These parts are the computation of a minimum image to include a particular landmark and the triangulation to arrive at the new position. Therefore, in this example the author functioned as the SELECTOR for many tasks. What follows is a simplified step by step description of the operation of the system (divided into sections labeled with the module currently in control).

5.3.1. The SELECTOR selects

The room was first examined for interesting landmarks which could be easily located and seemed disparate enough to result in a small uncertainty for the new

position. It was decided that three landmarks would be a good number for demonstrating the system. They were a wall phone, a coffee cup taped to a blackboard, and a wall outlet with a cord plugged into it. The positions and sizes of these landmarks were measured precisely and entered into the database. The layout of the landmarks in the room is shown in Figure 5.1. The existance of suitable images for these landmarks was then verified followed by the determination of those suitable images with the longest focal lengths. They were as follows:

Image specifications			
Landmark	Bearing of image center	Focal length (mm)	FOV (degrees)
phone	58	54	10
cup	96	96	6
plug	139	114	5

Templates for eventual use by the MATCHER were then created for these landmarks. This was done by obtaining high resolution images in which the landmarks were distinct from the surroundings. These images were then reduced to the size they would appear at from the vehicle's assumed current position. The gradient direction was then calculated at strong edge points to obtain a set of object points suitable for the MATCHER (see Chapter 2).

The landmarks chosen were then specified to the FINDER along with the corresponding images specified above.

5.3.2. The FINDER obeys

The FINDER then requested that the camera be positioned to obtain each of the three images designated. These images are shown in Figure 5.2. After each one was obtained, it called the MATCHER routines to find likely locations for the corresponding landmark in the image. The MATCHER used a Laplacian of size 3×7, the top 25 percentile of the edge points, the top 15 percentile of those based on gradient direction informativeness, and matched image points whose gradient directions were within 15 degrees of an object point's gradient direction. This resulted in the following:

Possible image locations		
Phone	**Cup**	**Plug**
phone1: (85,31) 100	cup1: (84,63) 63	plug1: (53,27) 25
	cup2: (60,66) 36	plug2: (39,30) 19
		plug3: (39,26) 19
		plug4: (53,32) 18
		plug5: (53,38) 17

Figure 5.3 shows these possible locations on the original images. These three lists of likely locations, with their respective confidences, were then checked against each other for angular consistency as described in Chapter 3. The resulting consistency graph is shown in Figure 5.4a. The graph was then pruned resulting in the removal of 3 out of the 8 nodes. Figure 5.4b shows the graph after pruning and Figure 5.5 shows the remaining possible locations overlaid on the images. The maximal complete subgraphs were three 3-cliques as follows:

Maximal complete subgraphs			
1)	(phone1(100)	cup1(63)	plug1(25))
2)	(phone1(100)	cup1(63)	plug4(18))
3)	(phone1(100)	cup1(63)	plug5(17))

These maximal complete subgraphs were then evaluated using the confidences to arrive at the best, which was the third one above. In Figure 5.6, the template points that were actually used in the matching are shown overlaid on their respective original images at the final locations. The bearings were then calculated for these image locations and the values passed back to the SELECTOR.

5.3.3. The SELECTOR triangulates

The SELECTOR then simply triangulated, using the landmark bearings with their known locations, to get the new position. The new uncertainty was then calculated (as described in Chapter 4). These results are displayed graphically in Figure 5.7.

6. Related Literature

Ballard introduced the generalization of the Hough transform [Ball81] and presented the parameterization used in this thesis in Ballard and Brown [Ball82]. Davis presented extensions to the generalized Hough transform for matching hierarchically organized point patterns or patterns of line-segments [Davi82]. Hakalahti, Harwood and Davis [Haka83] constrained matching based on local properties of contour points and Davis, Kitchen, Hu and Hwang [Davi83] used the generalized Hough transform to match patterns of blobs and ribbons.

The generalized Hough transform can also be used for recognizing three-dimensional objects in images. If our landmarks were represented as 3D models, and only very poor estimates were available of our position, then such 3D matching would be of interest. Silberberg [Silb84a,Silb84b] considered the special case where only position and rotation on the ground plane are unknown. This would be a good representation for our problem. See also Ballard and Sabah [Ball83], and Stockman & Esteva [Stoc84].

Research on autonomous vehicles has led to several papers describing other methods of automatic position determination. Fukui [Fuku81] presented a method using a specific diamond-shaped landmark and using distortion of the shape to determine the vehicle's relative position. This process was extended in Courtney and Aggrawal [Cour83]. Related papers under the general category of

camera calibration are Hung, Yeh & Mansbach [Hung83] and Hung, Yeh & Harwood [Hung84]. Methods using acoustic or laser ranging sensors are described in Crowley [Crow83] and Jarvis [Jarv83], respectively.

References

[Alle82a] - E. M. Allen, R. Trigg, and R. Wood, Maryland Artificial Intelligence Group Franz Lisp Environment, CS-TR-1266, Computer Science Department, University of Maryland (1982).

[Alle82b] - E. M. Allen, YAPS: Yet Another Production System, CS-TR-1146, Computer Science Department, University of Maryland (1982).

[Andr84] - F. P. Andresen, The Franz Lisp - C Interface, CAR-TR-1411, Computer Vision Laboratory, University of Maryland (1984).

[Ball81] - D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition 13*, 111-222 (1981).

[Ball82] - D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Eaglewood Cliffs, New Jersey, 482 (1982).

[Ball83] - D. H. Ballard and D. Sabbah, Viewer independent shape recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence 5*, 653-660 (1983).

[Cour83] - J. W. Courtney and J. K. Aggarwal, Robot guidance using computer vision, *Proceedings of the IEEE 71*, 57-62 (1983).

[Crow83] - J. L. Crowley, Dynamic world modeling and position estimation for an intelligent mobile robot, 7th International Conference on Pattern Recognition, Montreal, Canada (1984).

[Davi82] - L. S. Davis, Hierarchical generalized Hough transforms and line-segment based generalized Hough transforms, *Pattern Recognition 15*, 277-285 (1982).

[Davi83] - L. S. Davis, Image matching using generalized Hough transforms, CAR-TR-27, Computer Vision Laboratory, University of Maryland (1983).

[Davi85] - L. S. Davis, Visual algorithms for autonomous navigation, submitted to IEEE International Conference on Robotics and Automation (1985).

[Fuku81] - I. Fukui, TV image processing to determine the position of a robot vehicle, *Pattern Recognition 14*, 101-109 (1981).

[Hara79] - R. Haralick and L. Shapiro, The consistent labeling problem, *IEEE Trans. on Pattern Analysis and Machine Intelligence 1*, 173-183 (1979).

[Harw84] - D. Harwood, M. Subbarao, H. Hakalahti, and L. S. Davis, A new class of edge-preserving smoothing filters, CAR-TR-59, Computer Vision Laboratory, University of Maryland (1984).

[Hung83] - Y. Hung, P. Yeh, and P. Mansbach, Calibration of a structured light vision system, CAR-TR-29, Computer Vision Laboratory, University of Maryland (1983).

[Hung84] - Y. Hung, P. Yeh, and D. Harwood, Passive ranging to known planar point sets, CAR-TR-65, Computer Vision Laboratory, University of Maryland (1984).

[Jarv83] - R. A. Jarvis, A laser time-of-flight range scanner for robotic vision, *IEEE Trans. on Pattern Analysis and Machine Intelligence 5*, 505-512 (1983).

[Mora83] - H. P. Moravec, The Stanford Cart and the CMU Rover, *Proceedings of the IEEE 71*, 872-884 (1983).

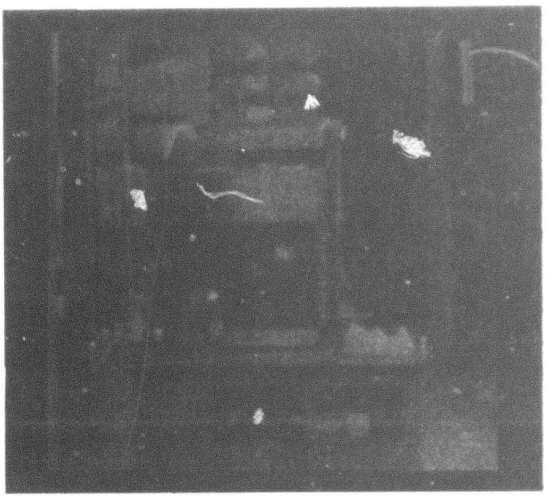
[Silb84a] - T. M. Silberberg, D. Harwood, and L. S. Davis, Object recognition using oriented model points, First Conference on Artificial Intelligence Applications, Denver, Colorado (1984).

[Silb84b] - T. M. Silberberg, L. S. Davis, and D. Harwood, An iterative Hough

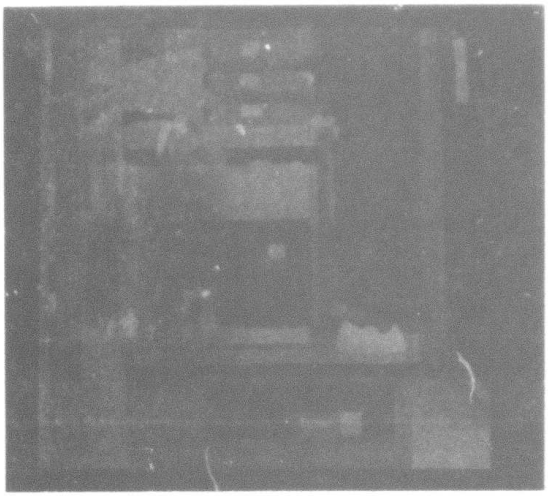
procedure for three-dimensional object recognition, *Pattern Recognition*, to appear.

[Stoc84] - G. Stockman and J. C. Esteva, Use of geometrical constraints and clustering to determine 3D object pose, *Proceedings: ICPR-7*, Montreal, Canada, 742-744 (1984).

