Conf-9410186--1

Unknown-Terrains Navigation of a Mobile Robot Using an Array of Sonars†

Nageswara S. V. Rao
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364
nrao@plato.epm.ornl.gov

Submitted to *International Dedicated Conference on Robotics, Motion and Machine Vision*, October 31 - November 4, 1994, Aachen, Germany.

†Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc.

Unknown-Terrains Navigation of a Mobile Robot Using an Array of Sonars

Nageswara S. V. Rao
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364, USA
nrao@dizzy.epm.ornl.gov

Abstract

A mobile robot equipped with an array of sonars is required to navigate to a destination through a planar terrain populated by polygonal obstacles whose locations and shapes are unknown. A navigation method is proposed based on a trapezoidal decomposition of the terrain for an abstract formulation, where elementary navigational steps consist of following the obstacle edges and turning around the corners. The convergence of an abstract version of the algorithm is first analytically established. Then experimental results on implementing the algorithm on an experimental mobile robot are reported.

1 Introduction

Navigation and path planning constitute critical tasks in the operation of autonomous mobile robots. In the past decade, various abstract formulations of this problem have been solved by several researchers (Latombe [4], and Hwang and Ahuja [2]). The *navigation problem* deals with computing a collision-free path for a robot from a source position to a destination position in a terrain populated with obstacles. In a *known terrain*, a complete model of the terrain is available, and the path planning can be performed using several techniques such as retraction, decomposition, etc. (Sharir [10]). In an *unknown terrain*, the model of the terrain is not known but a sensor system is employed for navigational purposes. For the first part of this paper, we consider an abstract formulation consisting of point-sized mobile robot equipped with a *discrete vision* sensor, which detects all visible (from the present location) parts of the obstacle boundary in a single *scan* operation. We then briefly consider a *continuous vision* sensor that detects all visible parts of the obstacle boundary as the robot moves along a path. The abstract formulation allows us to theoretically validate the algorithm. Then we describe an implementation of a navigation algorithm for a mobile robot equipped with an array of sonars.

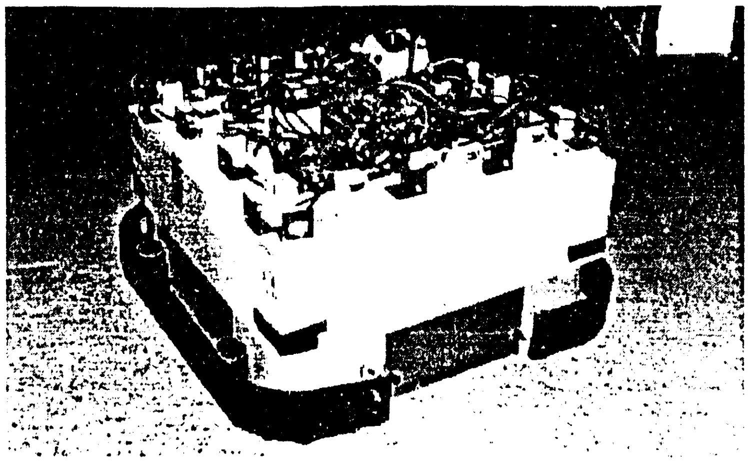

Figure 1: TRC Labmate with sonar arrays.

A number of sonar-based navigation algorithms have been developed based on different methods. For example, Elfes [1] describes a method based on occupancy cells, Watanabe and Pin [11] describe a method based on fuzzy logic, and Mataric [5] describes a reactive method. These methods are designed for a circular arrangement of sonars and are not specifically designed to take into account the polygonal nature of the obstacles. Main attractive features of sonars are their low cost and simplicity of the information returned. On the negative side, the information returned by them is not very accurate and is prone to errors. A single sonar returns an estimate of a distance to an obstacle contained within a spatial cone in a direction. Thus even if the distance estimate is accurate, there are information-based limitations in "realizing" a scan operation (described above) using sonar arrays. Sonar readings, however, are fairly reliable and accurate in measuring distances to walls located within certain operative range. Our method employs an array of sonars to follow walls and turn around the corners so that the obstacles are kept (to the most extent) in a suitable range of the sonars.

