SUBMITTED PAPER
AAAI Fall Symposium Series 1998
Integrated Planning for Autonomous
Agent Architectures

Planning Robot Motions for Range-Image Acquisition and Automatic 3D Model Construction

Héctor González-Baños Jean-Claude Latombe
Department of Computer Science
Stanford University, Stanford, CA 94305
{hhg,latombe} @robotics.stanford.edu

Abstract

In this paper we explore the application of motion planning techniques in the acquisition of range-images using a mobile robot equipped with range sensors. These techniques were developed to further our goal of automatically constructing three-dimensional representations of indoor environments using a team of mobile agents. The key question we ask is the following: How can we deploy the team of robots in such a way as to keep the number of sensing operations to a minimum and do so while allowing the acquired data to be registred and merged into a single consistent representation? We present two algorithms that use a two-dimensional map to estimate the locations where sensing will be most effective. These algorithms are randomized approaches that solve an extended version of the art gallery problem. One of the algorithms is a set-coverage approach, while the other is an incremental scheme that samples the constraint of the problem instead of its domain. Additionally, we explore some of the motion planning considerations that arise when image registration issues are considered.

1 Introduction

Motion planning techniques have extended well beyond their original conception as methods for finding collision-free paths from initial to goal configurations. As more motion problems outside traditional robotics are identified and studied, the more patently it becomes that "robot" and "path" are terms not necessarily restricted to the mechanical. Nowadays we find motion planning techiques being used in so seemingly diverse topics such as assembly planning, drug design, automatic movie generation, and target tracking. These are all specific instances of spatial reasoning.

Even within robotics, however, motion planning has extended beyond basic planning under the presence of obstacles. Such is the case when visibility requirements are present or when vision sensors act themselves as end-effectors. An example of this is the task of keeping a moving target in view as it moves through a workspace. As explained in [5], when the presence of visual and motion obstructions are considered, the tracking problem trascends the machine vision context into motion control and planning domains. Another example of vision as end-effector is the hide-and-seek problem presented in [3]. Here the task is to move a team of robots in order to localize an unpredictable and arbitrary fast target with absolute certainty. This problem is very interesting from the theoretical point of view as it is a generalization of the art gallery problem when the museum guards are not restricted to be stationary.

In general, high-level vision-oriented robot operations involve important aspects of image analysis, robot localization, and dynamic control which must be addressed prior to the development of a practical system. But, even in the absence of these complications, there remains a fundamental planning problem that needs to be solved [2]. Such is the case behind the automatic exploration of an environment by a team of autonomous agents. In this problem, the task is for the team to build a visual representation of the environment, possibly for both graphic fly-through operations and physical navigation, to serve as a reconnaissance tool in architecture, archeology, and high-risk military urban missions. The question we are interested in is the following: How can we deploy the team of robots in such a way as to keep the number of sensing operations to a minimum and do so while allowing the acquired data to be registred and merged into a single consistent representation? This is the question that brings the motion planning aspect into model contruction.

Our practical approach to model construction consists on four phases: range-data acquisition and processing, texture mapping, alignment and fusion of multiple views, and sensing strategy planning. The first three phases deal with range image acquisition and registration aspects, which on themselves have motivated a fair amount of research within the framework of computer graphics and machine vision, but mostly under the assumption that sensing operations are cheap and/or executed in fixed platforms [10, 1, 7]. These three phases, however, impose constraints on the last one, and it is the main goal of this paper to show how the motion plans in model construction are highly dependent not only on the physical capabilities of the vision sensors, but on the image processing algorithms as well.

This paper deals with the sensing strategy planning phase of our model construction framework. Section 2 is the core of the paper, and contains the description of two randomized algorithms that solve for a near-optimal guard placement in an extended version of the art gallery problem that includes the physical limitations of a vision sensor. Section 3 presents the motion problem behind image alignment and how an art gallery algorithm may be modified to deal with local nature of the numerical algorithms used in image registration. Finally, in the discussion section we present the conclusions and some future extensions to our work.

2 Planning Motions for Data Acquisition

A classical problem in automatic range-image acquistion is known as the *next-best-view problem* [1, 7]: Where to place the sensor next in order to maximize the amount of information that will be added to the partial model built so far (Fig. 1)? Existing techniques are not ideally suited for robot sensors. One reason is that the next-best-view problem is inherently a *local* planning problem [4]. Obtaining a sequence of local next-best views to build a complete model may yield an excessive number of sensing operations. Even if data-acquisition is instantaneous sensing implies a cost. Part of the cost is the time spent mobilizing to all the different sensing positions. A significant additional cost, however, is the effort spent aligning 3-D and texture data with previously acquired views, a task that becomes more computationally intensive as errors in robot localization increase. This last issue is the most serious limitation of most next-best-view techniques, which are usually developed for fixed platforms with superior positioning capabilities to those found in a mobile robot.