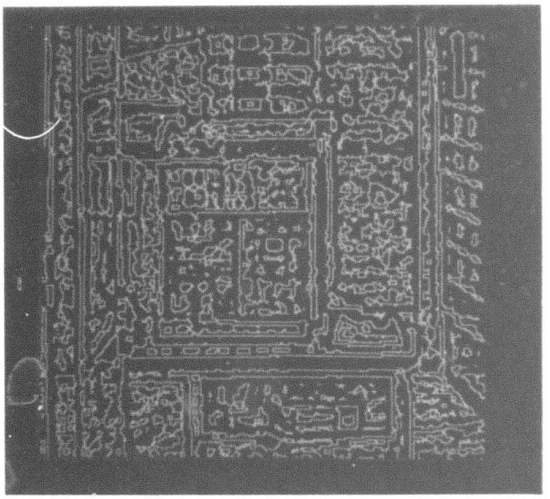
[Waxm85] - A. Waxman, J. LeMoigne, and B. Srinivasan, Visual navigation of roadways, submitted to IEEE International Conference on Robotics and Automation (1985).



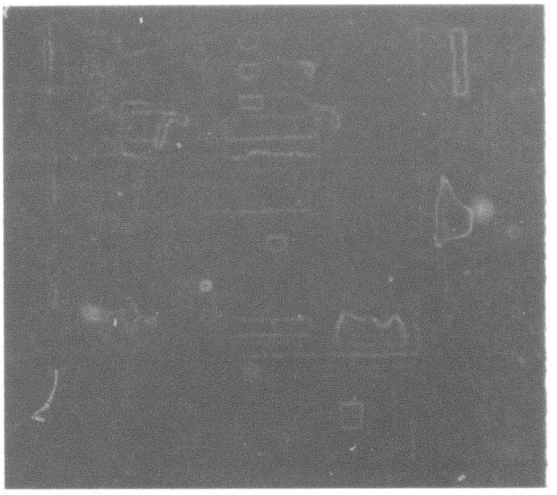
a) The original image



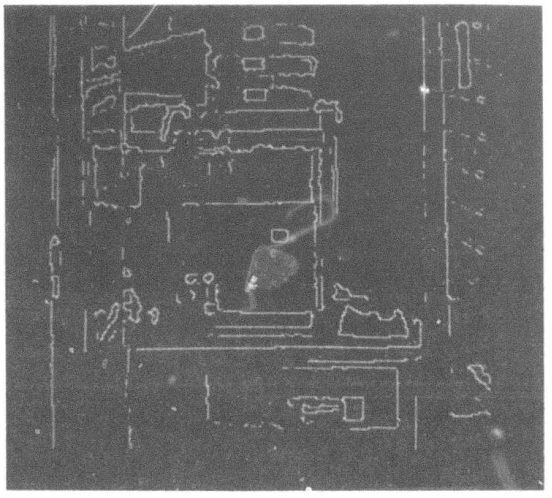
b) Symmetric K-nearest neighbor smoothing



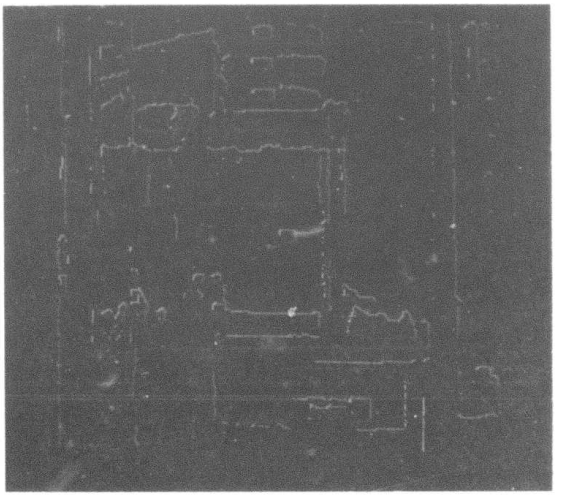
c) Zero-crossings of the Laplacian of a Gaussian



d) Symmetric contrast operator applied

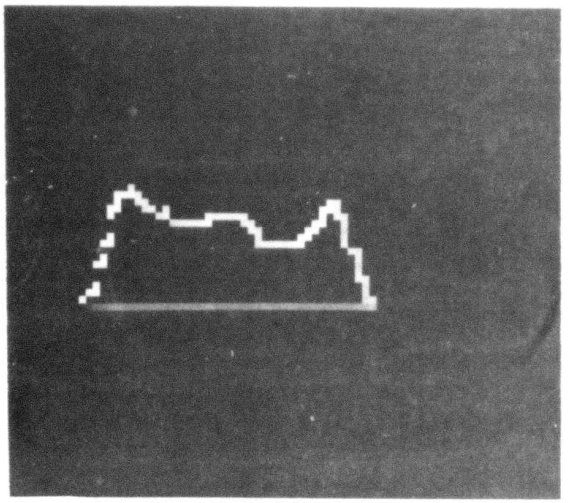


e) Edge points picture

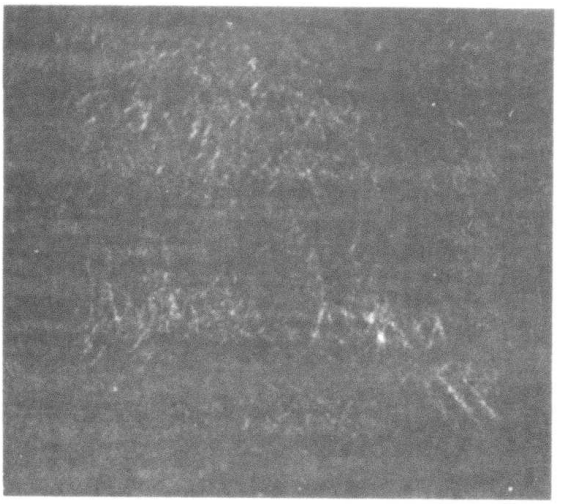


f) Gradient directions at edge points

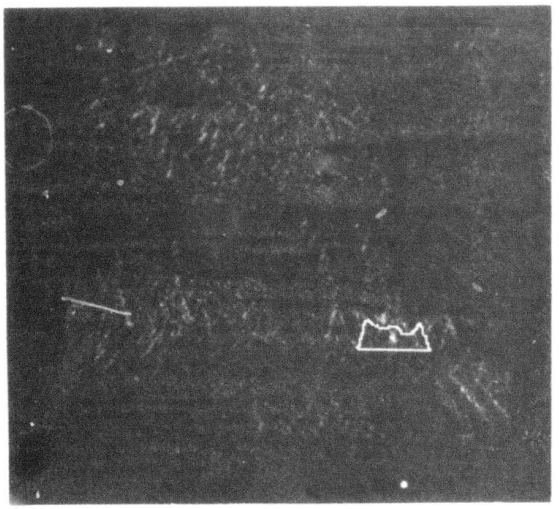
Figure 2.1 Follows the MATCHER through each stage of image processing for locating a tape dispenser near a coffee maker



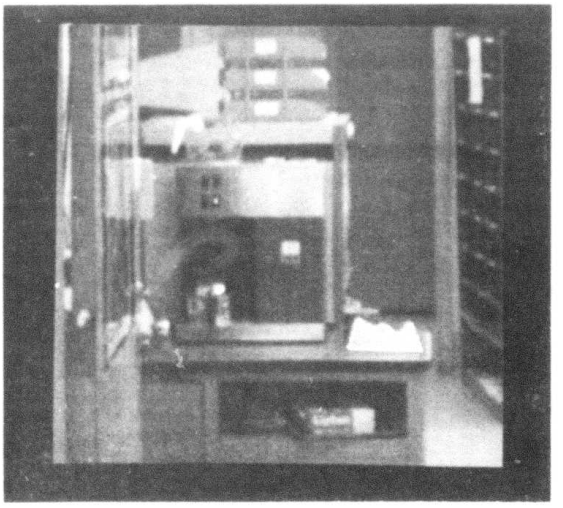
g) Template displayed as an image



h) Hough space

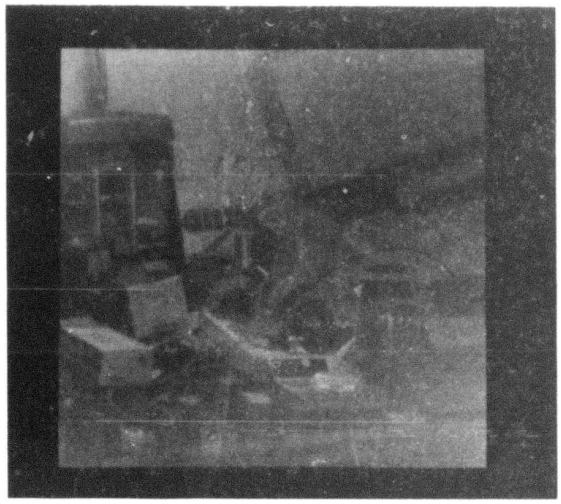


i) The solution overlaid on the Hough space

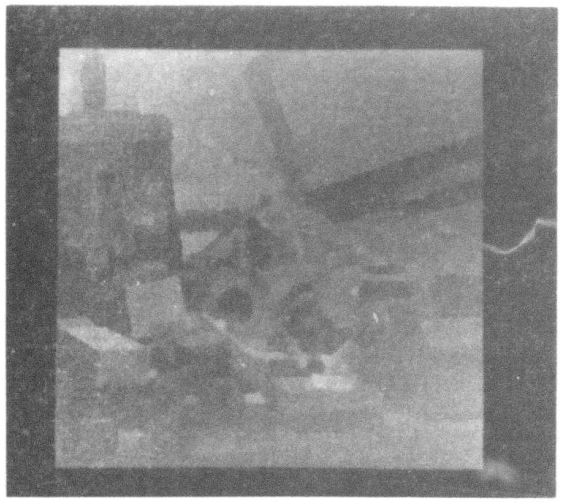


j) The solution overlaid on the original image

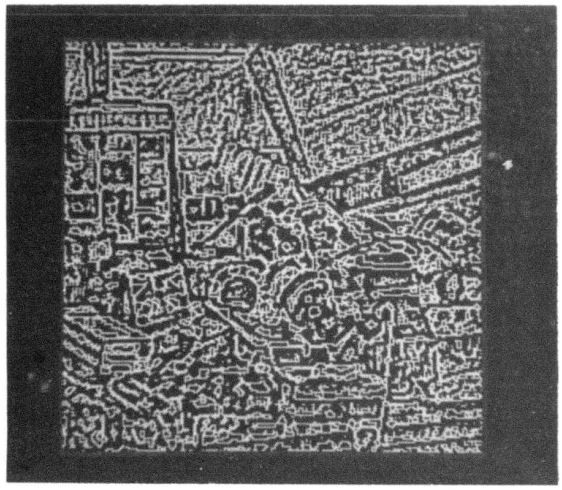
Figure 2.1 (continued)



