A Randomized Art-Gallery Algorithm for Sensor Placement

Héctor González-Banos
CS Robotics Laboratory
Stanford University
Stanford, CA 95405, USA
hhg@robotics.stanford.edu

Jean-Claude Latombe
Department of Computer Science
Stanford University
Stanford, CA 95405, USA
latombe@cs.stanford.edu

ABSTRACT

This paper describes a placement strategy to compute a set of "good" locations where visual sensing will be most effective. Throughout this paper it is assumed that a *polygonal 2-D map* of a workspace is given as input. This polygonal map — also known as a *floor plan* or *layout* — is used to compute a set of locations where expensive sensing tasks (such as 3-D image acquisition) could be executed. A map-building robot, for example, can visit these locations in order to build a full 3-D model of the workspace.

The sensor placement strategy relies on a randomized algorithm that solves a variant of the *art-gallery problem* [12, 15, 19]: Find the minimum set of guards inside a polygonal workspace from which the entire workspace boundary is visible. To better take into account the limitations of physical sensors, the algorithm computes a set of guards that satisfies incidence and range constraints. Although the computed set of guards is not guaranteed to have minimum size, the algorithm does compute with high probability a set whose size is at most a factor $\mathrm{O}(\log(n+h)\cdot\log(c\log(n+h)))$ from the optimal size c, where n is the number of edges in the input polygonal map and h the number of obstacles in its interior (holes).

1. INTRODUCTION

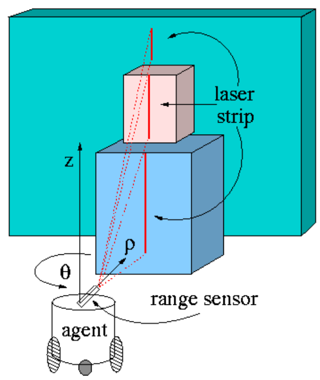
One of the most basic tasks for a mobile robot is to build a representation of the environment using its sensors. The model constructed by the robot may be the goal of a reconnaissance or exploration mission, or instead can be used to facilitate subsequent tasks to be performed by the robot or other agents. Sometimes a 2-D model of the workspace is sufficient, in which case a system like the one described in [9] can be used to build such a representation. In other cases — most notably in systems that allow remote users to "fly-trough" a virtual representation of the environment — it is necessary to efficiently acquire full 3-D and texture data in order to synthesize a realistic visual and/or geometric model. While the environment can be represented in several ways (e.g., geometric primitives, image-based rendering, or light-field models [10]), one thing is certain: visual sensors have to be placed throughout the workspace in order to acquire this representation. An important question is thus the following: How should we place the necessary sensors (cameras, range-finders, etc.) to gather the information required to build the visual/geometric model as efficiently as possible? Or in the case of a mobile robot equipped with a range sensor, which locations should the robot visit in order to collect the necessary data?

Consider the robot/sensor configuration shown in Figure 1(a). It consists of a laser range-sensor mounted on top a mobile robot. This sensor acquires distance information along a vertical cross-section of the environment. By performing a rotational sweep, and acquiring multiple scans during the sweep, the robot is able to capture a 3-D image of the environment at a particular location (Figure 1(b)). This 3-D image consists of a set of points, and its resolution is a function of the rotational speed used during the sweeping operation (a slow sweep produces a high resolution image). As a result, the acquisition of a high-quality 3-D image is a relatively costly operation, and post-processing this image can be computationally expensive because the set of points may be large.

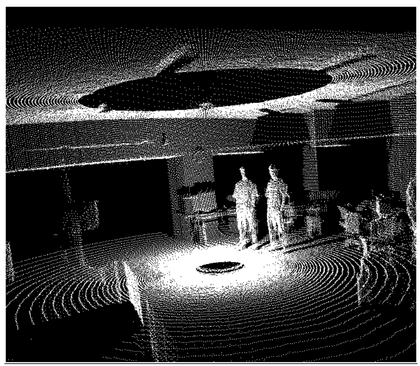
Suppose that we want to build a high-quality 3-D model of a large workspace. In order to expedite this operation, the problem is now to minimize the number of locations where the robot should perform the rotational sweeps. Assume that a polygonal 2-D map of the environment is available. This paper proposes to compute the smallest set of locations in the 2-D map from which the entire polygonal contour is visible. Once this set is computed, the robot is then sent to these locations to acquire 3-D images that are later merged into a complete model of the workspace. If necessary, the robot may later be sent to additional locations during a refinement stage to fill any remaining gaps in the model.

The purpose of this paper is not to investigate 3-D modeling. Model construction using range-sensors is a well-studied engineering problem (e.g., see [1, 7, 5, 11, 14, 20]). Instead, our goal is to investigate algorithms to compute a set of good sensor locations in a polygonal model of a workspace, locations at which expensive sensing operations will later be executed. This placement problem is closely related to the *art-gallery problem* [12]: Find the minimum set of guards such that any point in an art gallery is visible from at least one guard. Practical range sensing, however, introduces two complications: the operational range of most sensors is lower- and upper-bounded, and range finders cannot re-

SCG'01, June 3-5, 2001, Medford, Massachusetts, USA.



(a)



(b)

Figure 1: A robot as a 3-D sensor: (a) a robot equipped with a laser range-finder may capture 3-D information by sweeping the environment with a plane of light; (b) an example of a range-image captured using rotational sweeps.

liably detect surfaces oriented at grazing angles relative to the sensor's line of sight.

