An Efficient Strategy for Fast Object Search Considering the Robot's Perceptual Limitations

Javier Cabanillas, Eduardo F. Morales, and Luis Enrique Sucar

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro 1, Tonantzintla, Puebla, Mexico
{javier.cabanillas}@ccc.inaoep.mx,
{esucar,emorales}@inaoep.mx
http://ccc.inaoep.mx

Abstract. Searching for an object in an environment using a mobile robot is a challenging task that requires an algorithm to define a set of points in which to sense the environment and an effective traversing strategy, to decide the order in which to visit such points. Previous work on sensing strategies normally assume unrealistic conditions like infinite visibility of the sensors. This paper introduces the concept of *recognition area* that considers the robot's perceptual limitations. Three new sensing algorithms using the recognition area are proposed and tested over 20 different maps of increasing difficulty and their advantages over traditional algorithms are demonstrated. For the traversing strategy, a new heuristic is defined that significantly reduces the branching factor of a modified *Branch & Bound* algorithm, producing paths which are not too far away from the optimal paths but with several orders of magnitude faster that a traditional *Branch & Bound* algorithm.

1 Introduction

One of the main goals in robotics is to design an autonomous robot able not only to move around an environment and to avoid objects, but also to interact with objects accomplishing a visibility-based task. In particular, this work is focused in the problem of searching for an static object in an known environment. Our goal is to generate a set of sensing locations and develop a motion strategy to visit this set to find the object in a reasonable time.

In any environment of n vertexes and h holes, $\lfloor (n+h)/3 \rfloor$ sensing locations are sufficient, and sometimes necessary, to see every interior point of the environment [4]. This set of sensing locations, however, is usually far from optimal in terms of cardinality. Finding the minimal number of sensing locations is an open problem called the Art Gallery Problem (AGP) [8]. This problem consists of minimizing the cardinality of the set of guards or sensing locations required to cover a polygon. There are different schemes to generate and to combine this set of sensing locations and there is a large number of algorithms that have been proposed related to the AGP and its variants to generate a set of sensing locations (e.g., [7,4]).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 552–561, 2010.

There is also ample work on exploration and mapping within the area of rescue, search and surveillance robots, but generally it is focused on exploring an unknown environment, which is a different task (e.g., [6,2,1]).

In known environments, one approach to search for objects considers a robot continuously sensing the environment. In this approach, a simple Levy's flight or a greedy strategy using the gradient of the new visibility area can be used to guide the search [12]. An alternative approach considers only a set of specific locations where the robot senses the environment, which simplifies the task. In [9,10] the authors assume that a set of sensing locations is given as input. In order to solve the search problem, they used a heuristic based on an utility function, defined by the probability of see the object in the location over the distance traveled. A methodology that takes into account the problem of recognition is proposed in [11]. In order to achieve a reliable recognition, an attention mechanism based on receptive field co-occurrence histograms and SIFT feature matching is proposed. The authors also present a method for estimating the distance to objects. In this work, however, the set of sensing locations is also given as input.

A common weakness of current approaches that work with sensing locations is that they assume infinite visibility for the sensors, as define in the classic AGP [7]. This is clearly an unrealistic assumption for the sensors used in real robots. In this paper, we introduce the concept of *recognition area* that depends on the sensing capabilities of the robot and the characteristics of the object.

Different sensing locations strategies have been proposed in the past. In this paper, we incorporate the recognition area concept into three new sensing location algorithms. The first algorithm improves over a triangulation-based algorithm that generates a smaller number of points and also moves further away from obstacles. The second algorithm incorporates a pre-processing method to a probabilistic road map (PRM) algorithm to significantly improve its performance. The third algorithm is based on a random generation process followed by a greedy search strategy to very quickly generate sensing paths.

Once the set of sensing locations has been defined, they have to be visited in a specific sequence in order to minimize the total traveled distance to find the object. This combinatorial problem is solved using a new heuristic that reduces the branching factor of a *Branch & Bound* algorithm. The results show that the distance traveled is not far from the optimal and the run time is up to 1000 times faster in the tested maps than a standard *Branch & Bound* algorithm.