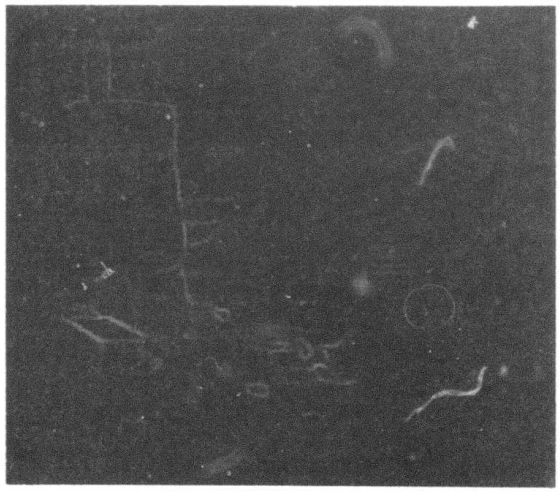
a) The original image



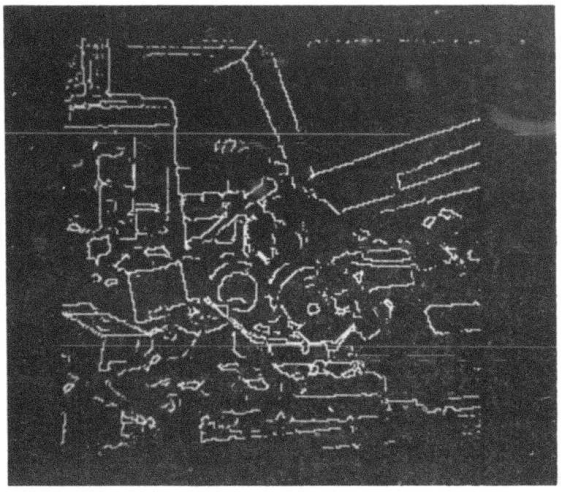
b) Symmetric K-nearest neighbor smoothing



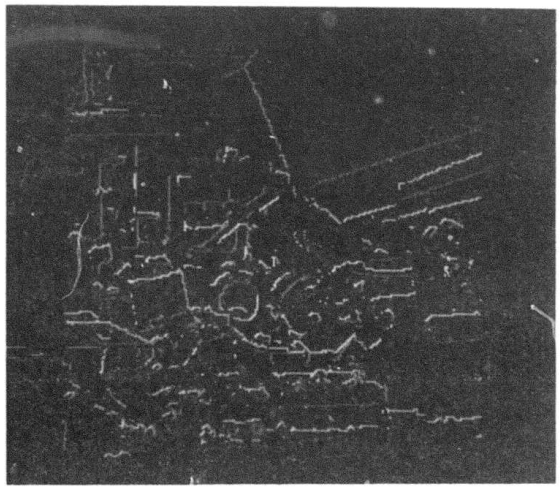
c) Zero-crossings of the Laplacian of a Gaussian



d) Symmetric contrast operator applied

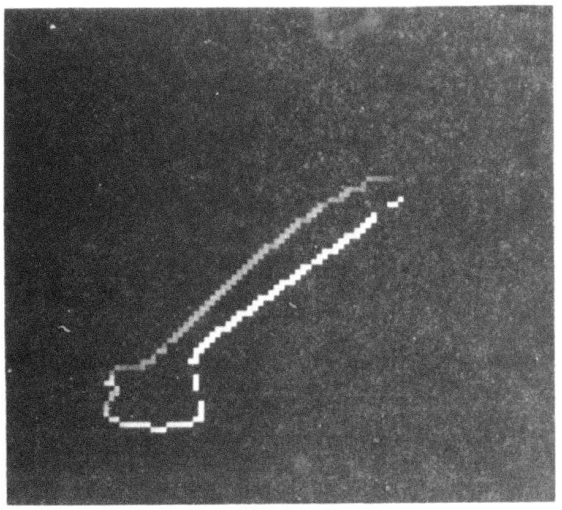


e) Edge points picture



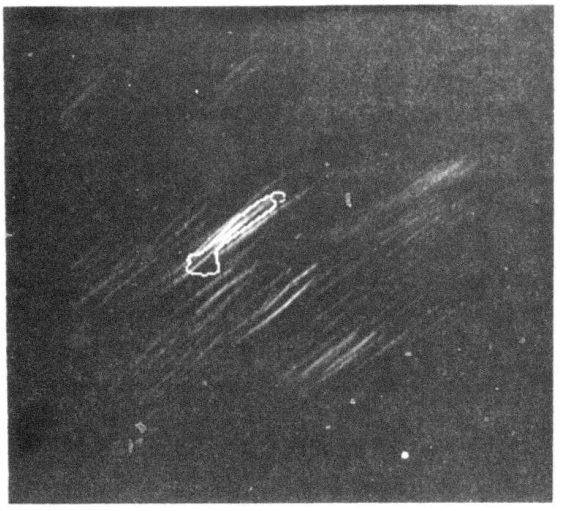
f) Gradient directions at edge points

Figure 2.2 The same as Figure 2.1 except for locating a brush



g) Template displayed as an image

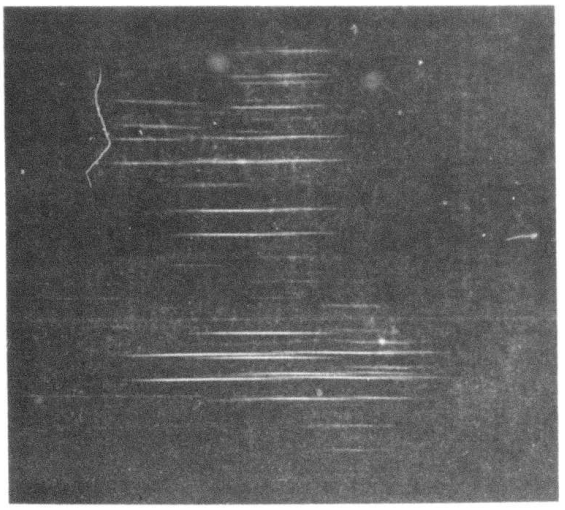
h) Hough space



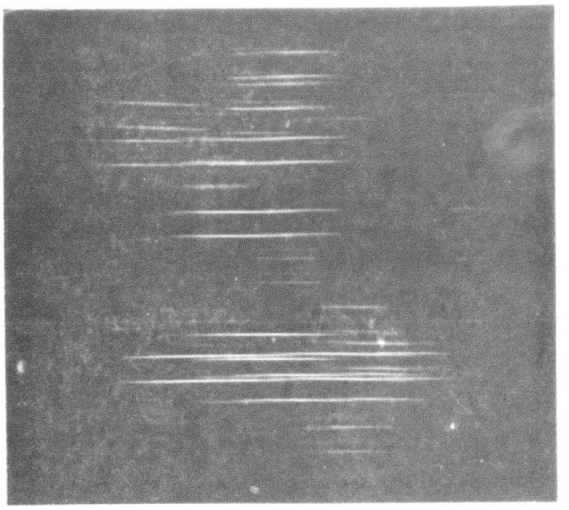
i) The solution overlaid on the Hough space

j) The solution overlaid on the original image

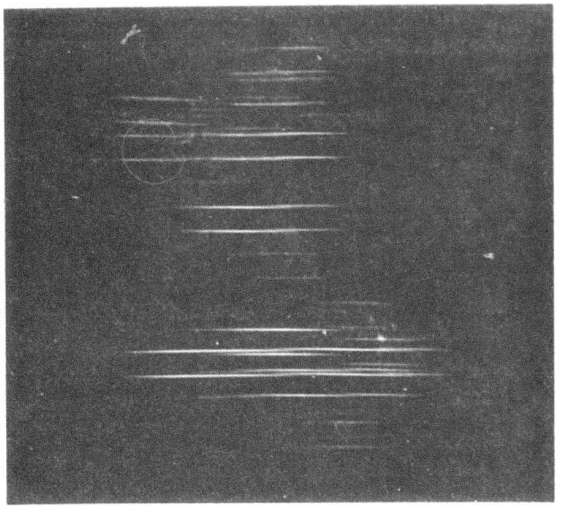
Figure 2.2 (continued)



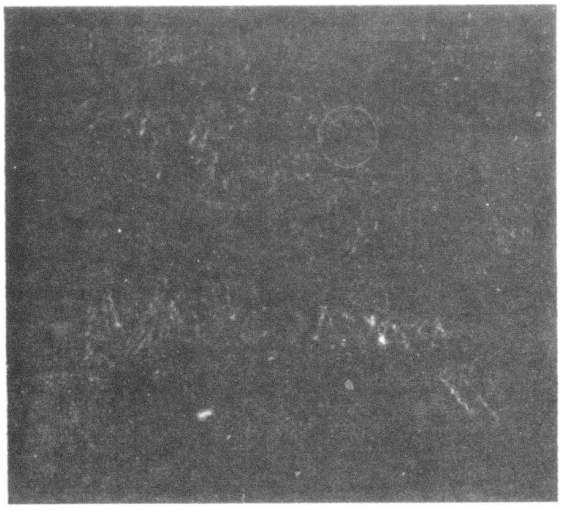
a) Using 100% of the gradient directions



b) Using 50% of the gradient directions



c) Using 30% of the gradient directions

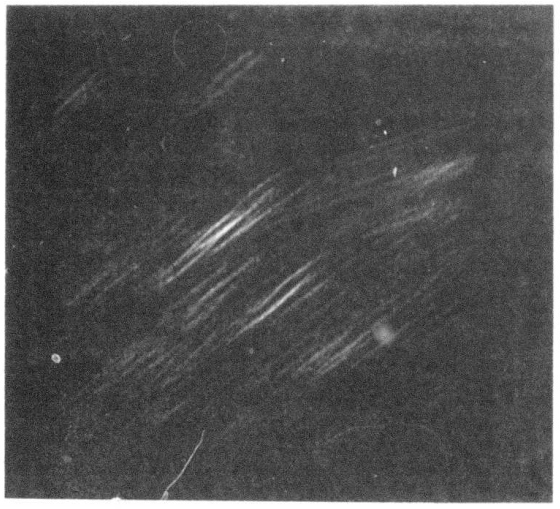


d) Using 15% of the gradient directions

Figure 2.3 Results of gradient direction informativeness (GDI) filtering for the tape dispenser

