VISUAL ALGORITHMS FOR AUTONOMOUS NAVIGATION

Fred P. Andresen
Larry S. Davis
Roger D. Eastman
Subbarao Kambhampati

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, Maryland 20742

ABSTRACT

The Computer Vision Laboratory at the University of Maryland is designing and developing a vision system for autonomous ground navigation. Our approach to visual navigation segments the task into three levels called long range, intermediate range and short range navigation. At the long range, one would first generate a plan for the day's outing, identifying the starting location, the goal, and a low resolution path for moving from the start to the goal. From time to time, during the course of the outing, one may want to establish his position with respect to the long range plan. This could be accomplished by visually identifying landmarks of known location, and then triangulating to determine current position. We describe a vision system for position determination that we have developed as part of this project. At the intermediate range, one would look ahead to determine generally safe directions of travel called *corridors of free space*. Short range navigation is the process that, based on a detailed topographic analysis of one's immediate environment, enables us to safely navigate around obstacles in the current corridor of free space along a *track of safe passage*. We describe a quadtree based path planning algorithm which could serve as the basis for identifying such tracks of safe passage.

1. INTRODUCTION

The Computer Vision Laboratory at the University of Maryland is designing and developing a vision system for autonomous ground navigation. This vision system will be integrated by Martin Marrietta Corp., Denver Colorado, into a navigation system that will autonomously drive a ground vehicle over a network of roads at speeds of up to 10 kilometers/hour. The vehicle will be equipped with a variety of sensors, including TV sensors and an active ranging sensor, as well as a sophisticated inertial navigation system. The vehicle will be driven, initially, by a suite of computers including a VAX 11/750 and a VICOM image processor. The project is supported under DARPA's Strategic Computing Program, and will involve the collaboration of many industrial and university research laboratories.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under contract DAAK70-83-0018 (DARPA order 3206) as well as the U. S. Army Engineering Topographic Laboratory under Contract DAAK70-81-C-0059 is gratefully acknowledged.

The vision research is focused by a set of project milestones that involve incrementally extending the capabilities of the vehicle to deal with more and more complex situations. The first milestone requires that the vehicle be able to navigate a single, obstacle free road, with no intersections, at speeds on the order of 5 kilometers/hour. A demonstration of this milestone is initially scheduled for late summer 1985. Subsequent milestones include navigating on a road with widely spaced obstacles, and navigating a network of roads having different compositions and quality.

It is expected that many of these milestones might not be achievable at the desired speeds using commercially available computers. Therefore, a large part of the Strategic Computing Program is devoted to fostering the development of appropriate computer architectures for the machine intelligence tasks associated with navigation. Prototype machines developed under the program will first be introduced into the AI and Vision Laboratories studying navigation, and eventually integrated into a test vehicle.

Our approach to visual navigation segments the task into three levels, called long range, intermediate range and short range navigation. The general flow of control between levels is that goals flow from levels of greater abstraction to levels of lesser abstraction (long - intermediate - short) while status information concerning the achievement of these goals flows in the opposite direction. Each level of navigation maintains a map of (some subset) of the environment to be navigated, with the map representations becoming more detailed as one moves from long range down to short range navigation. Specific sensors and visual capabilities are associated with each level of navigation; these sensors and capabilities function to maintain the correctness of the map representation at that level.

The decomposition of the navigation problem into these three levels is intuitively motivated by the types of navigation tasks that a human might perform in moving

CH2152-7/85/0000/0856$01.00

through a generally unfamiliar environment with the aid of a low resolution map (of the sort that one might obtain upon entrance to a national park). At the so-called long range, one would first generate a plan for the day's outing, identifying the starting location, the goal, and a low resolution path for moving from the start to the goal. This path would be chosen based on gross considerations of the terrain to be crossed and the capabilities of the "vehicle". From time to time, during the course of the outing, one may want to establish his position with respect to the long range plan. This could be accomplished by visually identifying landmarks of known location, and then triangulating to determine current position. Section 2 of this paper describes a vision system for position determination that we have developed as part of this project. A more detailed description is available in Andresen and Davis[1].

At the intermediate range, one would look ahead to determine generally safe directions of travel. This would involve assessing the nature of the terrain in one's immediate visual environment and identifying those portions of that environment through which it is feasible to move. We refer to those navigable portions of the environment as *corridors of free space.* In the context of the current set of project milestones, corridors of free space naturally correspond to segments of road in the vehicle's field of view. We have developed a system of algorithms that can, in many cases, detect and follow roads in sequences of imagery. They are based principally on an analysis of dominant linear features extracted from the individual frames in the image sequence. These algorithms are currently being evaluated on a large database of images acquired at Martin Marrietta's test site in Denver. The details of the algorithm design are included in Waxman et. al[2].