The algorithms presented in this paper acknowledge these practical visibility restrictions by solving a variant of the art-gallery problem. Although the art-gallery problem is NP-hard, our strategy produces a set of guards that with high probability is at most a factor $O(\log(n+h) \cdot \log(c \log(n+h)))$ from the optimal size c, where n is the complexity of the input polygonal map and h is the number of obstacles in its interior (holes).

2. EXTENDED ART-GALLERY PROBLEM

The art-gallery problem is now a classic problem in the study of algorithms. Although many extensions to the traditional problem exist [15, 19], most of the past work assumes a classical "line-of-sight" visibility model: one point sees another if the line segment between them does not intersect any object.

Here, we restrict this classic notion of visibility to account for range and incidence restrictions. Our visibility model is as follows:

Definition 2.1. (Visibility under Incidence and Range Constraints). *Let the open subset $\mathcal{W} \subset \Re^2$ describe the workspace layout. Let $\partial\mathcal{W}$ be the boundary of $\mathcal{W}$. A point $\boldsymbol{w} \in \partial\mathcal{W}$ is* visible *from a point $\boldsymbol{q} \in \mathcal{W}$ if the following conditions are true:*

1. Line of sight constraint*: The open line segment $S(\boldsymbol{w}, \boldsymbol{q})$ joining $\boldsymbol{q}$ and $\boldsymbol{w}$ does not intersect $\partial\mathcal{W}$.*
2. Range constraint*: $d_{min} \leq d(\boldsymbol{q}, \boldsymbol{w}) \leq d_{max}$, where $d(\boldsymbol{q}, \boldsymbol{w})$ is the Euclidean distance between $\boldsymbol{q}$ and $\boldsymbol{w}$, and $d_{min} \geq 0$ and $d_{max} > d_{min}$ are input constants.*
3. Incidence constraint*: $\angle(\boldsymbol{n}, \boldsymbol{v}) \leq \tau$, where $\boldsymbol{n}$ is a vector perpendicular to $\partial\mathcal{W}$ at $\boldsymbol{w}$, $\boldsymbol{v}$ a vector oriented from $\boldsymbol{w}$ to $\boldsymbol{q}$, and $\tau \in [0, \pi/2]$ is an input constant.*

Consider the robot/sensor configuration of Figure 1(a). This implements a mobile omnidirectional 3-D sensor, subject to range and incidence restrictions. The robot/sensor pair can be modeled with the definition stated above. With slight modifications, Definition 2.1 can also model other sensors restricted to move in a plane. For example, when a standard CCD video camera is used to capture images for image-based rendering, we would add the condition that the segment $S(\boldsymbol{w}, \boldsymbol{q})$ lies inside a cone oriented along the camera axis to account for the fact that the camera is not omnidirectional. For stereo systems, a point in the $\partial\mathcal{W}$ is visible if Definition 2.1 holds for *both* cameras. Etc.

In order to construct a 3-D model of the environment, it is necessary to scan walls and other objects throughout the workspace. Therefore, we are interested in solving an art-gallery problem that requires the guards to cover only the *boundary* of the 2-D layout, including the boundary of any "holes" produced by obstacles in the workspace. This extended art-gallery problem is defined as follows:

Problem 2.1 (Extended Art-Gallery Problem). *For a given layout $\mathcal{W} \subset \Re^2$, compute the minimum set of guard locations $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ in $\mathcal{W}$, such that every point $\boldsymbol{w} \in \partial\mathcal{W}$ is visible from at least one point in $\mathcal{G}$ under Definition 2.1.*

A variant of the problem is to require coverage of a large fraction of $\partial\mathcal{W}$ instead of the entire boundary. For the reminder of this paper, it is assumed that $\mathcal{W}$ is a polygonal subset of $\Re^2$ bounded by a list of polygons $\partial\mathcal{W}$ (the outer boundary and the list of holes).

Effect of the constraints

Not all layouts can be fully covered under Definition 2.1, while others may require an infinite number of guards. For instance, narrow corridors cannot be covered by any set of guards if d_{min} is too large for the choice of τ (Figure 2(a)). Likewise, walls meeting at an acute angle cannot be covered by a finite set of guards (Figure 2(b)).

An effect of the incidence constraint is that a layout does not admit a finite solution if the internal angle of any of its vertices is less than $90 - \tau$ degrees. This fact can be easily inferred from the situation shown in Figure 2(c). An interesting consequence is that no triangular shape admits a finite cover when $\tau < 30$ deg.

Some optimal covers may require extremely precise sensor positioning to carry out in practice, even for sensors without range or incidence restrictions. Consider the layout in Figure 3. The optimal cover consists of a single guard located at the center of the layout, but the sensor has to be placed exactly at this location. Any deviation will leave a