a) Using 100% of the gradient directions

b) Using 50% of the gradient directions

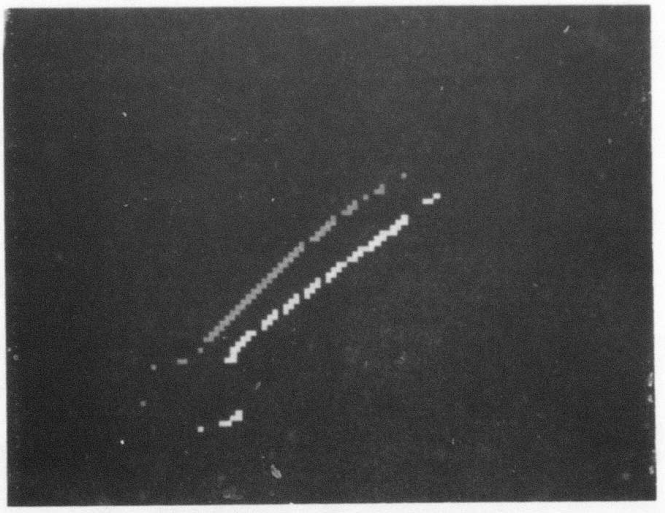


c) Using 30% of the gradient directions

d) Using 15% of the gradient directions

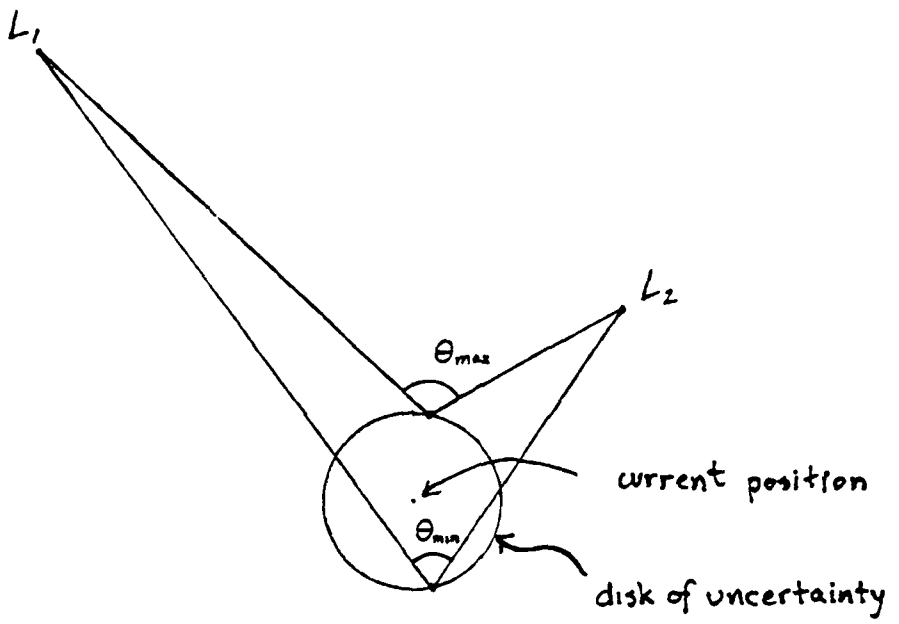
Figure 2.4 Same as Figure 2.3 except done for the brush

a) The tape template

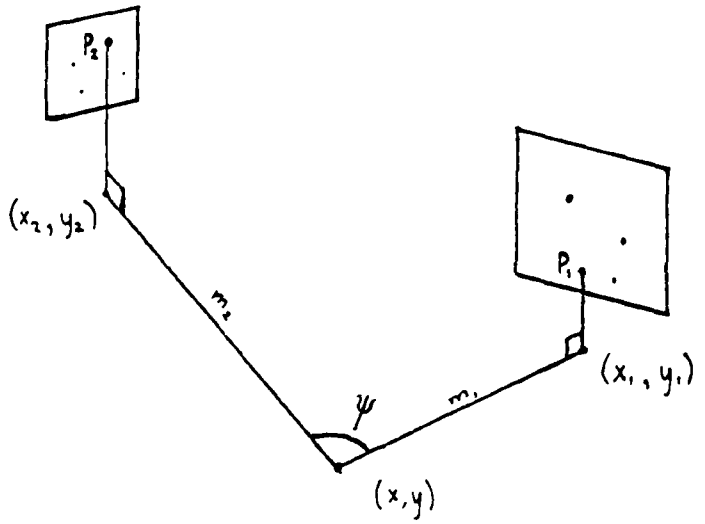


b) The brush template

Figure 2.5 The templates after GDI filtering



a) Determining the range of angular differences



b) The angular difference between peaks

Figure 3.1 Consistency between peaks

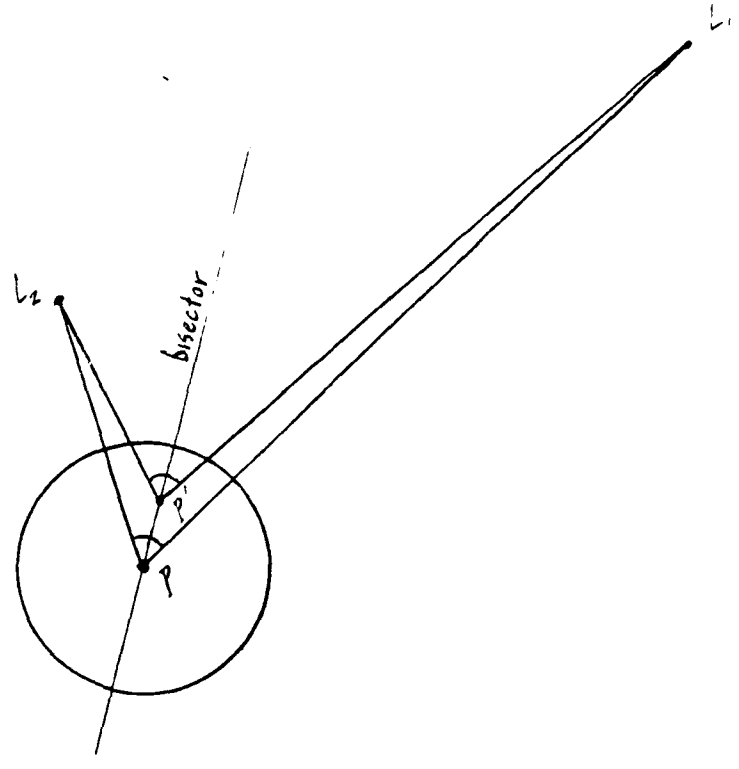


Figure 3.2 Proof of a circumferential solution

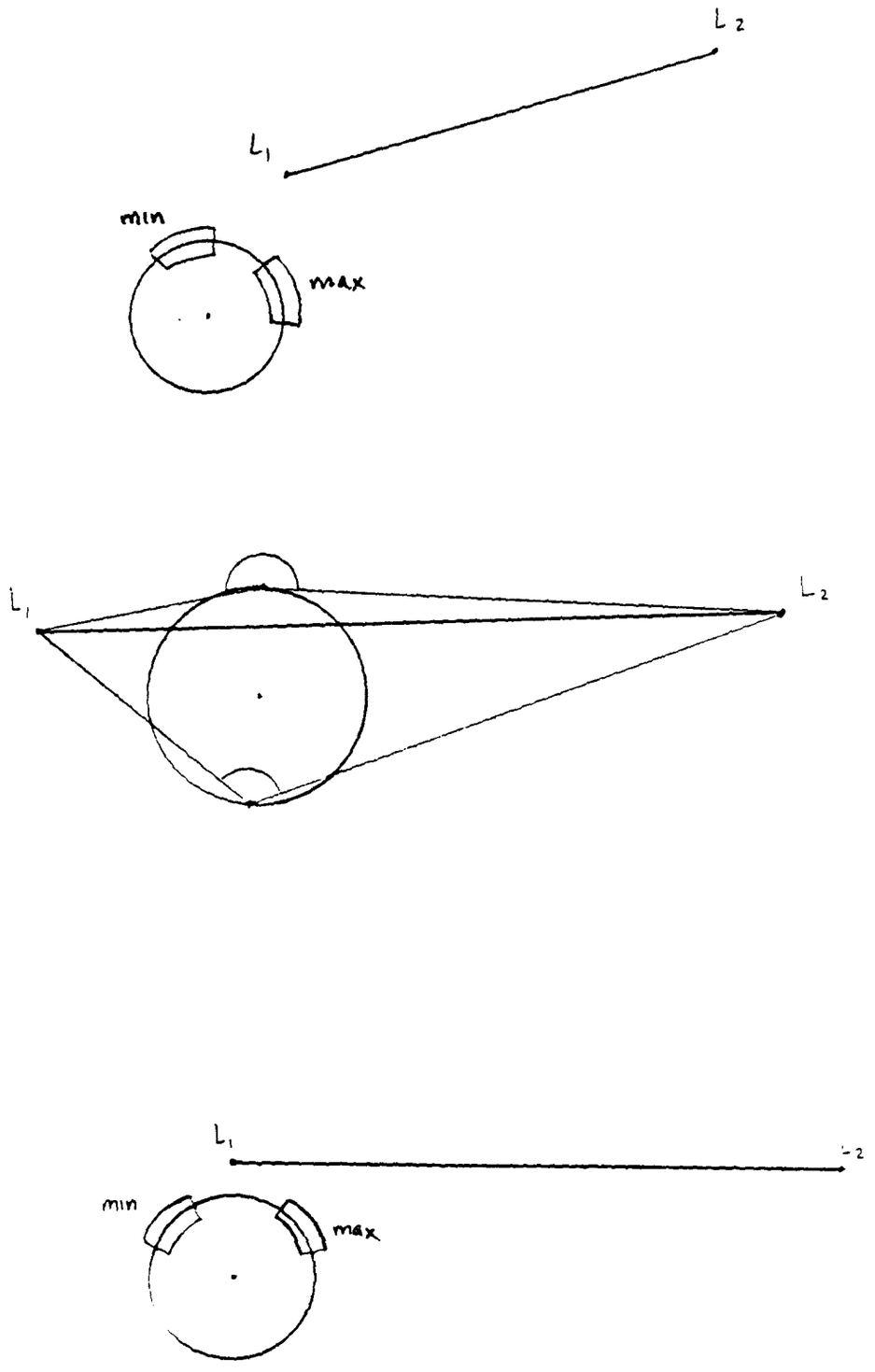
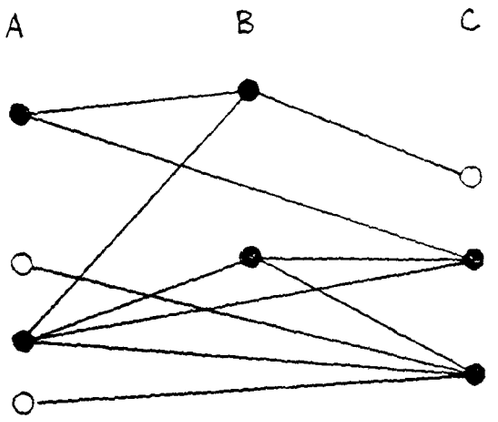
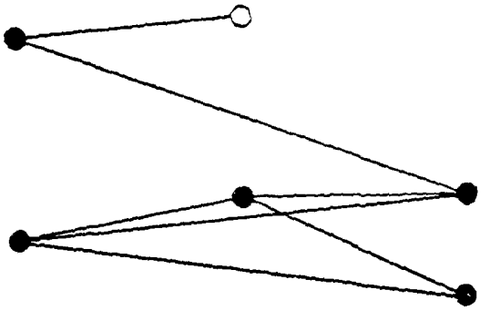