Finally, short range navigation is the process that, based on a detailed topographic analysis of one's immediate environment, enables us to determine safe positions for foot placement, and to navigate around obstacles in the current corridor of free space. Human navigators probably use stereo and motion vision to derive the three dimensional information needed to solve these problems. The vehicle, which is a wheeled vehicle as opposed to a legged vehicle, will be equipped with a laser range sensor, currently being constructed at the Environmental Research Institute of Michigan (Zuk and Dell'eva[3]), which is capable of acquiring two 100 x 100 arrays of range data per second. This range data can, for example, be converted into a simple array in which regular patches of terrain are classified as either "navigable" or "non-navigable" and then appropriate path planning algorithms can determine what we refer to as a ***track of safe passage*** and generate the corresponding motion control algorithms to the pilot of the vehicle to negotiate that track. In Section 3 of this paper we describe a quadtree based path planning algorithm which could serve as the basis for identifying such tracks of safe passage. A more detailed description with extensive examples is presented in Kambhampati and Davis[4].

2. LANDMARK BASED VEHICLE POSITIONING

An autonomous vehicle must have the capability of computing its current position for a variety of reasons - e.g., to be able to anticipate, on the basis of available cartographic data, important events that are likely to occur (road intersections, bends, changes in terrain, etc.) and to use these expectations to guide its sensory processing. The position of the vehicle can be determined using many different approaches. The vehicle will have an onboard inertial navigation system. Although such systems are very accurate, they do suffer from some degree of drift which can build up to substantial errors over long distances. The vehicle might also have access to a satellite positioning system such as the Global Positioning System. While such a system does not suffer from drift, its accuracy is not as high as an inertial system; furthermore, since such systems depend on components that are external to the vehicle, there is no guarantee that they will be available when needed.

It is also possible to compute the position of the vehicle by visually identifying objects of known scale, location and appearance and then simply triangulating to determine the vehicle's position. Ordinarily, one has some initial estimate of the vehicle position (perhaps from the inertial system) and one wishes to improve the accuracy of that estimate using visual landmark recognition.

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.

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.

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 positions 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 positions for these landmarks in the images. It then eliminates possible positions for individual landmarks which are not consistent with the possible positions found for other landmarks. The FINDER then

evaluates the remaining possible positions to determine the actual positions 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 positions in such images for subsets of those landmarks. With these positions, 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.

Section 2.1 - 2.3 describe the MATCHER, FINDER, and SELECTOR, repectively.

2.1. The MATCHER

A generalized Hough transform [5] is employed to locate landmarks of known image orientation and scale. The landmarks are represented by lists of boundary points; these points are then 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 discussed in more detail below.

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 $M \times N$ 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 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, we developed a measure of gradient direction informativeness (GDI) to rate the gradient directions. 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.

2.2. The FINDER

This section 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 positions for the landmarks in their respective images. Since the search for any specific landmark may result in several possible image positions (which we will refer to from now on as "peaks") for that landmark (at most one of which can be correct), a simple geometric constraint propagation algorithm is employed to eliminate many of the false peaks.

The geometric constraint propagation algorithm considers possible peaks for a pair of landmarks and determines if they could both be the correct peaks for their respective landmarks. Two possible 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.

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 positional 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.

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. 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 can be found in [1].

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 practical, 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 [6].

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**. 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 peaks.

2.3. The SELECTOR

This section 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 abilities to determine both the ease of identification of any given landmark and the effect of its identification on the vehicle's position uncertainty. Here, we consider only the latter; see [1] for a discussion of the former.

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 1a. 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 1b. 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 1c.

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. 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.

To express in one parameter the uncertainty represented by an arbitrary 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.

3. QUADTREE PATH PLANNING

We are developing a system of algorithms for mobile robot path planning based on a multiresolution representation of the robot's immediate environment. The multiresolution representation used is the quadtree (Samet[7]). Figure 2 illustrates the quadtree representation for a simple binary array where black points represent obstacle points and white points represent free space. The quadtree is a recursive decomposition of that array into uniformly colored (i.e., either black or white) $2^i \times 2^i$ blocks. Thus, if there are large areas of free space (or of obstacles) then those areas can be represented by a few large blocks in the quadtree and can be dealt with as units by the planning algorithms.

The quadtree representation thus offers a compromise between a simple homogeneous array representation (which is straightforward to construct but then computationally costly to analyze) and a free space region representation (e.g., Brooks[8]) which is more costly to construct, but on the other hand more efficient to analyze.

Before discussing the planning algorithm, we should point out that there are several important differences between path planning requirements for a mobile robot and for the more familiar manipulators (see also the discussion in Thorpe[9]). For example:

1) A mobile robot may have only an incomplete model of its environment, perhaps because it constructs this model using vision and thus cannot determine what lies in the shadow of an object.