The rest of this paper is organized as follows: Section 2 presents the terminology and main concepts used in this paper, Section 3 introduces the proposed solutions both for generating sensing points and for creating a traversing strategy. The experiments and results are described in Section 4. Finally, conclusions and future research directions are given in Section 5.

2 Problem Definition

Polygons are a convenient representation for many real-world environments because a polygon is often an accurate-enough model for many real objects and polygons are easily manipulated in computational terms [8]. Therefore, it is assumed that the environment A is known and modeled as $P-H$ (see Figure 1 (b)), where P is an external polygon of the environment and $H = h_1, h_2, .., h_i, .., h_n$ is the set of internal polygons or holes (see Figure 1 (a)). If $|H| = 0$, P is considered as *simple*. If P has one or more holes, any hole in P is *simple*, which means that no hole can have holes within itself. The robot is considered as a simple polygon R and Minkowsky sums of (R, h_i), $\forall h_i \in H$ are used to get a workspace W that takes into account the geometric restrictions of R and the shape of A [5] (See Figure 1 (c)).

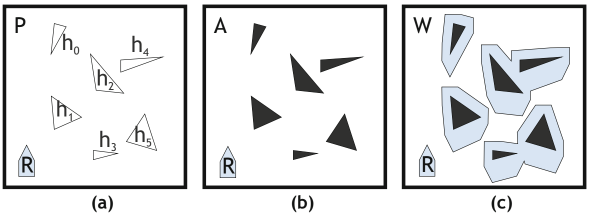


Fig. 1. (a) The exterior polygon P and the set of interior polygons $H = h_0, h_1, h_2, h_3, h_4, h_5$. (b) The environment $A = P - H$. (c) The calculated workspace $W = MinkowskySum(R, h_i) \forall h_i \in H$.

Since $Area(W) \subset Area(A)$, in the infinite visibility model, the largest line segment inside the visibility polygon of any point $p \in W$ is not larger than the largest axis of the bounding box of A. This allows to define a limited radius for an infinite range camera sensor. However, this is not enough to overcome the limitations of the model as shown in Figure 2.

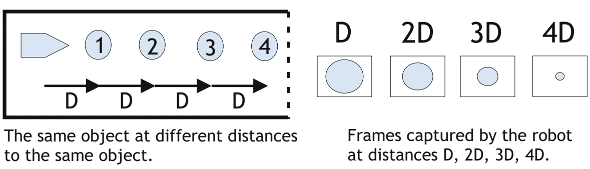


Fig. 2. If the distance to the object continues to increase, the object can be impossible to identify by the robot

Considering infinite visibility, the object finding problem ends when the object is within the visibility polygon of the observation point p. However, in practice, even if the object is inside the visibility radius from p, the recognition algorithm can fail when the distance to the object is too large or too small. This is like having at the same time a myopic robot, that cannot recognize objects that are too far away, and a robot with astigmatism that cannot recognize objects that are at very short distances.

Definition 1: The *recognition area* of a sensor s is defined by the minimum (d_{min}) and maximum (d_{max}) distances at which s is able to reliably recognize an object.

The size of the recognition area depends on the quality of the sensors and is an adjustable parameter of the proposed algorithms. We simulated different sizes of recognition areas in the reported experiments, however, it can be estimated by moving a robot from a long distance towards an object. Although the recognition algorithm can fail even at the recognition area, in this paper it is assumed that the sensor is "perfect" within this range.

After the generation of the set of sensing locations, G, it is still necessary to implement a strategy to visit all of them. The robot starts at a particular location, $g_0 \in G$, and visits the other locations as time progresses. The objective in this step is to find the route that minimizes the traveled distance to find the object. The searching task finishes when the object is inside the recognition area of the actual robot location.

3 Proposed Solution

The proposed solution has three steps: (i) compute the workspace using Minkowsky Sums, (ii) find a set of sensing locations that covers as much as possible the environment in a reasonable time considering the recognition area of the sensors, and (iii) find a traversing strategy in a reasonable time that minimizes the total travelled distance. The complete strategy is shown in Figure 3.