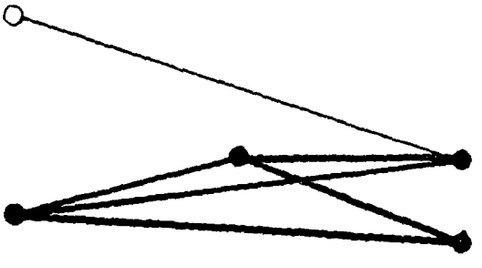
Figure 3.3 Counter-examples to some simple non-solutions



a) Original graph



b) After one iteration of **k-star**



c) After two iterations of **k-star**;
bold arcs are after three iterations

Figure 3.4 Graph pruning with **k-stars** and **k-fans**
O: will be deleted in the next iteration

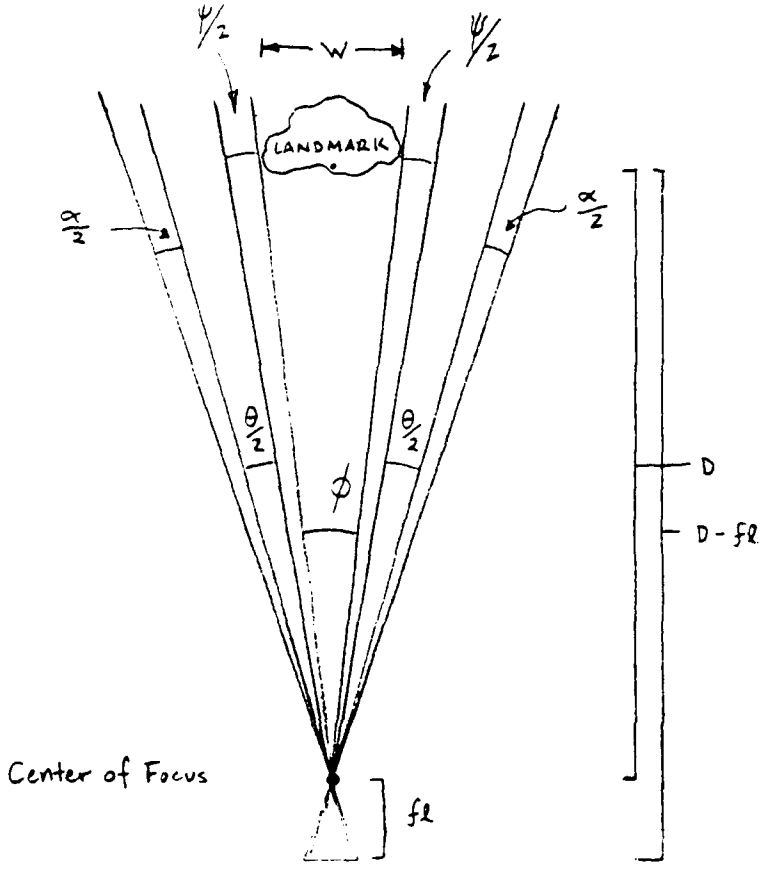


Figure 4.1 Determining minimum-FOV

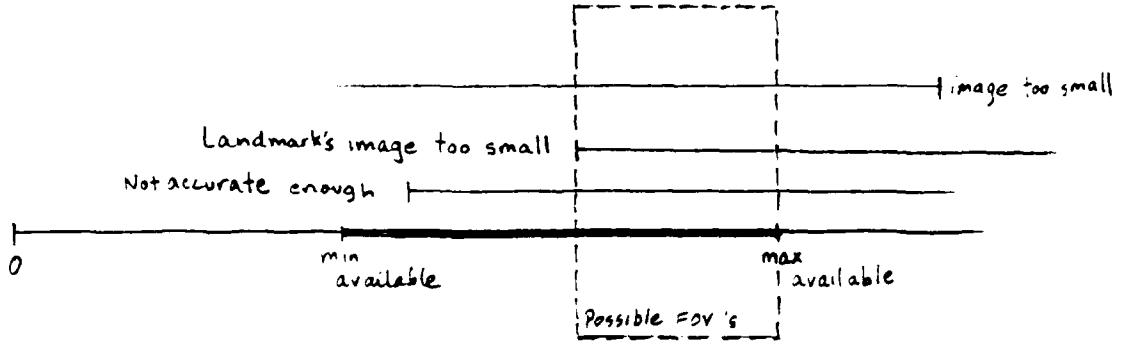


Figure 4.2 Constraint checking

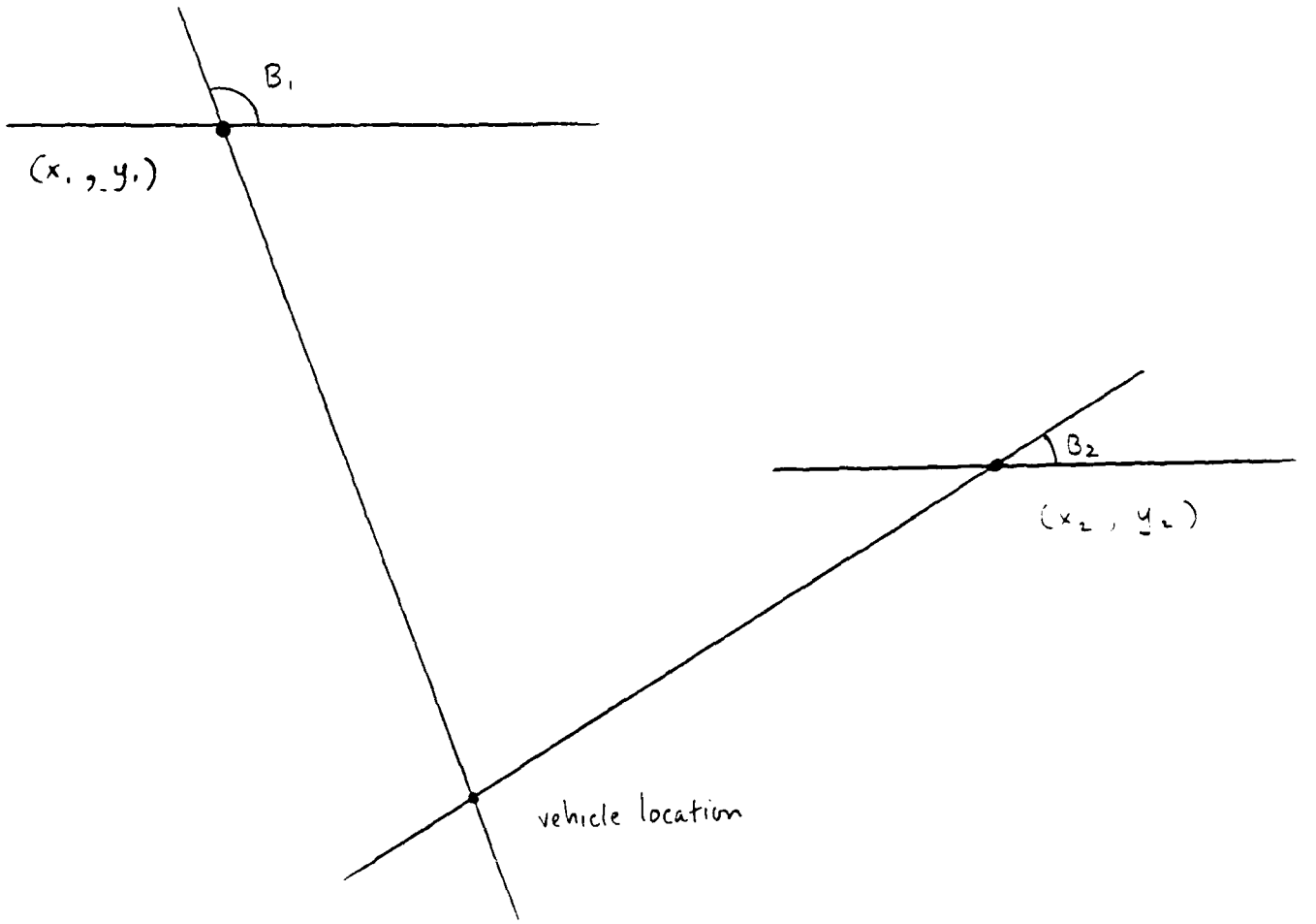
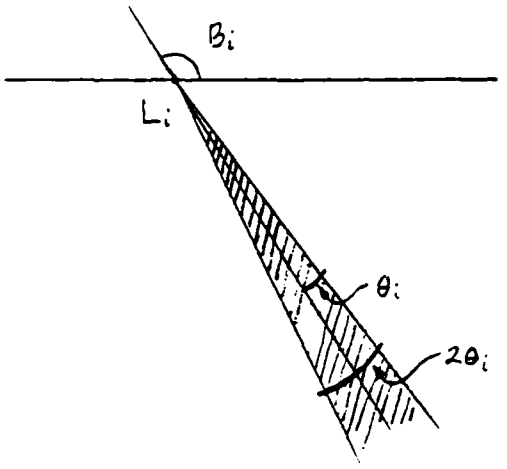
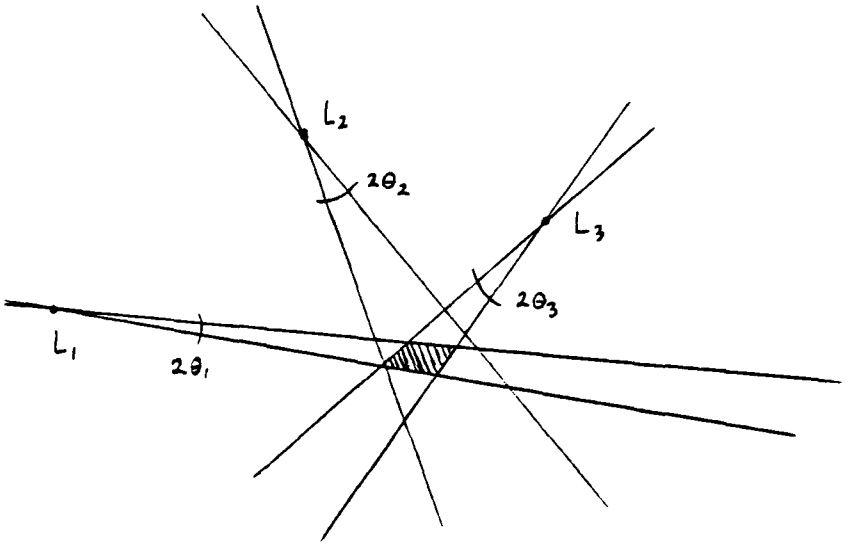