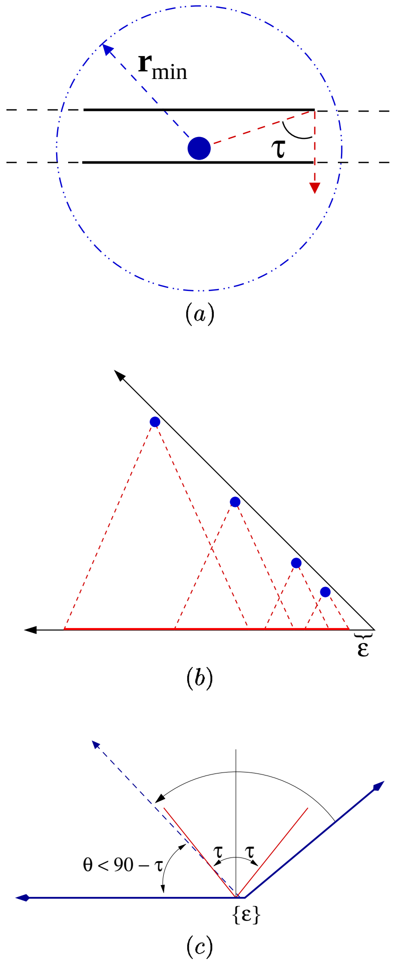


Figure 2: Examples of workspaces that cannot be completely covered under incidence and range constraints: (a) narrow corridors may be impossible to cover if d_{min} is too large for a given τ; (b) under incidence limitations there always remains an unseen section ϵ for walls meeting at a very acute angle, regardless of the number of guards; (c) a workspace cannot be fully covered if the angle of any of its vertices is smaller than $90 - \tau$ degrees.

section of ∂W uncovered. Optimal covers that exhibit this behavior have measure zero. In practice, they are difficult or impossible to achieve by mobile sensors, and they cannot be computed using the random-sampling techniques proposed later in this paper (since the probability of sampling a subset of measure zero is null).

3. A RANDOMIZED STRATEGY

The classical art-gallery problem is NP-hard [15], and Problem 2.1 is at least as difficult. Thus, we can only hope for an approximate solution. Our approximation algorithm first samples the workspace $\mathcal{W}$ at random to construct a relatively large set $\mathcal{G}_{sam}$ of guard candidates. Afterwards, it selects the subset $\mathcal{C} \subseteq \mathcal{G}_{sam}$ of minimum cardinality among all subsets covering $\partial\mathcal{W}$ as the solution to Problem 2.1.

Hence, the basic idea is to use random sampling to transform Problem 2.1 into a set cover problem. A near-optimal

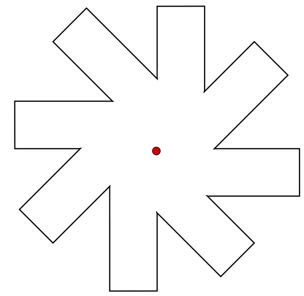


Figure 3: This workspace is fully covered using a single guard located at the center (here $d_{min} = 0$, $d_{max} = \infty$, and $\tau = 90$ deg), but any deviation from this position will leave a section of $\partial\mathcal{W}$ uncovered.

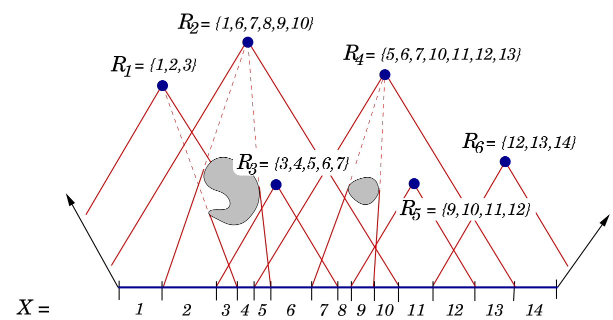


Figure 4: Each edge of the workspace is decomposed into cells, such that all points within the same cell are seen by exactly the same subset of $\mathcal{G}_{sam}$. Each cell is labeled with an integer and grouped under X. A set R_i is defined as the set of cells visible from g_i.

solution to the later problem can be computed using a greedy algorithm. This solution in turn becomes an approximate solution to Problem 2.1 as well. We will see, however, that traditional bounds for the greedy approximation to the optimal set cover do not produce a useful competitive ratio for the approximation to the original art-gallery problem. This will lead us in Section 5 to consider a different set-cover algorithm.

3.1 Basic Algorithm

Let $\mathcal{G}_{sam} \subset \mathcal{W}$ be a set of m locations selected at random in $\mathcal{W}$. For every edge $e \in \partial\mathcal{W}$, compute the fraction seen by each element in $\mathcal{G}_{sam}$. The arrangement of all covered portions decompose each edge into cells such that all points in the same cell are seen by exactly the same subset of $\mathcal{G}_{sam}$ (see Figure 4). Next, enumerate all cells in the decomposition of $\partial\mathcal{W}$ and group them under the ground set $X = \{1, 2, \ldots, l\}$, where l is the number of cells. Build the set family $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$, where $R_i = \{x \in X \mid g_i \text{ covers } x\}$ is the subset of all the elements from X that are covered by $g_i \in \mathcal{G}_{sam}$. By construction, the union of all the sets in $\mathcal{R}$ is equal to X. Let the set system $\Sigma = (X, \mathcal{R})$ represent the sampled (or discretized) instance of Problem 2.1.

In Section 4, we will show that under some conditions $\mathcal{G}_{sam}$ covers $\partial\mathcal{W}$ (i.e., any point in $\partial\mathcal{W}$ is seen from at least one member of $\mathcal{G}_{sam}$), and that with high probability the optimal set of guard locations $\mathcal{G}$ is contained in $\mathcal{G}_{sam}$. Problem 2.1 is henceforth reduced to that of finding a *set cover* of minimum size, where a set cover is a sub-collection $\mathcal{C} \subseteq \mathcal{R}$, such that the union of all the R_i's in $\mathcal{C}$ equals X. The size of $\mathcal{C}$ is the number of sets in $\mathcal{C}$.