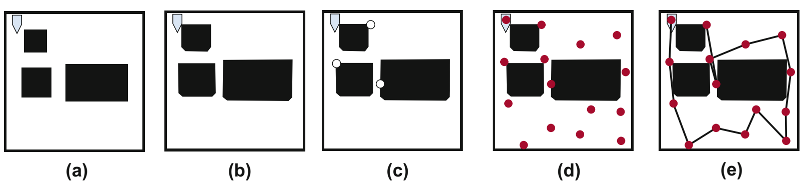


Fig. 3. The figure shows the complete strategy and its application in a simulated environment. The path is computed in 0.14 seconds for the square environment of 512 units of side and 100 units of visibility radius. (a) Robot and environment. (b) Minkowsky Sums. (c) Generation of a candidate set. (d) Generation of the final set. (e) Computation of the path.

3.1 Sensing Locations

In this paper we introduce three algorithms for sensing location based on: a) polygon partition, b) Probabilistic RoadMaps and c) a completely random process and a greedy strategy. Their advantages and disadvantages of each one are analyzed with 20 different maps.

Algorithm based on Triangulation. The decomposition of a simple polygon into triangles is called a triangulation. Various algorithms have been developed for triangulation, each characterized by its asymptotic order as n grows without bound. The simplest algorithm, called Ear Clipping, is the algorithm used in this paper. Algorithms with better asymptotic order exist, but are more difficult to implement [3]. Ear Clipping was designed for simple polygons but can also be applied to polygons with holes by following the process described in [3]. To improve its performance, a candidate set C is built as the union of the set of vertexes (external and internal) and the set of circumcenters (see Figure 4).

The goal is to generate a visibility or guarding set $G \subset C$ that is as small as possible; it is expected for $|G|$ to be much smaller than $|C|$. The algorithm greedily adds candidates using a scoring function to the guarding set G, until the entire polygon is covered. The score, $\sigma(c)$, for a candidate $c \in C$ is the number of points in C that are seen from c and that are not already seen by a point in G. At each iteration of the algorithm, the candidate c with the highest score $\sigma(c)$ is added to G, and the scores $\sigma(*)$ are updated accordingly. The algorithm stops when all the $c \in C$ are considered or when A is totally covered. The proposed algorithm has two main advantages over the basic triangulation algorithm: (i) It generates a smaller number of sensing points and (ii) the generated points tend to be farther away from obstacles. However it needs more computational time than the classic triangulation algorithm.

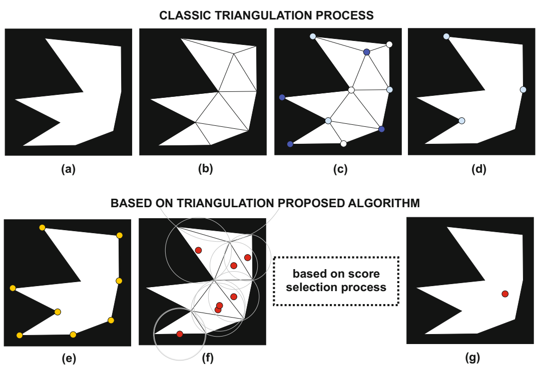


Fig. 4. The figure shows the proposed triangulation process. (a) The polygon. (b) Triangulate the polygon. (c) 3-Colored Algorithm. (d) Selected sensing locations of the basic algorithm. (e) The candidate set 1. (d) The candidate set 2. (f) The selected sensing location after the greedy algorithm process.

Improved Algorithm Visib-PRM. Roadmaps are usually used to solve motion planning problems, but in this case, it is used to obtain a set of sensing locations in the collision-free space (i.e., $P - H$). Due to the probabilistic completeness of the PRM method, we are sure that eventually G will complete the connectivity of the free space, but to prevent exorbitant running times, normally the node generation process is guided by heuristics that create higher node densities in difficult areas. In this paper a list of points G_i of least visibility that are closer to obstacle edges and around the convex obstacle corners is obtained and one point by each obstacle is selected by its respective area. This preprocessing step incorporated into the original algorithm significantly boosts the performance of the generation set algorithm. Figure 5 show the proposed Visib-PRM algorithm.