Abstract (non-heuristic) algorithms for the robot navigation in unknown terrains have been studied by a number of researchers in the last decade (Rao et al. [9]). A navigational course is a 1-skeleton embedded in the set of all free-positions of the robot. In the case of a discrete vision sensor, Rao [7] showed that if the navigation course satisfies the four properties of finiteness, connectivity, terrain-visibility and local-constructibility, then navigation problem can be solved by employing a graph search algorithm. For polygonal terrains, one can employ navigational structures based on the visibility graphs and the Voronoi diagrams [7]. Generally, visibility graph methods require that the robot navigate along the obstacle walls; thus this method is not practical due to inaccuracies in robot motions. On the other hand, the retraction based methods keep the robot far way from the obstacles so that accurate obstacle information is not easy to obtain (for terrain mapping purposes): this is particularly a problem with a circular sonar arrangements because the sonar resolution decreases with the increase of distance from the robot. Furthermore for sonars, the accuracy of the readings is good when the obstacles are within some proximity of the sensors. In this paper, we present a method based on a trapezoidal decomposition of free-space: the idea of this method was first proposed by Kim [3] and analyzed by Rao [8]. In this method, the robot uses the obstacle edges as guidelines without getting too close to or too far away from them.

The organization of this paper is as follows. We discuss preliminaries and an abstract algorithmic framework for the navigation in unknown terrains in Section 2. In Sections 3 and 4, we discuss dual graphs based on trapezoidal decomposition that are suitable for

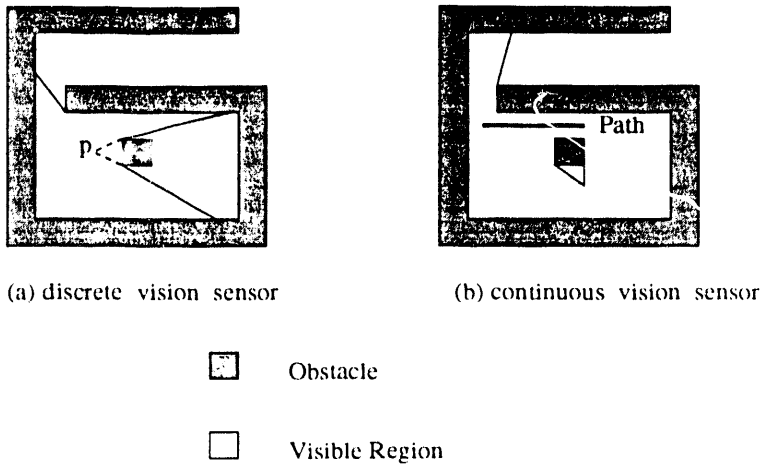

Figure 2: Discrete and continuous vision sensors.

navigation using discrete and continuous vision sensors respectively. An implementation on TRC Labmate mobile robot, shown in Fig. 1, is described in Section 5.

2 Preliminaries

We consider a finite two-dimensional terrain populated by a finite and non-intersecting set O of polygons. Two points p and q are *visible* to each other if the line segment joining p and q is not intersected by any obstacle. For the abstract formulation of the problem, the robot R is point-sized and equipped with a vision sensor. A *discrete vision sensor* is characterized by a *scan* operation: a scan operation performed from a position (point) p returns the *visibility polygon* of p, which is the polygonal region consisting of all points in the terrain visible to p (Fig. 2(a)). A *continuous vision sensor* when invoked as the robot moves along a path P returns the generalized polygonal region such that every point of this polygon is visible from some point on P (Fig. 2(b)), i.e. the polygon returned in this case is the union of the visibility polygons of all points on P. Thus, in general, the operation of a continuous vision sensor cannot be simulated by a discrete vision sensor if only a finite number of scan operations by the latter are allowed.

We now describe dual graphs based on a trapezoidal decomposition. First, we decompose the free-space into trapezoids by sweeping a line (for example, moving a horizontal line from top to bottom) such that whenever the line passes through a vertex, we extend a *sweep-line segment* from this vertex into free-space until it touches an obstacle boundary or extends to infinity as shown in Fig. 3. Now free-space is partitioned into trapezoids. For each sweep-line segment we have one of the two following cases: (a) if the segment is finite, the dual graph node corresponds to the mid point of the segment, or (b) if the segment is not finite, the dual graph node corresponds to a point on the segment at a distance δ from the vertex. Two nodes belonging to the boundary of the same trapezoid are connected by an edge of the dual graph (see Fig. 3).