The set cover problem is NP-hard, and its corresponding decision problem is NP-complete [4]. Thus, our problem remains NP-hard. However, algorithms for finding near-optimal set covers have been well studied. A particularly appealing one due to its simplicity is the GREEDY algorithm: Find the set $R \in \mathcal{R}$ with largest cardinality, remove this set from $\mathcal{R}$, and delete the contents of R from X and from the remaining covering sets $\mathcal{R} \setminus \{R\}$. In the next iteration, select again the set with largest cardinality and repeat the process until the set X is empty.

Figure 5 shows examples of computed guard locations using our strategy. The full algorithm is summarized below:

Algorithm *Randomized Art-Gallery Algorithm*
Input: 1.- Polygonal Region $\mathcal{W}$
2.- Visibility constraints $\{\tau, r_{mn}, r_{mx}\}$
3.- An integer m (the number of samples)
4.- Function SET COVER that computes a near-optimal set cover (e.g. GREEDY)

Output: A near-optimal set of guards $\mathcal{C}$ covering $\partial\mathcal{W}$
1. Construct the set of guard candidates $\mathcal{G}_{sam}$ by sampling $\mathcal{W}$ m times uniformly at random.
2. For every $g \in \mathcal{G}_{sam}$, compute the portions of the edges in $\partial\mathcal{W}$ that are visible under Definition 2.1.
3. Compute the decomposition X of $\partial\mathcal{W}$ and the set family $\mathcal{R} = \{R_1, \ldots, R_m\}$.
4. Invoke SET COVER on $\Sigma = (X, \mathcal{R})$, and return the computed cover $\mathcal{C}$ as the near-optimal set of guards.

3.2 Computational Cost

Step 1. Verifying that a sample point g falls in $\mathcal{W}$ can be done in time $O(n)$. It is possible to reduce this time to $O(\log n)$ by using a hierarchical triangulation scheme whose construction takes $O(n \log n)$ of preprocessing time [17]. To construct $\mathcal{G}_{sam}$, merely generate a point uniformly at random inside the bounding box of $\mathcal{W}$, verify whether it falls inside $\mathcal{W}$ (discard it otherwise), and repeat until m valid samples are generated.

A quicker way of generating $\mathcal{G}_{sam}$ is to construct a pseudo-uniform distribution of sample points. First, $\mathcal{W}$ is triangulated, which can be done in $O(n \log n)$ time by a line-sweep algorithm [2]. Afterwards, the m sample points are distributed among all triangles according to their relative areas. Each triangle is then uniformly sampled, which can be done in $O(1)$ per sample point. The total cost of this sampling scheme is $O(n \log n + m)$.

Step 2. The visibility region from each point $g \in \mathcal{G}_{sam}$ can be computed in time $O(n \log n)$ [13] by a ray-sweep algorithm. Under Definition 2.1, the edges of the classic visibility polygon must be cropped to satisfy the sensor constraints, but this operation takes $O(1)$ per edge. Therefore, even under the presence of incidence and range constraints, the total cost of Step 2 is $O(mn \log n)$.

Step 3. The number of cells in the boundary decomposition is $O(m(n+h))$, where h is the number of holes in $\mathcal{W}$. Indeed, the subset of $\partial\mathcal{W}$ visible from a sample point s consists of up to $2(n+h)$ separate pieces (under no incidence or range restrictions the visible subset contains at most $n+h$ segments, adding upper-range and incidence

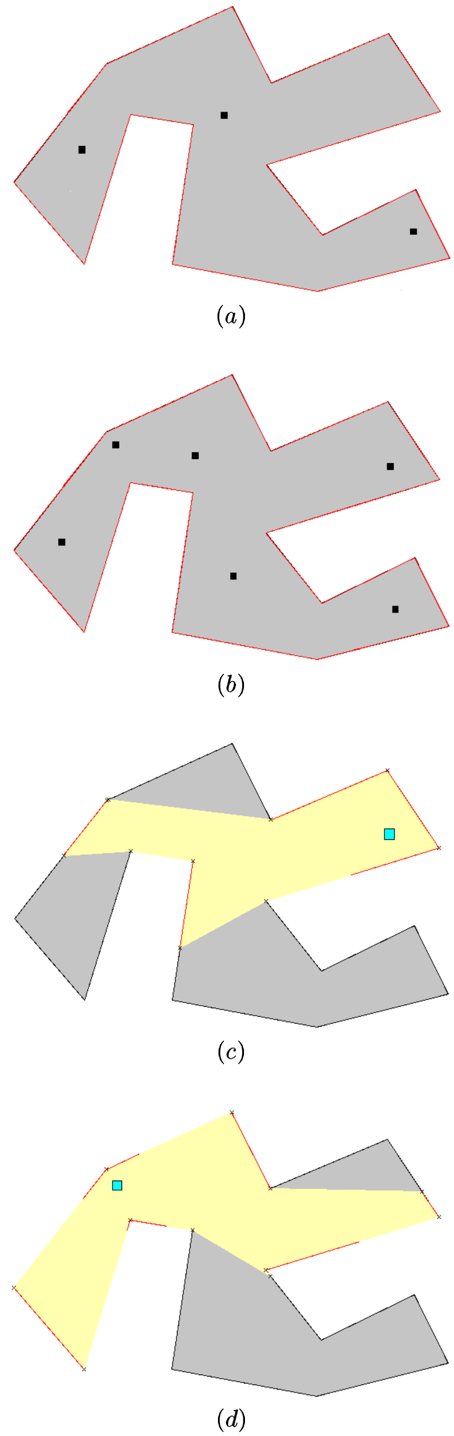


Figure 5: Computed guard positions in a 2-D map: (a) with no constraints three guards are needed; (b) with a minimum required incidence of 60 degrees, six guards are needed. The portions of walls seen from two guard locations are shown in (c)-(d).