2) A mobile robot will ordinarily only negotiate any given path once (as opposed to a manipulator which might perform the same specific task thousands of times). Therefore, it is more important to develop a negotiable path quickly than it is to develop an "optimal" path, which is usually a costly operation.

3) A mobile robot will be moving according to a previously computed path while it is computing an extension or modification to that path - i.e., path planning for a mobile robot is a continuous, online process rather than a single, offline process.

The algorithm that follows only addresses the second of the above three points. A path generated from a quadtree is a sequence of blocks through which it is possible for the robot to move. The detailed motion within any single block is not determined at this level; a default assumption of straight line motion through the block is assumed. Although this will not ordinarily be an optimal path, it will be a negotiable path.

Given the quadtree representation in which blocks of 0's represent free space and blocks of 1's represent obstacles, we first compute the *distance transform* of the set of 0's. This determines, for each block of free space, the minimal distance between the center of that block and the boundary of a block of obstacles. Samet[7] describes an algorithm for computing the distance transform for a quadtree.

The path planning algorithm itself is a simple A^* search algorithm with the evaluation function, f, defined as follows:

$$f = g + h$$

where g is the distance of the current node in the search from the start node, and h is the heuristic estimate of the goodness of the remainder of the path passing through that node. The heuristic h is the difference of two components, h_G and h_d, where h_d is the distance of the nearest obstacle from the current node (determined by the distance transform algorithm) and h_G is the straight line distance between the current node and the goal.

In developing the search, only the horizontal and vertical neighbors of any block are considered for building extensions of paths. Diagonal neighbors, which share only single points with the current node, would result in inflexible paths which clip corners.

The result of applying the A^* algorithm to the quadtree is a list of nodes from the quadtree (ordinarily of varying sizes) which define a set of paths between the start node and the goal. We can, if desired, determine the optimal path through these blocks, or we can simply connect the center points of consecutive blocks on the list to compute a path.

Figure 3 contains a simple example. Figure 3a is a binary array with start and goal points marked, along with an indication of the path determined by the algorithm. Figure 3b contains the tree data structure that represents the quadtree, with the blocks on the computed path marked with P's.

An important extension of this simple path planning algorithm involves the ability to deal with grey nodes in the quadtree (nonterminal nodes which always have both black and white descendants). Dealing with grey nodes can greatly reduce the number of blocks that the planning algorithm needs to consider in building an initial estimate of a path. Such an algorithm is presented in [4].

4. CONCLUSIONS

The framework for visual navigation that we have presented in this paper must itself be incorporated into an even more comprehensive navigation framework that considers analyses of many other sources of information - e.g., other sensors, maps, reconnaissance data, etc. Planning and executing navigation tasks at this level will require very complex models for data fusion, resource allocation and problem representation and solving. An important goal of the autonomous navigation project is to structure this framework in such a way that it is possible for a broad spectrum of research groups to contribute to, and experiment with, the vehicle (or an appropriate simulation of the vehicle). Achieving this goal would, hopefully, lead to a better understanding of not just vision as an isolated activity, but of vision as part of a more comprehensive intelligent activity.

ACKNOWLEDGEMENTS: The framework for visual navigation described in Section 1 of this paper evolved during many months of discussion in the Computer Vision Laboratory. Allen Waxman and Jacqueline LeMoigne made significant contributions to that work.

REFERENCES

1. Andresen, F. and Davis, L., Vehicle positioning using landmark identification, University of Maryland Center for Automation Research Technical Report, in preparation.

2. Waxman, A., LeMoigne, J., and Srinivasan, B., Visual navigation of roadways, this proceedings.

3. Zuk, D. and Dell'eva, M., Three-dimensional vision system for the adaptive suspension vehicle, Environmental Research Institute of Michigan Technical Report, January, 1983.

4. Kambhampati, S. and Davis, L., Path planning in quadtrees, University of Maryland Center for Automation Research Technical Report, in preparation.

5. Ballard, D., Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition, 13,* 111-122, 1981.

6. Haralick, R. and Shapiro, L., The consistent labeling problem, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1, 173-183, 1979.

7. Samet, H., The quadtree and related hierarchical data structures, University of Maryland Center for Automation Technical Report 23, November 1983.

8. Brooks, R., Solving the find-path problem by representing free space as generalized cones, M.I.T. Artificial Intelligence Memo 674, May 1982.

9. Thorpe, C., Path relaxation: Path planning for a mobile robot, Proceedings of the National Conference on Artificial Intelligence, Austin, Texas, August 1984.