Figure 4.3 Triangulation

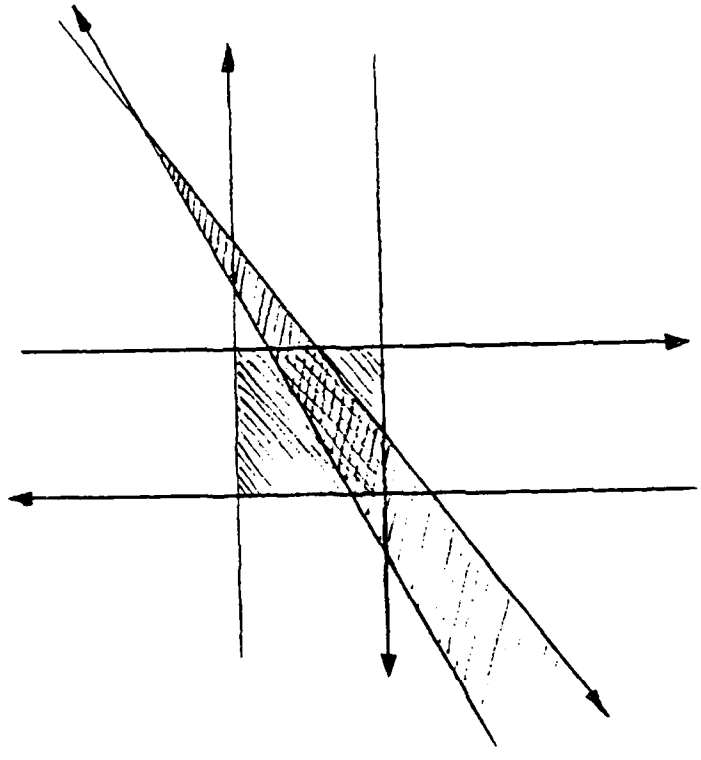


a) Possible positions given one landmark's bearing

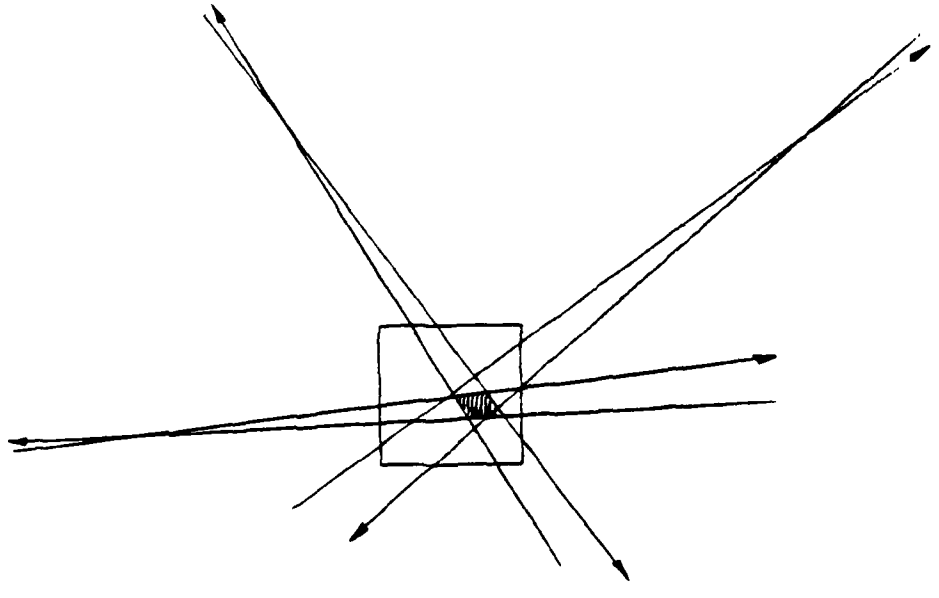


b) Possible positions with three landmarks' bearings

Figure 4.4 Determining position uncertainty



a) With one wedge and a square



b) With several wedges and a square

Figure 4.5 Halfplanes

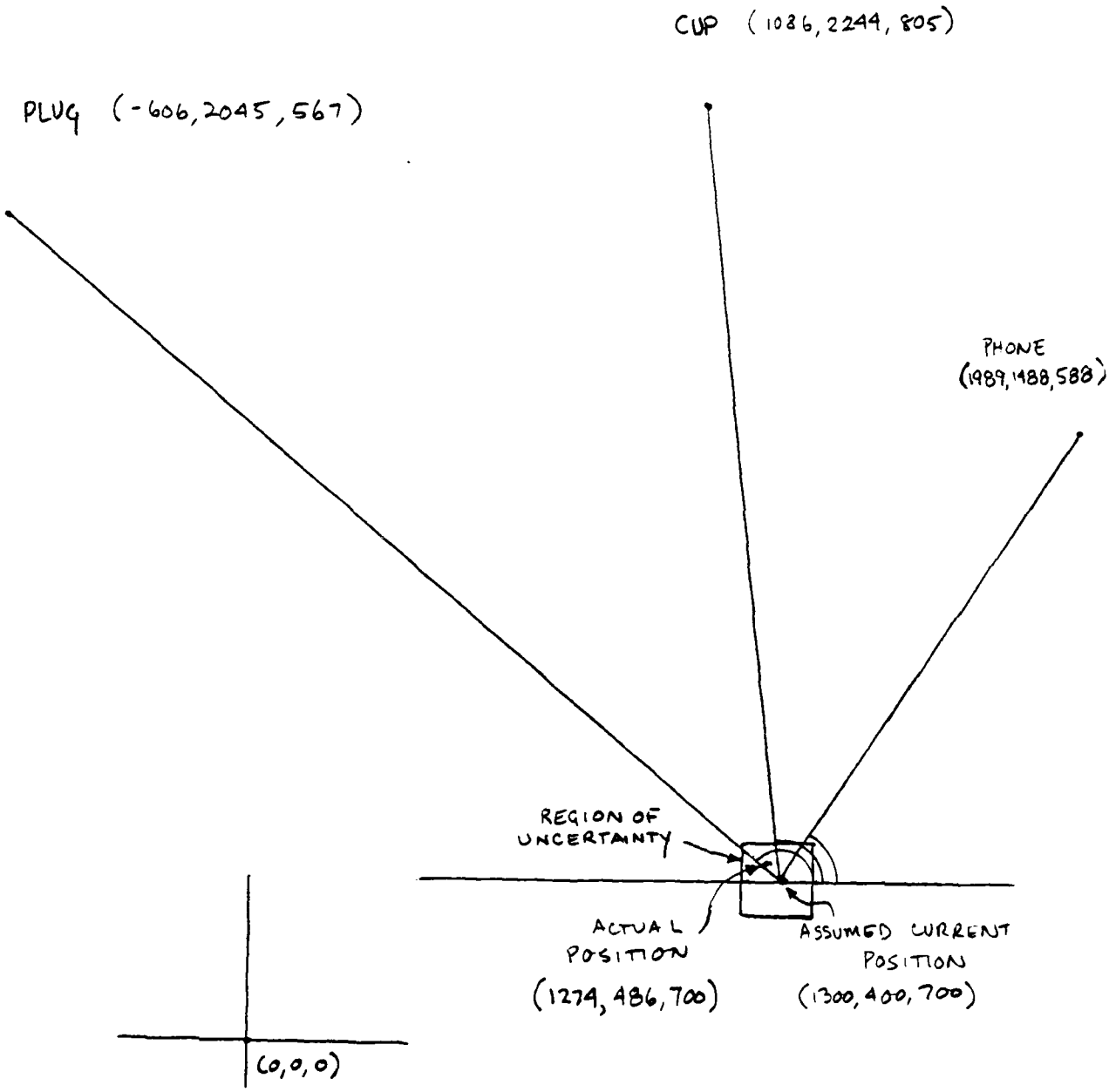
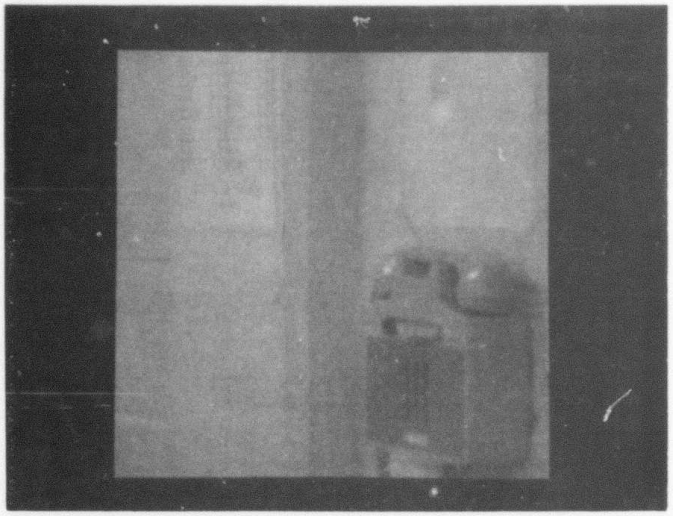
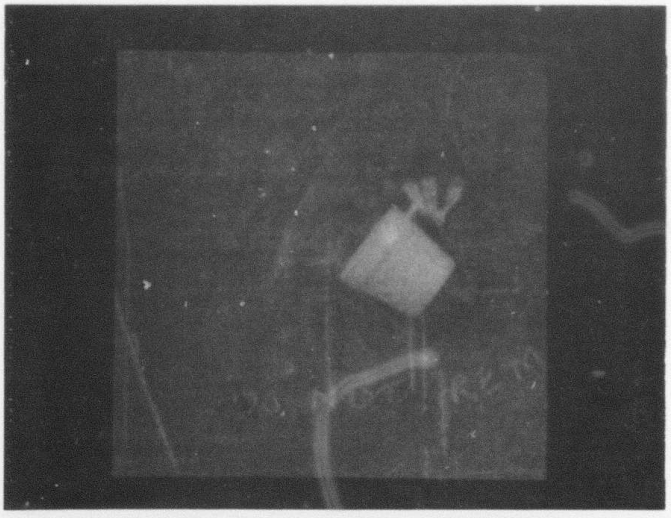