constraints leaves the total unchanged, but the minimum range restriction may split a visible segment into two separate pieces). The endpoints of all visible pieces must be sorted along the edges in $\partial\mathcal{W}$ in order to construct the arrangement X. This is done in time $O(nmh\log(mh))$, for all edges in $\partial\mathcal{W}$, by using n balanced trees to maintain a sorted sequence of intervals along each edge. In practice, however, the usual size of the decomposition is much smaller than $O(m(n+h))$. Most samples do not influence the decomposition of an edge either because they see it entirely, or because they do not see it at all.

Step 4. The computational cost of this step depends on the function SET COVER. Let GREEDY be our choice for the function SET COVER. Then the cost is $O(|R_1|+|R_2|+\ldots+|R_m|)$. This is because, as function GREEDY executes, every element in X must be deleted along with all its representatives in $\mathcal{R}$. Therefore, every set in $\mathcal{R}$ is eventually deleted, and the sum of their cardinalities gives the cost for Step 4.

A very generous upper bound for the cost of GREEDY is $O(m|R_L|)$, where R_L is the set in $\mathcal{R}$ with largest cardinality. This in turn is bounded by $O(m^2(n+h))$, because there are at most $O(m(n+h))$ boundary elements in X and R_L cannot be larger than X. However, most of the samples in $\mathcal{G}_{sam}$ observe only a fraction of the elements in X, and the size of X is usually much smaller than this upper bound. Therefore, the running time of GREEDY is commonly below $O(m^2(n+h))$.

3.3 Quality of the Solution using GREEDY

Is the approximate solution computed using GREEDY close to optimal? Assume for the time being that m is large enough and that the optimal solution to the art-gallery problem (Problem 2.1) is contained in $\mathcal{G}_{sam}$ (see Section 4). Under this assumption, the problem is now just a matter of extracting the right subset from $\mathcal{G}_{sam}$. Thus, the quality of the set cover approximation defines the quality of the overall algorithm.

The set cover solution computed by GREEDY can be proven to have an approximation ratio bounded by $(1+\log|R_L|)$ [6], where R_L is the largest set in $\mathcal{R}$. At first glance this logarithmic factor looks good, but in reality it is not very useful. R_L can be as large as $O(m(n+h))$, which results in an approximation factor that is a function of the number m of sample points. Not only m can be very large, but it is not a fixed constant for a given workspace. While a large m will increase the probability that $\mathcal{G}_{sam}$ contains the optimal solution to Problem 2.1, this will also reduce the quality of the set cover.

The problem lies with the bound for the greedy solution. In most applications, the size $|R_L|$ is a small fraction of the size of the ground set X, and so the greedy set cover is guaranteed to be close to optimal. Unfortunately, this is not the case here — R_L can be as large as X. Are there better general bounds for the greedy set cover? It turns out that the ratio bound has been proved to be *exactly* $\log(|X|) - \log\log(|X|) + \Theta(1)$ (see [16]). This means not only that the greedy solution falls within a logarithmic factor of $|X|$ from the optimal, but that there are instances where this bound is actually achieved.

A large X, however, does not imply that the set-cover problem is particularly "complicated." In fact, we have observed during experiments that the randomized art-gallery algorithm always produces a reasonable number of guards once the sampling becomes dense enough. If the sampling is made denser, the solution remains about the same and it never becomes worse. This is evidence that the set system $\Sigma = (X, \mathcal{R})$ possesses additional structure that GREEDY fails to exploit during the computation of the set cover. In Section 5 we will describe a different set-cover algorithm that exploits this structure.

4. SAMPLING

In general, the optimal solution to the sampled art-gallery problem is not the solution to the original problem (Problem 2.1). Indeed, if the sample set is poor (e.g., it contains too few sample points and/or has an incorrect distribution), or the workspace admits no finite solution, or the optimal cover has measure zero, an optimal set of guards cannot be obtained by solving the sampled problem.

The advantage of sampling is that it usually produces satisfactory guard placements at a small cost even for "impossible" cases. But more importantly, most workspaces can be solved in this way with high probability if the sampling is dense enough. That is, the probability that $\mathcal{G}_{sam}$ contains the optimal set of guards quickly approaches 1 under most scenarios. This is true because the optimal solution for most problems can be perturbed slightly and remain optimal. We call this property the *elasticity* of the solution.

We have seen in Section 2 that some solutions to the art-gallery problem require perfect positioning of the sensor to be executed in practice (see Figure 3). This type of solutions cannot be found by randomized sampling, and in fact solutions like this are usually undesirable in engineering applications. If we focus only on those solutions that do not require perfect positioning, it is then possible to give a qualitative assessment of the efficiency of the sampling process.