An algorithmic framework for solving the navigation and terrain model acquisition problems using discrete vision sensors has been proposed by Rao [7]. Here the robot R uses a one-skeleton (a collection of one-dimensional curves) $\xi(O)$, called the *navigational course*: $\xi(O)$ can be simply viewed as a combinatorial graph such that each ξ-vertex specifies a position for R and each ξ-edge (u, v) specifies a path from u to v. To navigate from s to d, R performs a "graph search" on $\xi(O)$. Initially $\xi(O)$ is not known to R, but it is incrementally constructed from the sensor operations. From s, R initially checks to see if d is reachable, and moves to it if yes. If not, R computes a start ξ-vertex v_0 and moves to it, and from v_0, R

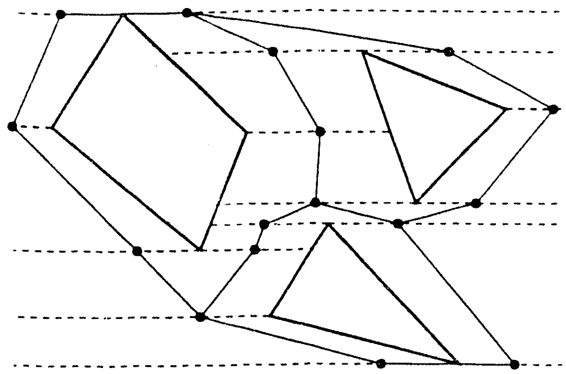

Figure 3: Dual graph based on trapezoidal decomposition.

keeps visiting new ξ-vertices until it reaches a ξ-vertex v_f from which d is found reachable. In the case d is not reachable from s, R visits all ξ-vertices and concludes that the destination is not reachable. The requirement on the graph search algorithm is that it must be capable of visiting all vertices of a connected component of the $\xi(O)$ in which it is initiated, e.g. depth-first search and A^* algorithms [6].

We now consider four properties for $\xi(O)$: (i) *finiteness* property requires that the number of ξ-vertices and edges be finite, (ii) *terrain-visibility* requires that every point outside the obstacles is visible from some ξ-vertex, (iii) *connectivity* property requires that every pair of ξ-vertices be connected by a graph path on $\xi(O)$, and (iv) *local-constructibility* requires that the adjacency list of a ξ-vertex v be computable from the information obtained by performing a finite number of sensor operations. Then the following result can be easily shown [7].

Result 2.1 *Given a navigational course $\xi(O)$, that satisfies the properties of finiteness, connectivity, terrain-visibility and local-constructibility, a graph search algorithm can be employed to solve the navigation problem using a discrete vision sensor.*

3 Navigational Structure and Algorithm

In the trapezoidal decomposition, each trapezoid is bounded by at most two (continuous segments of) obstacle edges and at most two (but at least one) sweep-line segments; each such sweep-line segment contains at least one obstacle vertex. Note that each trapezoid is convex, and the line segment joining two vertices of a trapezoid does not intersect the interior of any obstacle.

Theorem 3.1 *Consider a terrain such that no two obstacles are co-linear with respect to the sweep-line. The number of vertices in the dual graph $D_T(O)$, based on the trapezoidal decomposition, is upperbounded by $N+2n$ for a terrain of n polygonal obstacles consisting of a total of N vertices. Also, $D_T(O)$ satisfies the properties of connectivity, terrain-visibility and local-constructibility.*

Proof: First consider the bound on the vertices of D_T. The obstacle vertices can be classified into two main categories, those which form an inflection point (a point that supports a local tangent) in the chosen sweep-line direction and those that do not; vertices of first kind are called *inflection vertices* and those of second kind are called *non-inflection vertices* (see Fig. 4). Without loss of generality we assume that the sweep-line is horizontal. An inflection vertex v is *pointing up* (*pointing down*) depending on if the obstacle in the vicinity of v is below (above) the sweep-line through v. From each non-inflection vertex there will be a segment of the sweep-line (through it) that contains precisely one node of D_T. In general