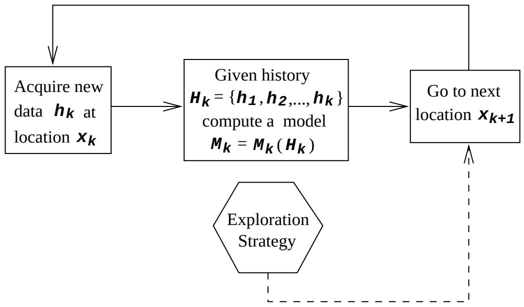


Figure 1: *Scene exploration process.*

In order to significanlty reduce the number of sensing operations we propose the following scheme. Let us assume that we are given a 2-D layout of a horizontal cross-section of the environment at approximately the height of the range sensor. An art-gallery algorithm will be able to compute the positions that must be visited to eventually see the entire 2-D environment [6]. Of course, this basic model-building approach requires additional considerations to make it work in practice:

(1) Approximating the solution to the 3-D sensing problem with the output of a planar art-gallery algorithm relies on the existence of a 2-D map. We feel, however, that this is not a serious limitation for indoor applications. Building 2-D maps can be achieved using simpler sensors and is computationally less intensive than 3-D sensing. Furthermore, the layouts for most buildings of recent construction can be obtained from the architect in electronic form.

(2) Full visibility coverage of a 2-D environment from a set of positions does not, in general, entail coverage of 3-D environment. But this is almost true for many indoor environments. Remaining "holes" are likely to be a small percentage of the entire scene which could be filled by a few additional sensing operations at locations computed using a next-best-view technique.

(3) Art-gallery algorithms use a simple "line-of-sight" visibility model: a point sees another if the line segment between them does not cross any obstacle. Real sensors, however, have incidence and range limitations. This yields new variants of the art-gallery problem.

Following the proposed scheme, we have implemented our approach. Our planning software computes a set of sensing positions from which the perimeter of a given polygonal map is observable[1]. Since the classical art-gallery problem is NP-complete, we have opted for a randomized approach.

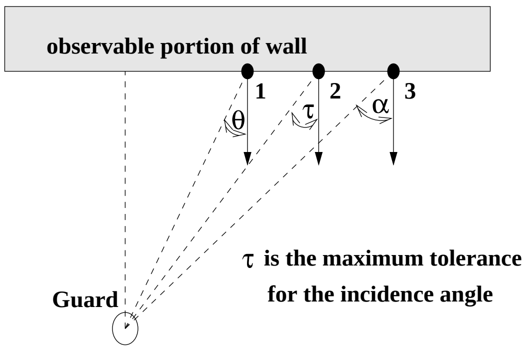
We have developed two randomized algorithms that compute near-optimal number of sensing positions from which to scan the workspace. In both algorithms we use the same visibility model: a point in a wall is *scannable* from a guard location if the angle between the line-of-sight from such point to the sensor and the surface normal is not greater than a tolerance τ, and if the distance to the sensor is within a range (r_{min}, r_{max}) (see Fig. 2a). This visibility model reflects the basic restrictions of real range-finders: a point cannot be sensed at grazing angles, nor is it visible if it lies too far or too close to the sensor.

2.1 Sampling of the interior and the greedy set-coverage algorithm

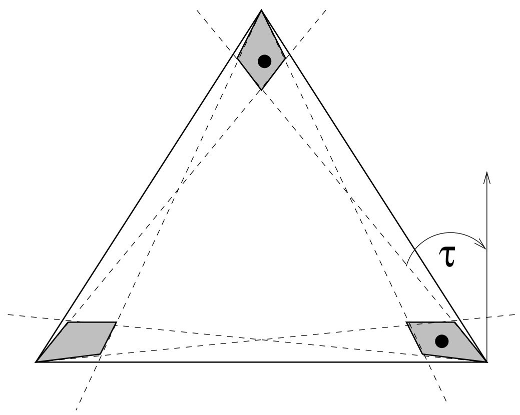
Our first approach consists in sampling the interior of the workspace in order to construct a set $\mathcal{G}$ of potential sensing (or guard) locations and select from it the subset of minimum cardinality that covers the perimeter of interest. If the kernel of the workspace (the region from which it is possible to scan the entire environment) has a non-zero measure, then with increasing probability as the number of samples grow will the set $\mathcal{G}$ contain a subset of guards with complete coverage.