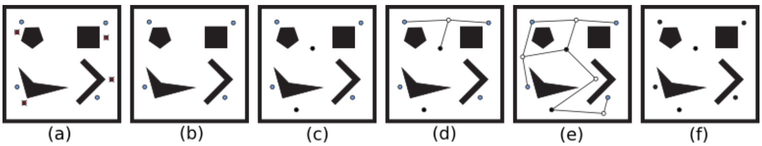


Fig. 5. (a) The selection process of the preprocessing step. (b) G_i (c,d,e) Application of Visib-PRM algorithm. (f) The final set of guards.

It is easy to adapt the original algorithm to work with a recognition area because although the basic algorithm assumes infinite visibility conditions, when it is applied in a domestic environment, this area of visibility is limited by the walls. Therefore, we can easily simulate restricted visibility conditions using a tunable parameter that simulate walls and that depends on the quality of the sensors.

Roma Algorithm. The proposed version of Visib-PRM is a partially random algorithm. Its principal weaknesses is that it depends on the environment and the selection process can need significant computational resources. The Roma algorithm is a very fast random-based algorithm combined with a greedy strategy that works as follows:

1. Generate a random point on the map.
2. Compute the covered and uncovered regions. This is shown in Figure 6 and can work both with an infinite or limited recognition areas.
3. Select the region of largest area and obtain its bounding box.
4. Generate a new random point until it is inside the previously computed bounding box.
5. Repeat the process until the number of generation attempts is equal to a threshold value M or the map is completely covered.

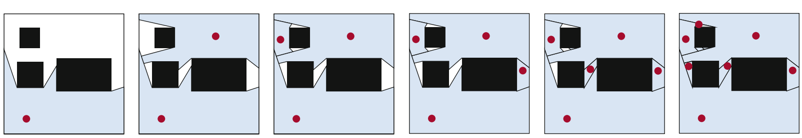


Fig. 6. The figure shows the generation process of ROMA algorithm using infinite visibility

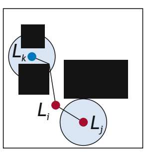
Randomized points On the MAp (ROMA): (i) can work an order of magnitude faster than the previous algorithms, (ii) does not use a candidate set, therefore it avoids the selection process of the previous algorithms, (iii) it generates points only in the configuration free-space, and (iv) its dependency on the map complexity is lower than the partition based algorithms.

3.2 Motion Planning

Given a set of sensing locations we need a strategy to visit them. We know the distances between the sensing locations, and the robot has to visit each location exactly once until the last location is reached or the object is found. This is equivalent to a Hamiltonian Path problem that can be solved using a *Branch & Bound* algorithm. In its simplest form *Branch & Bound* can follow a LIFO strategy, a FIFO strategy or a Lowest Cost strategy.

In this paper we present a modified *Branch & Bound* algorithm as the required time to compute a path grows exponentially with the number of nodes[1]. Our solution includes the definition of a new heuristic that significantly reduces the branching factor and consequently the computing time.

Definition 2: Let L be the set of sensing location points. A location L_j strictly dominates another location L_k if and only if the two following conditions are true: (i) The area of the recognition polygon of L_j is greater than the area of the recognition polygon of L_k and (ii) the distance of $L_i \in L$ to L_j is less than the distance of L_i to L_k.



DISTANCE (L_i,L_j) < DISTANCE (L_i,L_k)
RECOGNITION_AREA (L_j) > RECOGNITION_AREA (L_k)
Therefore, BOUND (L_k)

Fig. 7. The sensing location L_k is bounded because it is strictly dominated by L_j

The algorithm only considers non dominated nodes (see Figure 7):

– For the last location along the current solution (initially just the robot's starting location) explore the possible routes (create a new level in breadth-first search)
– For each node that needs to be expanded, compute the set of locations that are not strictly dominated by others and only choose those as children. This can be done with a convex hull algorithm in $O(nlog(n))$.
– Choose the best leaf according to the cost function, and start over with the best node as root.