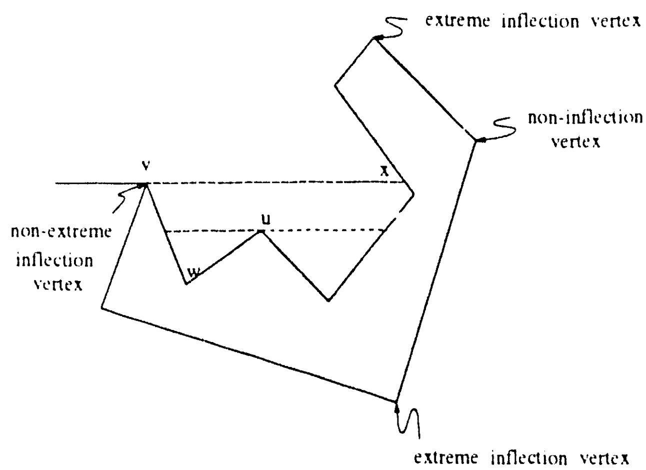

Figure 4: Inflection and non-inflection vertices.

each inflection vertex generates two vertices of D_T, and thus one can obtain a crude upper bound of $2N$ on the number of nodes of D_T. We now tighten this bound to $N + 2n$ nodes.

Consider the terrain with convex polygonal obstacles such that no two vertices have the same X-coordinate. Here each obstacle contains precisely two inflection vertices, each of which generates two nodes of D_T, and every non-inflection vertex generates a single D_T node. Thus the total number of nodes of D_T is exactly $N + 2n$ for this case. We now show that $N + 2n$ is an upper bound for the case containing possibly non-convex polygonal obstacles. As each obstacle is swept, we have two extreme inflection points corresponding to the first and the last times the obstacle is encountered: each of these inflection points account for two nodes of D_T.

We show that for each non-extreme inflection vertex v there corresponds a concave vertex w that does not generate a node of D_T; w accounts for one of the nodes on the segment through v. We show this result for pointing up non-extreme inflection points and that for pointing down non-extreme inflection points is similar. To make the discussion concrete, we assume that of the two graph nodes generated by v, the one to the left of v is accounted for by v and the one to the right is accounted by the yet to be described w. Visualize that we move the sweep-line from top to bottom. As we sweep, extend the sweep from non-extreme inflection point v to the right until it meets the obstacle that v belongs to at point x (pretend that the rest of the obstacles are transparent for this construction). Then consider the polygonal region P_1 enclosed by the line segment and the boundary of the obstacle from v to x. As the sweep-line moves down from v, we follow the intersection point of the boundary of P_1 with the sweep-line until we encounter a concave vertex w of the original obstacle. Note that no nodes of D_T are generated when the sweep-line meets w, since the sweep-line lies inside the obstacle around w. It is possible that before w is found another non-extreme inflection point u on the boundary of P_1 is met by the sweep-line: if that happens the graph node to the left of u is accounted by u. The proof is complete by noting that w always exists since P_1 is a polygonal region.

We now consider the connectivity property. Consider two points x and y in free-space and a shortest path P between them. Then P can be decomposed into line segments such that each segment is entirely contained in a trapezoid; let the resultant segments be denoted by $\overline{xp_1}, \overline{p_1p_2}, \ldots, \overline{p_{m-1}p_m}, \overline{p_my}$ listed in the sequence as we navigate from x to y. We now navigate along this sequence and generate a path on the dual graph as follows. In the trapezoid that contains x, note that p_1 lies on a sweep-line segment; now we slide the p_1-end of the segment $\overline{xp_1}$ along the sweep-line until we reach a graph node at point p_1^1. Then in the next trapezoid, we consider the segment $\overline{p_1^1p_2}$, and we slide the p_2-end of this segment along the sweep-line until we meet the node p_2^1. This process is continued in the sequence until the last trapezoid is reached. Note that each slide operation is possible since each trapezoid in which the sliding operation takes place is convex. The resultant path $x, p_1^1, p_2^1, \ldots, p_m^1, y$ is a path on the graph D_T. By restricting x and y to the nodes of D_T, the connectivity property follows.