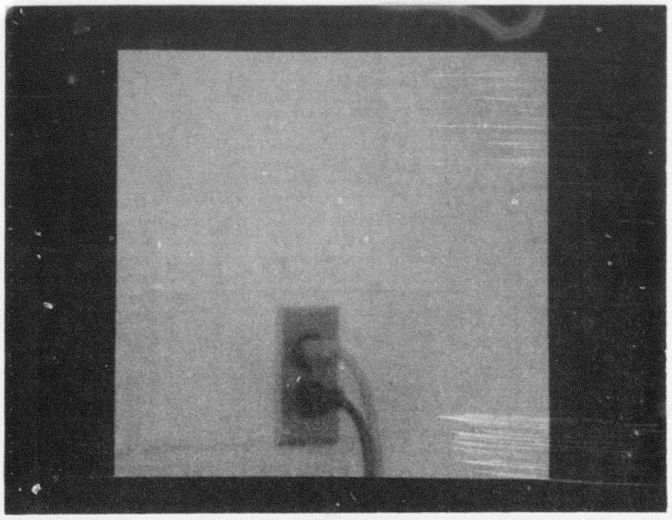
Figure 5.1 Layout of the experiment room



a) phone

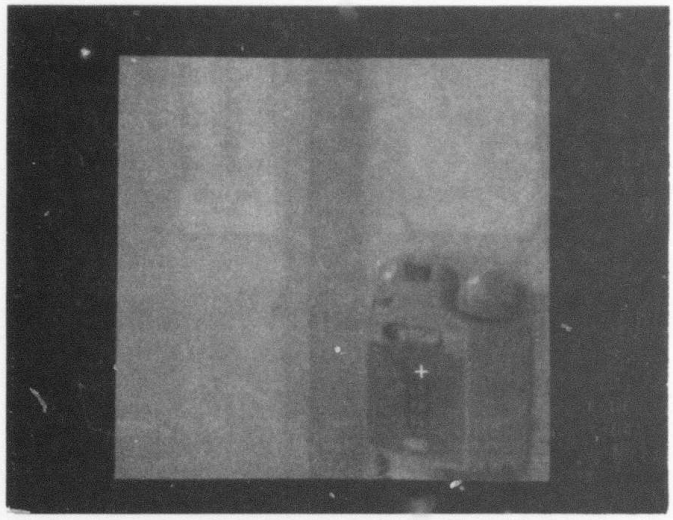


b) cup

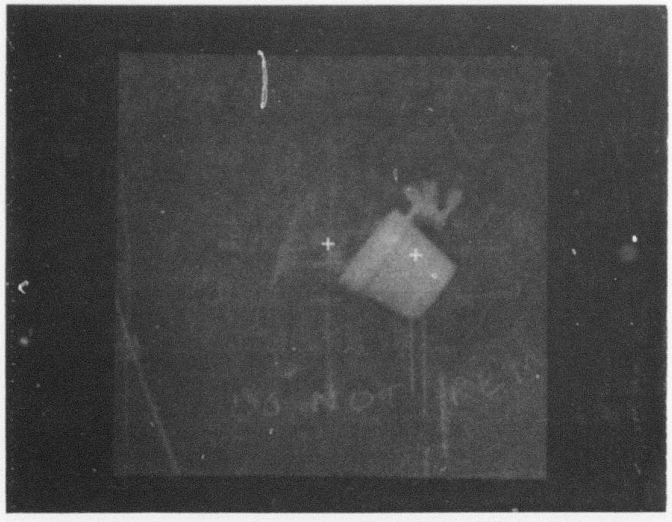


c) plug

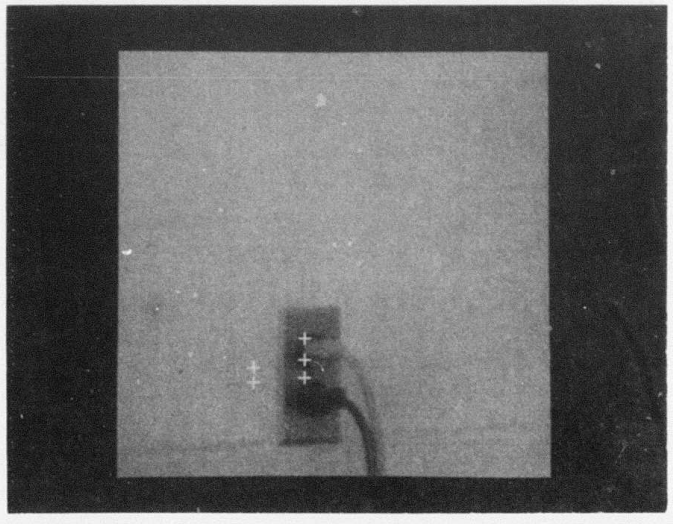
Figure 5.2 Original images for the landmarks