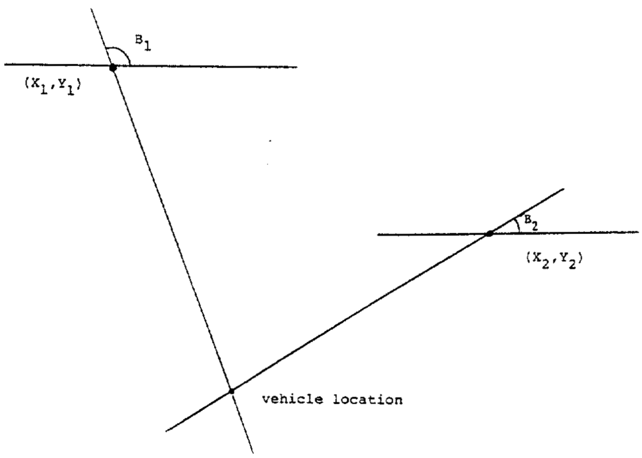


Figure 1a. Triangulation

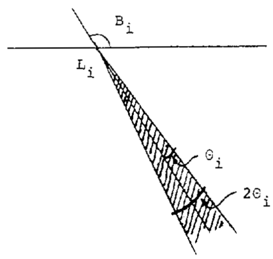


Figure 1b. Possible positions given one landmark's bearing

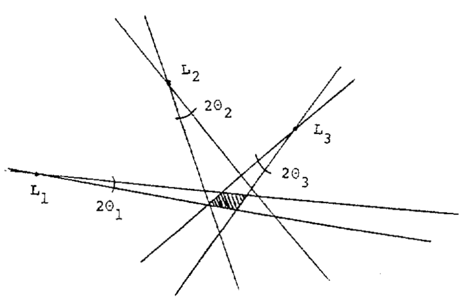
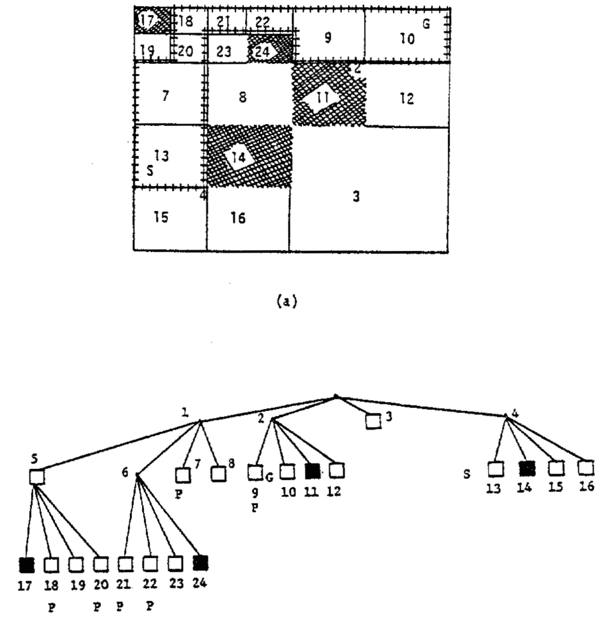


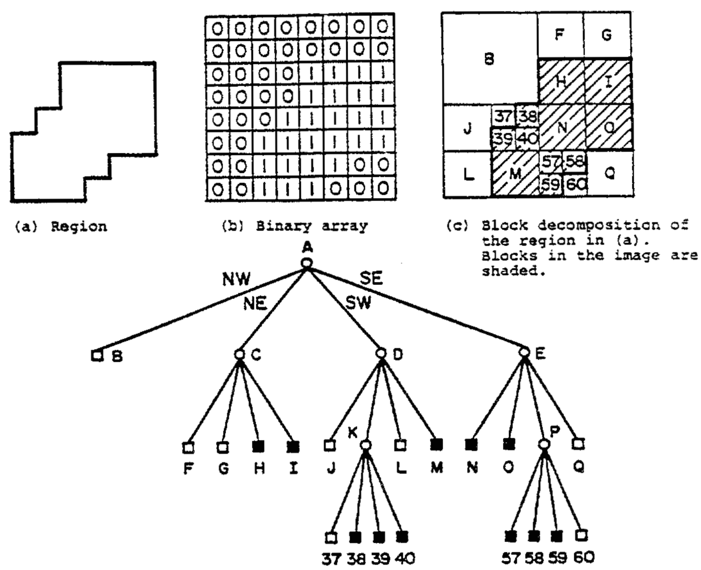
Figure 1c. Possible positions with three landmarks' bearings



(a)

(b)

Figure 3.



(a) Region

(b) Binary array

(c) Block decomposition of the region in (a). Blocks in the image are shaded.

(d) Quadtree representation of the blocks in (c).

Figure 2. A region, its binary array, its maximal blocks, and the corresponding quadtree (from [7]).