In the decomposition, the free-space is decomposed into trapezoidal regions and there is at least one node of V_T associated with each of the trapezoids. Since each trapezoid is convex every point in a trapezoid is visible from the corresponding node of D_T, and thus every point in the free-space is visible from some node of D_T.

When a scan operation is performed from a node $v \in D_T$, the trapezoidal region that contains v will be contained in the visibility polygon returned by the scan operation. Given the sweep-line direction, the required trapezoid region can be obtained by computing the closest vertex to the sweep-line containing v in the required part of the visibility polygon. Thus local-constructibility property is satisfied. □

In the navigation problem, the task is to reach a destination position d, while avoiding obstacles on the way, or to conclude that d is not reachable. R keeps visiting newly computed vertices until d is reachable. For example, R can employ the depth-first search to visit the vertices.

Finally, some other decompositions such as triangulation, convex polygonal decomposition, etc., can be used to generate suitable navigational courses. Also, even in the case of trapezoidal decomposition, there could be other ways of defining a dual graph. For example, each dual node could correspond to the centroid of a trapezoid, and a dual edge joins two nodes whose trapezoids share a sweep-line segment.

4 Continuous Vision Sensors

In terms of the general paradigm, we only require that the navigation course used by a continuous vision sensor is a 1-skeleton that satisfies the two following properties: (a) *connectedness*, which requires that there is a path on the navigation course between any two points of the navigation course, and (b) *terrain-visibility*, which requires that every point in the free-space is visible from some point on the navigation course. We also require that the navigation course to be a 1-skeleton in plane such that the total length of the 1-skeleton is finite. Given such a navigation course, if the robot has navigated along the entire navigation course, then entire free-space will be visible. Thus a robot can navigate along the navigation course until it the destination is reachable, or entire navigation course has been traversed. Any such algorithm can be seen to solve the navigation problem. By Theorem 3.1, the dual graph $D_T(O)$ based on the trapezoidal decomposition satisfies the required properties.

5 Implementation

The implementation has been carried out on TRC Labmate mobile platform which has been augmented with a sensory system of 16 ultrasonic sensors (Fig. 1) built by us. A schematic of the coverage of the side sensors is shown in Fig. 5. Both the mobile platform and the sensor system are controlled from SPARC workstation though a RS232 radio link, through an interface developed by us.

An approach for using sonar arrays is to emulate the basic navigational strategies of last section. However, there are two critical aspects to this approach. First, the distances in discrete directions provided by sonars can only approximate the scan information required by the vision-based algorithms. Second, the algorithm must be robust enough to handle the errors in sensor data and in the robot motions. The navigation algorithm based on

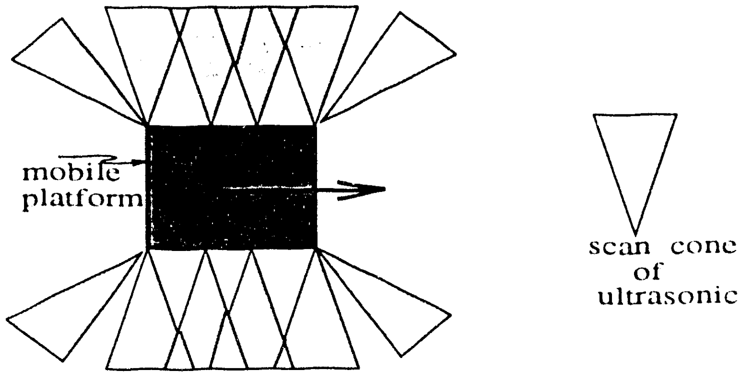

Figure 5: Top-view of the robot and sonar cones.