Alternatively, one may try to discretize the interior and construct a grid of possible guard locations. This approach will be *resolution* complete if for every point in the perimeter there is a region of measure non-zero from which it is possible to scan it –if the region is of measure zero (i.e., it consists of a single point) then the probability of sampling such guard position is zero. It is important to note that not all workspaces are fully scannable. Narrow corners cannot be scanned completely if the tolerance τ is too small. A consequence of this is that no triangle is scannable if the tolerance is smaller than 30 degrees (see Fig. 2b).

The set $\mathcal{G}$ contains an optimal subset with high probability, but we still have to find it. For this we decompose the boundary as shown in Fig. 3a. Each boundary element is labeled, and a data structure containing the covering guards is associated with it. The set of all seen elements is



(a)



(b)

Figure 2: (a) *Incidence contraint for a range-finder. Portions of the wall are seen only if* $|\theta| \leq \tau$. (b) *Complete coverage even for a triangle may be unattainable by a single guard if incidence restrictions are considered. A triangle is scannable only if the incidence restriction is less than 30 degrees.*

called $\mathcal{X}$. The problem is then reduced to find the optimal subset $\mathcal{S}$ of $\mathcal{G}$ that covers $\mathcal{X}$ –a classic set-coverage problem.

Set-coverage is NP-hard, but a near-optimal approximation is possible in linear time. For every element $g \in \mathcal{G}$ let $\mathcal{C}(g)$ be the set of boundary elements covered by guard g. By doing a greedy selection (choosing at each step the guard with the highest coverage of the remaining uncovered boundary), we find a solution in $O(|\mathcal{C}(g_1)| + \ldots + |\mathcal{C}(g_m)|)$ steps, where the operator $|\cdot|$ is the cardinality of a set. This solution is near-optimal in the sense that it is never worse than a logarithmic factor of the true optimum.

The visible region from each guard location can

[1] In order to construct a model we need to scan walls and obstacles, hence we are interested in covering the *boundary* of the workspace, not the interior.

be computed in time $O(mn \log n)$ (with n being the number of edges in the workspace and m the number of samples). The boundary decomposition consists at most in $O(mn)$ elements, and if there are no holes it can be constructed in time $O(nm \log m)$ with a balanced tree for each edge. The highest value of $|\mathcal{C}(g)|$ is mn (a guard may cover all the boundary elements), hence an upper bound for the running time of the set-coverage algorithm is $O(nm^2)$, making this the upper bound of the entire algorithm. However, most interior locations observe only a small fraction of the boundary elements under realistic range and incidence values. Hence, the average running time of the algorithm is significantly less than the worst case time.

Figure 4 shows an example of sensor positions computed by our software under incidence constraints. The algorithm is summarized below:

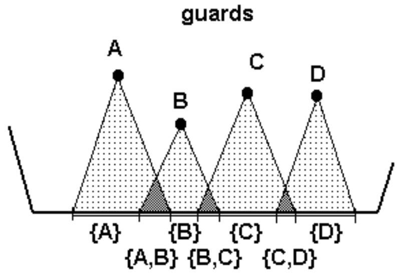
Algorithm *Interior samp. with greedy select.*
Input:
1.- A polygon description $\mathcal{P}$
2.- A set of vis. constraints $\{\tau, r_{mn}, r_{mx}\}$
3.- The number of interior samples
Output: A near-optimal set $\mathcal{S}$ of guard positions covering the workspace boundary

1. Sample the interior of $\mathcal{P}$ m times and contruct the set of potential guards $\mathcal{G}$.
2. For every sample $g \in \mathcal{G}$ compute the visible portion of the perimeter $\mathcal{V}(g; \tau, r_{min}, r_{max})$.
3. Compute the boundary decomposition $\mathcal{X}$, and associate to each element a data structure with the information of the guards covering it.
4. Solve the set-coverage problem with a greedy algorithm: find the near-optimal subset $\mathcal{S}$ of $\mathcal{G}$ that covers $\mathcal{X}$.