4.1 Elastic Solutions

Let $\mathcal{B}_\delta(g) = \{p \in \Re^2 \mid d(p,g) \leq \delta\}$, where $d(\cdot,\cdot)$ is the Euclidean distance between two points. The set $\mathcal{B}_\delta(g)$ is simply an open ball centered at g. A set of guards $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ is said to cover $\mathcal{W}$ with elasticity δ if: (1) every point $w \in \partial\mathcal{W}$ is visible from at least one point in $\mathcal{G}$ under Definition 2.1, and (2) the same is true for any other set of guards $\mathcal{G}' = \{g'_1, g'_2, \ldots, g'_n\}$ such that $g'_i \in (\mathcal{B}_\delta(g_i) \cap \mathcal{W})$ for $i = 1, 2, \ldots, n$.

The optimal elastic solution to Problem 2.1 is the collection $\mathcal{G}$ of minimum size that is also elastic for some $\delta > 0$. Note that given two covers $\mathcal{G}$ and $\mathcal{G}'$, with $|\mathcal{G}| < |\mathcal{G}'|$ and elasticities $\delta < \delta'$, $\mathcal{G}$ is closer to optimality even if it is not as elastic as $\mathcal{G}'$. Therefore, the open balls around the guards in an optimal elastic cover do not intersect with each other, otherwise we could construct a better cover of smaller size and smaller elasticity.

Suppose there is an optimal cover $\mathcal{G}$ of size c with elasticity δ, and suppose that we sample the workspace $\mathcal{W}$ a number of m times uniformly at random. What will be the probability that all the balls around the guards of $\mathcal{G}$ are sampled? This probability is given by the following equation:

$$P(c) = \sum_{k=0}^{c} \binom{c}{k} (-1)^k (1-k\sigma)^m, \qquad (1)$$

where σ is the area of $\mathcal{B}_\delta(\cdot)$ normalized with respect to the area of $\mathcal{W}$. This equation can be simplified when m is much

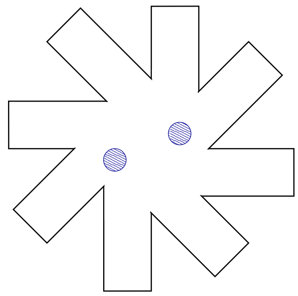


Figure 6: This workspace can be fully covered using a single guard located at the center, but the best elastic cover is of size 2 (here $d_{min} = 0$, $d_{max} = \infty$, and $\tau = 90$ deg).

larger than c and σc is a small fraction of the total area:

$$P(c) \approx (1 - (1 - \sigma)^m)^c, \text{ for } m \gg c. \qquad (2)$$

The expression above is simply the probability of sampling every ball in the cover when we have a very large supply of samples to toss.

Since the number of samples m appears as an exponent of the factor $(1 - \sigma)$, equation (2) approaches 1 very quickly. Therefore, with high probability, the set of samples $\mathcal{G}_{sam}$ will contain a set of guards that is equivalent to the optimal set $\mathcal{G}$. Afterwards, by solving the set cover problem described in the previous section, this optimal set of guards is extracted from $\mathcal{G}_{sam}$. In other words, we may assume that with high probability the optimal set cover $\mathcal{C}$ for the set system $\Sigma = (X, \mathcal{R})$ is also the optimal set of guards $\mathcal{G}$ for Problem 2.1.

There are three caveats to the argument presented here. First, increasing the value of m reduces the quality of the set cover computed using algorithm GREEDY. This problem can be circumvented by using instead the algorithm presented in the next section. What is inevitable is the impact of m on the running time of any set cover algorithm. Therefore, the choice of m should be selected wisely (a choice that is perhaps more a craft than a science).

The second caveat is that the best elastic solution is not necessarily the best solution in the traditional sense. Again, for the workspace shown in Figure 3, the best elastic solution is a set of size 2 (not 1) as shown in Figure 6. This can be a major problem in an application requiring a set of guards of absolute minimum size. But it can be an advantage on applications where guard covers with measure zero are infeasible to carry out in practice.

The final caveat is that an optimal elastic cover may not exist at all. As explained in Section 2, some workspaces cannot be fully guarded (see Figure 2). The question is then if we can detect these scenarios during the sampling process, and toss only enough samples to solve the fraction of the problem that can indeed be solved. It turns out that this is possible, as we will see in the next subsection.

4.2 A Dual Sampling Scheme

Sampling $\mathcal{W}$ uniformly at random has the serious drawback that most samples are wasted when the sensor range is lower- and upper-bounded. For instance, a sensor located at the center of a big room is unable to observe the walls if the sensor's range is smaller than the room's width. It is desirable to avoid sampling regions of the interior of $\mathcal{W}$ that cannot possibly cover the boundary. This can be achieved by sampling the constraints of the problem (the points in $\partial\mathcal{W}$ that must be to covered) before sampling the interior of $\mathcal{W}$.

First, a boundary point $w \in \partial\mathcal{W}$ is selected at random, and the region $\mathcal{V}(w) \subset \mathcal{W}$ from which such point can be observed is computed using the classic ray-sweep algorithm (in $O(n \log n)$). The region $\mathcal{V}(w)$ is sampled pseudo-uniformly by first triangulating the region, and then distributing m' sample points among all triangles according to their relative areas. These points are stored in the sample list $\mathcal{G}_{sam}$. Because $\mathcal{V}(w)$ is a visibility polygon, a triangulation computed from w is direct, and the cost of computing and sampling $\mathcal{V}(w)$ is $O(n \log n + m')$.