trapezoidal decomposition can be conceptualized in terms of moving from one trapezoid to the other as per the search strategy. Except for the trapezoids that extend from infinity to infinity, each trapezoid consists of an obstacle edge, which can be followed by a mobile robot using a sonar array. The robot follows an obstacle edge until it moves into new trapezoid, and can switch to a new edge (if required) as exemplified in Fig. 6. At present, the application is for navigating in corridors with sufficient clearances. The navigation algorithm consists of navigation along the edges and turning around the corners if necessary. The edge-following algorithm is implemented as a simple control algorithm, where the robot attempts to maintain within interval of a prescribed distance from the wall. The distance from all the four side sensors facing the wall are averaged, and the robot is given a correction in its heading based on the difference of the reading and a specified value. This task is achieved by a simple discrete-time control algorithm. The corners are detected by an abrupt change in the sensor distances. Notice that (i) we obtain reasonable distance estimates by keeping the robot within a suitable proximity of the obstacles, and (ii) the task of following the straight-line edges can be achieved reasonably well in the presence of sonar and motion errors.

At present, we tested the navigation algorithm around thin walls and convex obstacles. The corners tested are all convex with included angle of at least 90 degrees. The implementation is presently ongoing to extend present version to handle concave obstacles and cluttered environments.

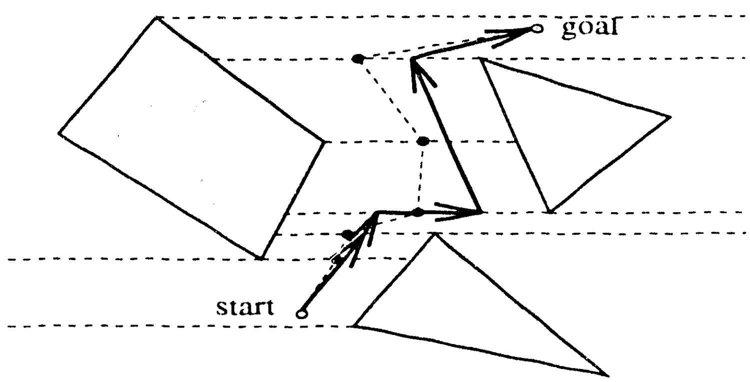

Figure 6: Navigation using trapezoidal decomposition.

6 Conclusions

The navigation problem deals with moving a point robot R through an unknown terrain from a source position s to a destination position d. A navigation algorithm that uses abstract vision sensors is presented based on a dual graph defined on the trapezoidal decomposition of the free-space. This algorithm is well suited for a mobile robot equipped with an array of sonars since the elementary navigation steps involve following walls and turning around the corners. Some experimental results are presented based on implementation on TRC Labmate mobile platform equipped with an array of sonars. The proposed abstract algorithm can be extended to the case where some of the obstacle edges are circular arcs [8]. In the implementation using sonar arrays, however, the present wall-following algorithm has to be enhanced since the present version can accurately follow only straight line walls; this is a topic of our ongoing research.

Acknowledgements

This research is sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc.

References

[1] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3:249–265, 1987.

[2] Y. K. Hwang and N. Ahuja. Gross motion planning - a survey. *ACM Computing Surveys*, 24(3):219–291, 1992.

[3] M.-S. Kim. private communication.

[4] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Pub., Boston, 1991.

[5] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, 1992.

[6] J. Pearl. *Heuristics: Intelligent Search Strategies For Computer Problem Solving*. Addison Wesley Pub., Reading, Mass., 1984.

[7] N. S. V. Rao. Algorithmic framework for learned robot navigation in unknown terrains. *IEEE Computer*, 22:37–43, 1989.

[8] N. S. V. Rao. Robot navigation in unknown generalized polygonal terrains using a vision sensors. *IEEE Transactions on Systems, Man and Cybernetics*, 1994. to appear.

[9] N. S. V. Rao, S. Kareti, W. Shi, and S. S. Iyengar. Robot navigation in unknown terrains: An introductory survey of non-heuristic algorithms. Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, Oak Ridge, TN, 1993.

[10] M. Sharir. Algorithmic motion planning in robotics. *IEEE Computer*, pages 9–20, March 1989.

[11] Y. Watanabe and F. G. Pin. Sensor-base navigation of a mobile robot using automatically constructed fuzzy rules. In *Proceedings of 6th International Conference on Advanced Robotics*, pages 81–87, 1993.

AIIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

301/587-8202

Centimeter

Inches

MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.