a) phone

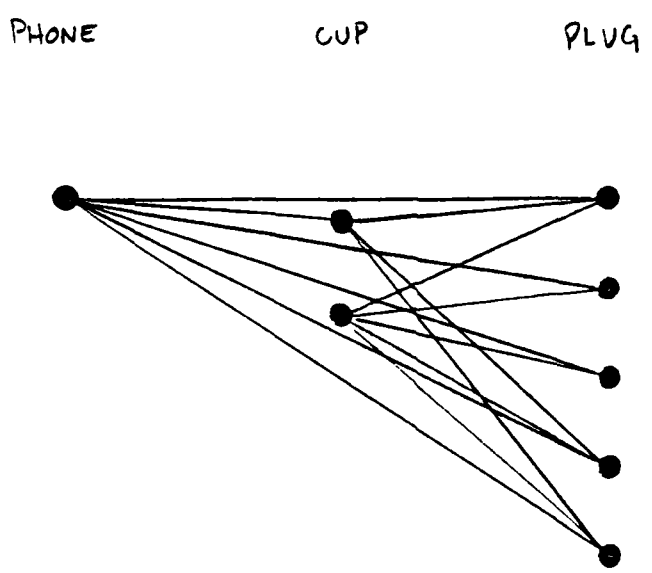


b) cup

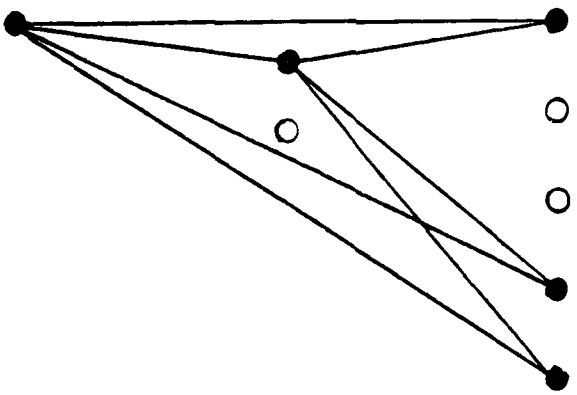


c) plug

Figure 5.3 Possible locations for the landmarks

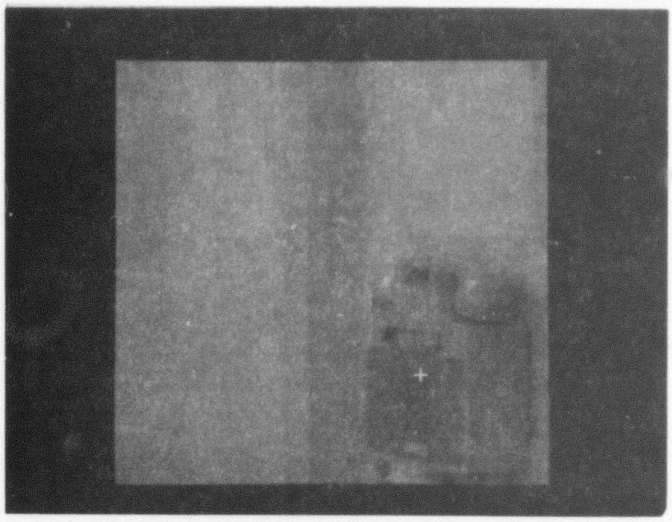


a) Original graph

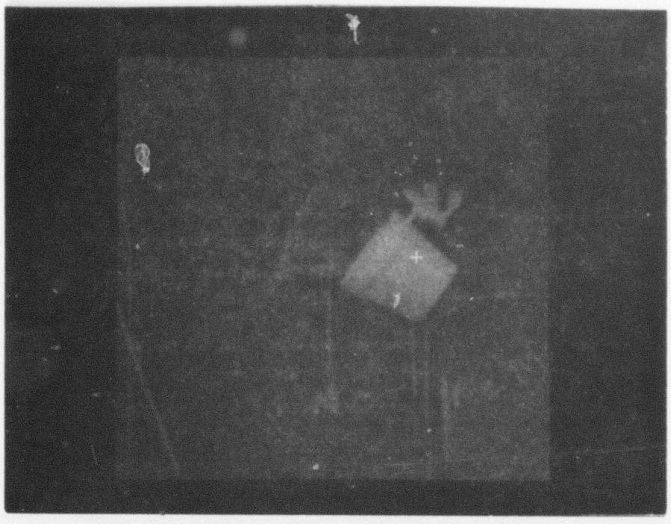


b) After pruning

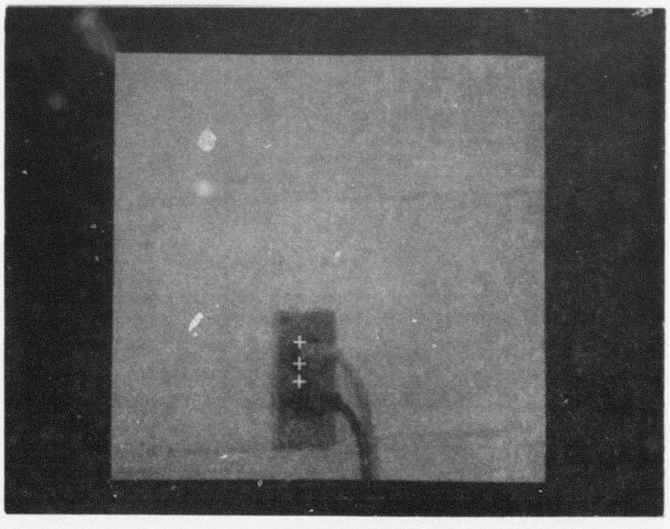
Figure 5.4 Pruning of the consistency graph
O: has been pruned



a) phone

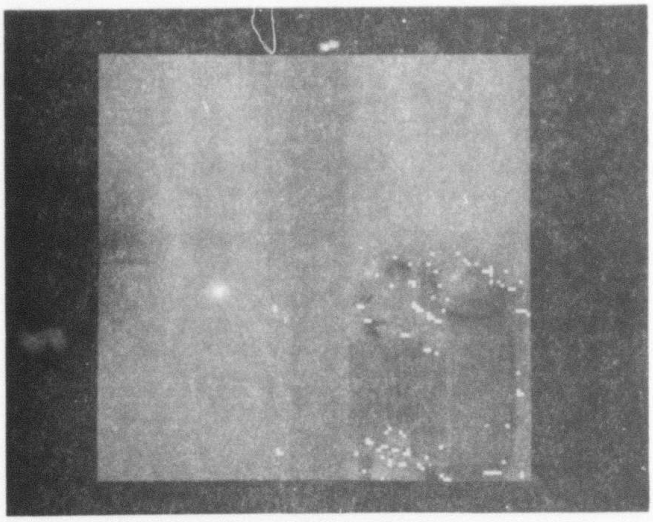


b) cup

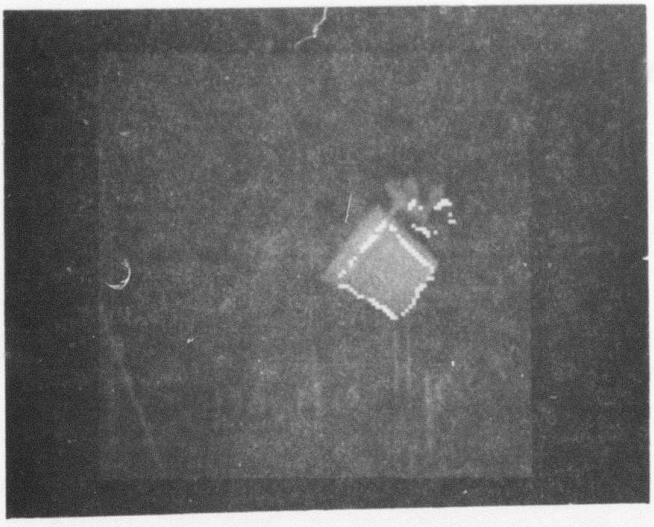


c) plug

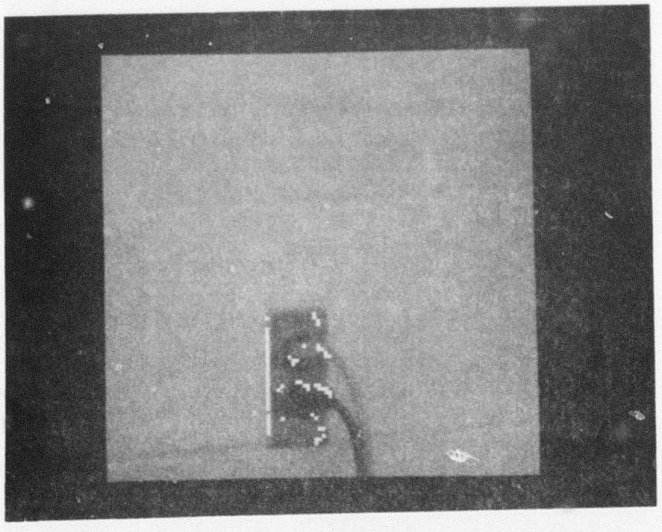
Figure 5.5 Results of pruning shown on images



a) phone



b) cup



c) plug

Figure 5.6 Models overlaid to show final match

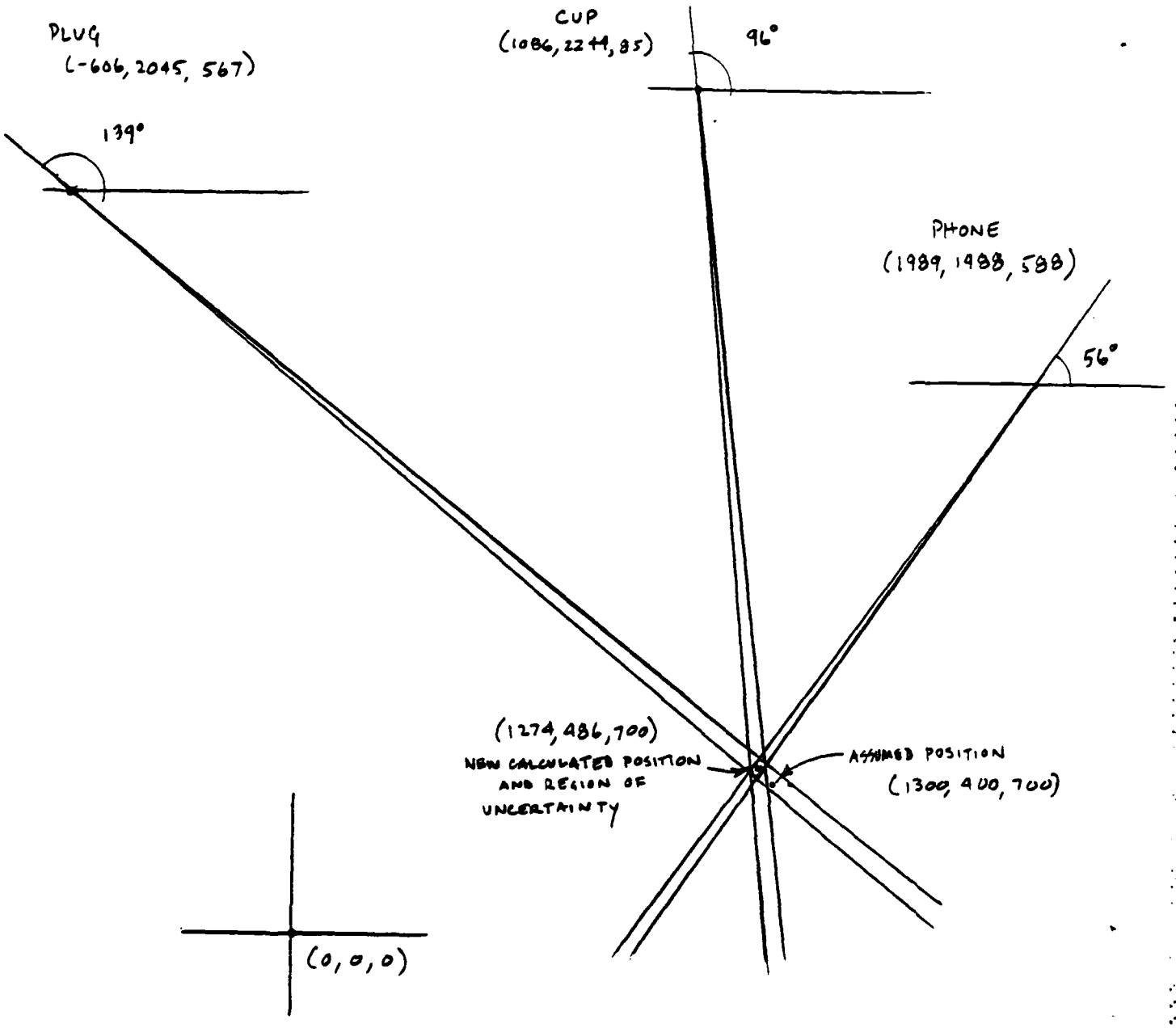


Figure 5.7 Results showing new position and uncertainty

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A156607	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) VISUAL POSITION DETERMINATION		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER CAR-TR-100; CSC-TR-1458
7. AUTHOR(s) Frederick P. Andresen Larry S. Davis		8. CONTRACT OR GRANT NUMBER(s) DAAK70-83-K-0018
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Automation Research University of Maryland College Park, MD 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency U.S. Army Night Vision Laboratory Ft. Belvoir, VA 22060		12. REPORT DATE November 1984 13. NUMBER OF PAGES 64
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Autonomous vehicle navigation
Position determination
Landmark location

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a system by which an autonomous land vehicle might improve its estimate of its current position. This system selects visible landmarks from a database of knowledge about its environment and controls a camera's direction and focal length to obtain images of these landmarks. The landmarks are then located in the images using a modified version of the generalized Hough transform and their locations are used to triangulate to obtain the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

new estimate of vehicle position and position uncertainty.