[1] We were not able to obtain paths in a reasonable time with more than 13 nodes in a standard laptop.

4 Simulation Results

The proposed approach was tested in 20 different maps of increasing complexity (with and without holes), some of which are shown in Figure 8.

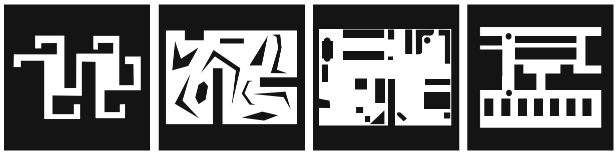


Fig. 8. Some of the maps used for testing

We measure the performance of the different sensing location algorithms in terms of: (i) number of vertexes of the map, (ii) number of guards generated, and (iii) computing time. We compared the proposed algorithms with their traditional counterpart algorithms and between each other (see Figure 9 and 10).

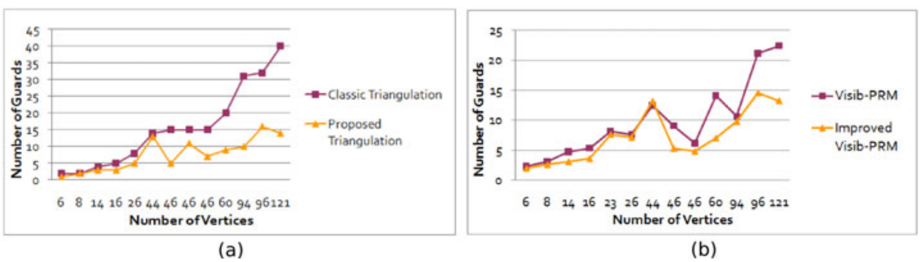


Fig. 9. For the different maps (the number of vertexes in each map is shown): (a) Comparison between the proposed triangulation algorithm and its traditional counterpart and (b) the same for the Visib-PRM algorithm

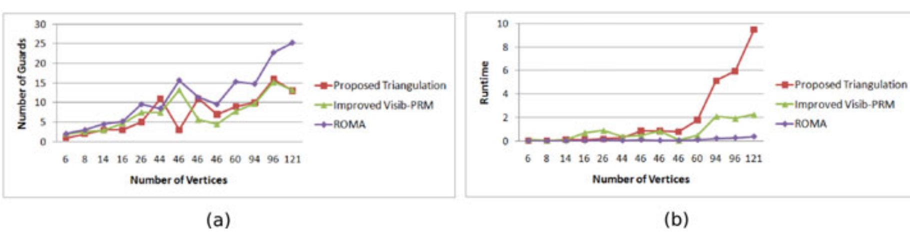


Fig. 10. For the different maps: (a) Comparison of the run-time under the number of vertexes. (b) Comparative of the number of guards under the number of Vertexes.

As can be seen from the figures, the proposed algorithms generate a smaller number of sensing points. On the other hand, when comparing the three new algorithms, it can be seen that the ROMA algorithm is the fastest one followed by the Visib-PRM algorithm. Also, the Visib-PRM algorithm offers a good compromise between the number of sensing locations and computational time.

We also performed experiments of our algorithms with infinite visibility and limited visibility for different restricted area sizes using the Visib-PRM algorithm. As expected the number of sensing points increases with more restricted visibility conditions (see Figure 11), however, it is very competitive for reasonable ranges.

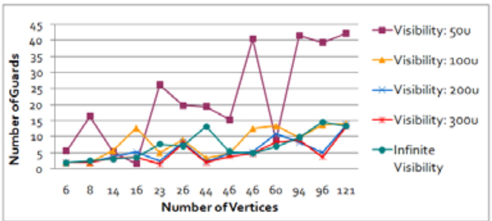


Fig. 11. Comparison between different ranges of visibility using Visib-PRM

We also compared the performance of our *Branch & Bound* algorithm against a standard *Branch & Bound* algorithm in terms of total distance of the generated paths and computing time. The results are shown in Figure 12 (note the log-scale in the run-time figure on the left).