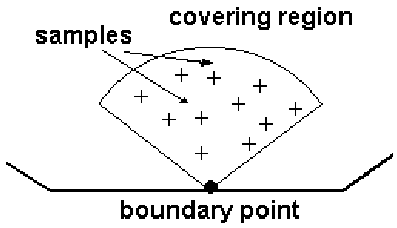
2.2 The dual-sampling scheme and the incremental algorithm

The previous algorithm has a quadratic dependancy on the number of samples, which is an inconvenience as this limits the degree of resolution we may have in practice. An alternative is to avoid sampling regions of the interior that cannot cover the boundary. This can be achieved by sampling the constraints of the problem (the points we *have* to cover) instead of its domain.

The second planner is incremental and incorporates a dual sampling scheme. First a boundary point $p \in \mathcal{B}$ is selected and the region $\mathcal{O}(p)$



(a)



(b)

Figure 3: (a) *Each boundary element in the decomposition is a section of an edge along with the set guards covering it; (b) in the dual-sampling scheme a boundary point is selected and the region from which the point is visible is sampled.*

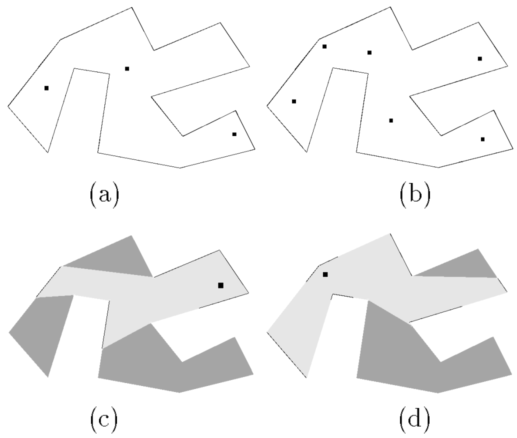


Figure 4: *Computed guard positions in a 2-D map: (a) with no constraints three guards are needed; (b) with a minimum required incidence of* 60 *degrees six guards are neeeded. The portions of walls seen from two AO positions are shown in (c)-(d).*

from which such point can be observed is calculated. The region $\mathcal{O}(p)$ is then sampled as shown in Fig. 3b. The sampled position with the highest coverage is retained (q^*) as a new guard, which reduces the length of the unobserved perimeter. A new boundary point is selected from the unseen perimeter and the process is then repeated.

The advantage of an incremental algorithm is that a user may prescribe a degree of coverage gain as a stopping criterion: i.e., *stop* when the reduction of the unobserved perimeter is not a significant fraction of the total. This is very useful, for in most cases few guards cover almost the entire perimeter but many more are required to cover narrow corners that compose a small fraction of the total boundary (law of diminishing returns). Workspaces that are not fully scannable can be dealt with using this scheme.

The unobserved portion of the perimeter is made of collections of disjointed sections per edge of the polygon. At iteration k we have already placed k guards, and the number of disjoint sections is in the worst case $O(k)$ per edge. The data structure representing the unseen perimeter can be constructed using balanced trees in time $O(nk \log k)$ (where n is the number of edges).

Sampling the boundary once can be done in time $O(\log n + \log k)$, and finding the region $\mathcal{O}(p_k)$ can be done in time $O(n \log n)$. Computing the portion of the boundary observable by a point in $\mathcal{O}(p_k)$ takes $O(n \log n)$, and intersecting this with the unseen perimeter elements takes $O(n \log k)$. Therefore, selecting the best q^*_k takes a time $O(\mu n \log n + \mu n \log k)$, where μ is the number of samples in region $\mathcal{O}(\cdot)$ per iteration. Overall, the running time of the algorithm is bounded by $O(\mu n_g n \log n_g n)$, where n_g is the number of guards found by the algorithm.

The algorithm is summarized below:

Algorithm *Placement with dual sampling*
Input:
1.- A polygon description $\mathcal{P}$
2.- A set of vis. constraints $\{\tau, r_{mn}, r_{mx}\}$
3.- The number of samples per iteration μ

Output: A near-optimal set $\mathcal{S}$ of guard positions covering the workspace boundary

1. Select a point p from the unseen portion of the perimeter and compute the region $\mathcal{O}(p)$ from which such point is visible.
2. Sample $\mathcal{O}(p)$ μ times, and select the position with highest coverage. Add this position to the set $\mathcal{S}$.
3. Update the data structure representing the unseen portion of the perimeter.
4. Repeat the algorithm until the unseen boundary is neglectable.

3 Planning for Image Alignment

As mentioned in the previous section, due to errors in robot localization different views have to be aligned prior to the construction of a consistent representation. It is even possible that the range-image sequence was captured using different agents. Reliable alignment techniques are essential in this context, which by neccessity require some degree of overlap between views.