Second, a point $p \in \mathcal{V}(w)$ is selected at random and used as a proxy of $\mathcal{V}(w)$. From this position we compute the fraction of $\partial\mathcal{W}$ visible from p, and we subtract this from $\partial\mathcal{W}$ to compute the new unobserved boundary $\partial\mathcal{W}'$. A new point $w \in \partial\mathcal{W}'$ is randomly selected, and the entire process is repeated

The advantage of this sample scheme is that a user may prescribe a degree of coverage as the termination criterion: i.e., *stop* when the reduction of the unobserved perimeter is not changing significantly with respect to previous iterations. This is very useful, for in most cases few guards cover almost the entire boundary, but many more are required to cover narrow corners that compose a small fraction of the total. But more importantly, workspaces that cannot be fully guarded can be dealt in practice using this scheme.

5. NEAR-OPTIMAL COVERS FOR THE SAMPLED PROBLEM

This section shows that the sampled instance of Problem 2.1, represented by $\Sigma = (X, \mathcal{R})$, can be solved within a factor $O(\log(n + h) \cdot \log(c \log(n + h)))$ from the optimal size c. The approximation factor depends only on the complexity of $\mathcal{W}$, and not on the size of $\mathcal{G}_{sam}$. To achieve this, we will use the algorithm proposed by Brönnimann and Goodrich [4] for finding near-optimal set covers for set systems with finite VC-dimension.

The main result from [4] is that, for set systems with VC-dimension d, it is possible to compute in polynomial-time a hitting set of size $O(dc \log(dc))$, where c is the size of the smallest hitting set. To apply this result, we must first introduce hitting sets and transform our problem into an instance of the hitting-set problem by computing the *dual* of Σ. Afterwards, we define the VC-dimension of a set system, and show that the dual of Σ has VC-dimension $d = O(\log(n + h))$.

5.1 Hitting Sets and the Vapnik-Červonenkis Dimension

Let $\Sigma = (X, \mathcal{R})$ be a set system. The *dual* set system $\Sigma' = (X', \mathcal{R}')$ is defined by $X' = \mathcal{R}$ and $\mathcal{R}' = \{\mathcal{R}_x | x \in X\}$, where $\mathcal{R}_x$ consists of all the sets $R \in \mathcal{R}$ that contain x. The dual set system for the art-gallery problem is illustrated in Figure 7. Notice that the set of guard candidates now becomes the ground set X'.

A *hitting set* for $\Sigma' = (X', \mathcal{R}')$ is a set $H' \subseteq X'$ such that $H' \cap R' \neq \emptyset$ for every set R' in $\mathcal{R}'$ (i.e., H' contains members from all the sets in $\mathcal{R}'$). The problem of finding the optimal set cover for Σ is *equivalent* to that of finding the smallest

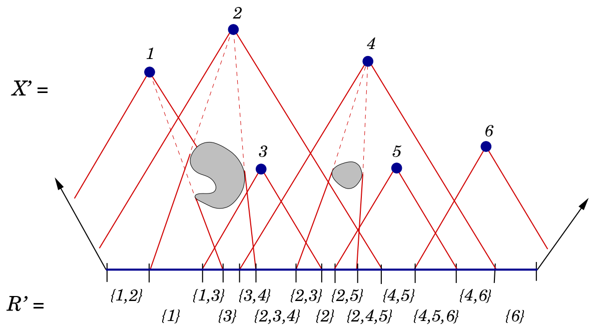


Figure 7: The art-gallery problem can be posed as a hitting-set problem. The guard candidates are labeled and grouped under X', and each set $R'_i \in \mathcal{R}'$ is the set of samples covering element i in the boundary decomposition. This formulation is the *dual* of the one shown in Figure 4. The problem of finding the best set cover is transformed into that of finding the best hitting set.

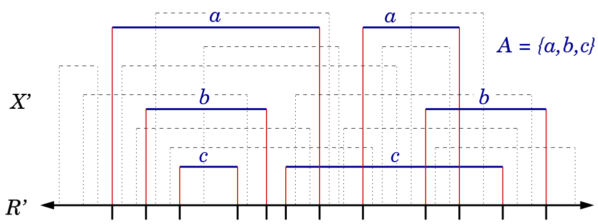


Figure 8: Each guard candidate projects a set of intervals onto the boundary of the environment; X' can be seen as an arrangement of intervals. A particular subset $A \subseteq X'$ creates its own sub-arrangement and subdivisions. For any two members of $\mathcal{R}'$ contained inside the same subdivision, $R'_i \cap A = R'_j \cap A$.

hitting set for Σ'. Let c be the size of the smallest hitting set (also the size of the smallest set cover).

The *VC-dimension* of Σ' is defined as follows:

Definition 5.1 (Vapnik-Červonenkis Dimension). *Let $\Sigma' = (X', \mathcal{R}')$ denote a set system. A set $A \subseteq X'$ is said to be* shattered *by $\mathcal{R}'$ if for any subset $B \subseteq A$ there exists some $R' \in \mathcal{R}'$ such that $B = A \cap R'$. The* VC-dimension *of Σ' is the cardinality of the largest shattered subset of X'.*

In other words, A is shattered if each of its subsets can be induced by intersecting A with some set in $\mathcal{R}'$. Although it may not be possible to shatter all sets of size d, as long as there exists one such set we say that the VC-dimension is at least d. To state that a set system has VC-dimension d we must prove that no set of size larger than d can be shattered.