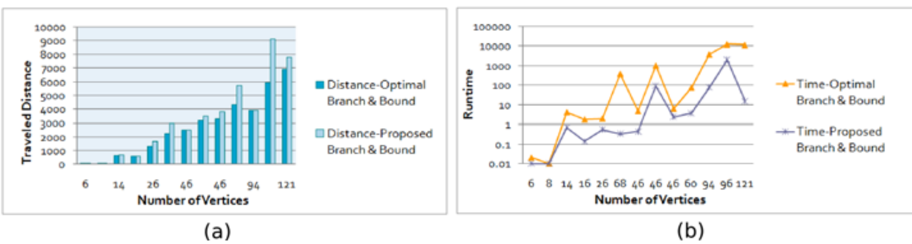


Fig. 12. Comparison between the proposed B&B algorithm and the optimal solution in terms of (a) distance and (b) time

As can be appreciated, the proposed algorithm is reasonably competitive in the total travelled distance when compared with the optimal solution (with 23% more distance in the worst case), but it is significantly faster (more than 3 orders of magnitude in the best case).

5 Conclusions and Future Work

In this paper we have introduced several strategies to solve the problem of searching an object in indoor environments. The problem involves two steps: (i) finding a set of sensing points that cover all the area and (ii) designing a strategy to visit them. We introduce the concept of recognition area to deal with more realistic conditions, and incorporate it into three new sensing point generation algorithms. For the path planning step we introduce a new heuristic to significantly reduce the branching factor and reduce the processing time. The experimental results show the benefits of the proposed algorithms in terms of processing time, realistic conditions and number of sensing points. We are currently working in the incorporation of the above strategies into a real robot. We are also planning to consider information about the prior probability of object presence inside the recognition area to modify and improve the searching process, starting by the most probable locations. This information could be used as domain knowledge or as global object features, such as color or shape, to focused the sensing locations on promising areas.

References

1. Baxter, J.L., Norman, M.: Multi-robot search and rescue: A potential field based approach
2. Casper, J., Murphy, R.: Human-robot interaction during the robot-assited urban search and rescue effort at the world trade center. IEEE Transactions on Systems, Man and Cybernetics Parts B 33(3), 367–385 (2003)
3. Eberly, D.: Triangulation by ear clipping (March 2008)
4. Hoffmann, F., Kaufmann, M., Kriegel, K.: The art gallery theorem for polygons with holes. In: The 32nd IEEE Symposium on the Foundation of Computer Science, pp. 39–48 (1991)
5. Latombe, J.C.: Robot motion planning, Boston (1991)
6. Nourbakhsh, I.R., Sycara, K., Koes, M., Yong, M., Lewis, M., Burion, S.: Human-robot teaming for search and rescue. IEEE CS and IEEE ComSoc (2005)
7. O'Rourke, J.: Art gallery theorems and algorithms. Oxford University Press Inc., New York (1987)
8. O'Rourke, J.: Computational Geometry in C. Cambridge University Press, Cambridge (1998) Hardback ISBN: 0521640105; Paperback: ISBN 0521649765
9. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: An efficient strategy for rapidly finding an object in a polygonal world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1153–1158 (2003)
10. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: A sample-based convex cover for rapidly finding an object in a 3-d environment. In: Proc. IEEE International Conference on Robotics and Automation, pp. 3497–3502 (2005)
11. Sjöö, K., Galvez-Lopez, D., Paul, C., Jensfelt, P., Kragic, D.: Object search and localization for an indoor mobile robot. Journal of Computing and Information Technology - CIT 17(1), 67–80 (2009)
12. Tovar, B., LaValle, S.M., Murrieta-Cid, R.: Optimal navigation and object finding without geometric maps or localization. In: Proc. IEEE International Conference on Robotics and Automation, pp. 464–470 (2003)