Alignment methods are mainly optimization procedures that, like most numerical optimizers, may converge to local minima. Visually this translates to situations where a pair of views is aligned incorrectly. This problem is readily solved if the 3-D representation is constructed under human assistance: simply ask the user to closely line up the views and rerun the aligment procedure. The algorithm will then do the final "zippering" [9]. However, if the operation is to be done automatically, there is a strong coupling between the alignment step and the exploration strategy. The strategy should consider alignment issues as part of the constraints of the problem, in addition to sensor limitations and constraints over motion.

Art-gallery algorithms strive to minimize the number of positions where to place the observers. However, 3-D/texture sensing at these positions can yield partial views that have little or no overlap. Yet, we believe that the typical art-gallery algorithms can be extended to incorporate part of the limitations of alignment methods.

Our approach is to assume that it is possible to bound the convergence radius of the alignment algorithm. This radius is a measure of how disjointed two similar views can be and still guarantee a match, and, along with the sensor physics, determine how far a robot may wander from succesive sensing positions and still be able to align two views of the same scene. The set of all workspace locations within this distance from a point is the *registrable vecinity* of a point.

Incorporating alignment issues into the basic art-gallery problem imply that the guard locations computed by our planners should have their closest neighbors within their registrable vecinity. A way to achieve this in practice is to use additional guards to aid the registration process. For a single robot, the procedure is the following: take the set of guard locations and generate a path by connecting the most recently visited location with closest unvisited one[2], if a pair of subsequent locations is

[2] If one seeks to minimize the path length then we must

not within the registrable vecinity add the necessary number of guards to allow for registration.

Our approach is currently far from optimal. A complete optimization should consider the registration constraint, the cost of sensing, the cost of moving, and the cost of planning simultaneously. We believe the planning problem then becomes closely connected with the *watchman route problem* [8]: Find the shortest closed path from which the entire workspace is visible. Surprisingly, this problem has been shown to be solvable in polynomial time although the basic art-gallery problem is NP-complete.

We our currently studying randomized approaches to solve for a watchman route under registration constraints.

4 Conclusion

High-level vision-oriented operations involve important motion planning problems that have received little attention so far. In this paper we presented novel planning approaches to enhance the acquisition of range images and the construction of efficient visual representations. Specifically, we have explored ways in which sensor limitations and image alignment issues could be incorporated into practical planners. We have developed two randomized planners with good results. One of the algorithms is incremental, allowing the possibility of varying the number guards as the requirements get stricter.

Our future work will focus in developing better algorithms to incorporate registration issues. We consider the watchman route problem to be highly related to this topic.

Acknowledgements

This research was funded by subcontract 5-31997-A of the University of Pennsylvannia NSF grant IIS-97-11380, by subcontract 5-23542-A of the University of Pennsylvannia MURI grant DAAH04-96-1-0007, and by DARPA contract DAAE07-98-C-L027.

solve a *travelling salesman* problem. Given that the problem is NP-hard, it may not be desirable to go to such lengths if the actual exploration time is mainly spent scanning and processing images.

References

[1] J. E. Banta, Y. Zhien, X. Z. Wang, G. Zhang, M. T. Smith, and M. A. Abidi. A "best-next-view" algorithm for three-dimensional scene reconstruction using range images. In *Proc. SPIE vol. 2588*, pages 418–29, 1995.

[2] H. González-Banos and et Al. Motion planning with visibility constraints: Building autonomous observers. In *Proc. 1997 International Symposium of Robotics Research*, 1997.

[3] L. Guibas, J.C. Latombe, S.M. LaValle, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. In *Proc. 5th Workshop on Algorihtms and Data Structures (WADS'97)*, pages 17–30. Springer Verlag, 1997.

[4] K. Kakusho, T. Kitahashi, K. Kondo, and J.-C. Latombe. Continuous purposive sensing and motion for 2d map building. In *Proc. IEEE Int. Conf. Systems, Man, & Cybernetics*, pages 1472–1477, 1995.

[5] S.M. LaValle, H. González-Banos, C. Becker, and J.C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proc. 1997 IEEE Int'l Conf. Robotics & and Automation*, April 1997.

[6] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, NY, 1987.

[7] R. Pito. A solution to the next best view problem for automated cad model acquisition of free-form objects using range cameras. Technical Report 95-23, GRASP Lab, University of Pennsylvania, May 1995.

[8] T. Shermer. Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399, September 1992.

[9] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proc. ACM SIGGRAPH*, pages 311–318, 1994.

[10] L. Wixson. Viewpoint selection for visual search. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 800–805, 1994.