The VC-dimension of our dual set system $\Sigma' = (X', \mathcal{R}')$ is upper bounded by the following theorem:

Theorem 5.1. (VC-dimension for Sampled Art-Galleries). *The VC-dimension of the dual of the set system representing the sampled instance of Problem 2.1 is bounded by $O(\log(n+h))$.*

Proof. We proceed as follows (see Figure 8): The portion of $\partial\mathcal{W}$ seen by each guard candidate consists of at most $k = 2(n+h)$ pieces. Thus, X' can be seen as an arrangement of k-intervals, where a k-interval is a set composed of at most k disjointed regular intervals. Select $A \subseteq X'$ with $d = |A|$. The set A defines a sub-arrangement of at most $2kd$ subdivisions in $\partial\mathcal{W}$, and any two members of $\mathcal{R}'$ within the same subdivision induce the same subset of A (i.e. $R'_i \cap A = R'_j \cap A$, if R'_i and R'_j are in the same subdivision). In order to shatter A each of its subsets must be induced, but no R'_i can induce more than one subset of A. Therefore, to induce all the subsets in A there must be at least 2^d subdivisions in the sub-arrangement. In other words, it is impossible to shatter A if:

$$2kd < 2^d \iff 2k < \frac{2^d}{d}. \tag{3}$$

For $d > 4$, $\log(2k) < d/2 \Longrightarrow 2k < 2^d/d$. This observation, along with the fact that $k = 2(n+h)$, implies that the VC-dimension is upper bounded by $2\log(4(n+h))$. □

5.2 Finding a Near-Optimal Hitting Set

The algorithm proposed by Brönnimann and Goodrich is based on finding an ϵ-net that approximates the optimal hitting set. Let $\Sigma = (X, \mathcal{R})$ denote a set system. A set $N \subseteq X$ is said to be an ϵ-net of Σ if it intersects each set $R \in \mathcal{R}$ of cardinality larger than $\epsilon|X|$. We can generalize this definition by including an additive weight function w on every subset of X. In the generalized case, an ϵ-net is required to hit every R with weight at least $\epsilon w(X)$.

Let a *net finder* of size s be an algorithm A that given a weight function w on X, returns a $(1/r)$-net of size $s(r)$ for a set system Σ. Also, a *verifier* is an algorithm B that given a subset $H \subseteq X$ either confirms that H is a hitting set, or returns a witness set $R \in \mathcal{R}$ such that $R \cap H = \emptyset$ when H is not a hitting set of X.

The main result from [4] is that given algorithms A and B, we can find a hitting set of size at most $s(4c)$ (where c is the optimal size) by executing the following procedure:

Procedure HITTING-SET$(X, \mathcal{R})$

1. Select $c' = 1$.
2. Given the net finder A and the verifier B, confirm if there is a hitting set H of size at most $s(2c')$:
   (a) Set the weights of all the elements in X equal to 1. Set $k = 1$.
   (b) Use A to find a $(1/2c')$-net of size $s(2c')$ (call this net N).
   (c) Using B, verify if N is a hitting set. If N is not a hitting set, then B returns a set $R \in \mathcal{R}$ such that $R \cap N = \emptyset$.
   (d) If N is a hitting set, then $H = N$ and Step 2 exits with TRUE. Else if $k > 4c' \log(|X|/c')$, then Step 2 exits with FALSE.
   (e) Else set $k = k + 1$, and double all the weights of the elements of R. Return to step (b).
3. If Step 2 returns TRUE (i.e., there is a hitting set H of size c'), then the procedure terminates and H is the near-optimal hitting set. Otherwise, set $c' = 2c'$ and repeat Step 2.

The termination condition in Step 2(d) is a remarkable result from [4]. Indeed, it can be proven that Step 2 always returns a hitting set within $4c' \log(|X|/c')$ iterations if one exists. Because Step 2 returns a hitting set of size $s(2c')$, and because of the doubling condition in Step 3, the hitting set H is of size at most $s(4c)$. The overall cost of this procedure is $O(c \log(|X|/c))(T_A(|X|, |\mathcal{R}|, c) + T_B(|X|, |\mathcal{R}|, c))$, where T_A and T_B are the running times of the net finder and verifier, respectively.

$\Sigma = (X, \mathcal{R})$ is said to have a *subsystem oracle* of degree D if given a subset $A \subseteq X$ it is possible in time $O(|A|^{D+1})$ to compute the subsystem $(A, \mathcal{R}_{|A})$ (where $\mathcal{R}_{|A}$ is the family of subsets $B \subseteq A$ that can be generated by intersecting A with the sets in $\mathcal{R}$). Likewise, the oracle is a *witness oracle* if for any $B \in \mathcal{R}_{|A}$, a witness $R \in \mathcal{R}$ satisfying that $B = R \cap A$ can be found in time $O(|X|)$.

If the subsystem oracle exists, it has been shown [3] that a $(1/r)$-net of size $O(dr \log(dr))$ can be computed in time $O(d)^{3D} r^D \log^D(rd)|X|$ for a set system Σ of VC-dimension d. Therefore, the running time for the net finder is linear in $|X|$. For the verifier, it is simply a matter of running the subsystem oracle on H. If $\emptyset$ is not in $\mathcal{R}_{|H}$ then H is a hitting set. Otherwise, run the witness oracle to find a witness R that verifies that $\emptyset$ is in $\mathcal{R}_{|H}$. Hence, the running time for the verifier is also linear on $|X|$.

Under the above conditions, the algorithm HITTING-SET returns a hitting set of size $O(dc \log(dc))$ in polynomial time. We only have to prove now that the dual set system of the sampled instance of the art-gallery problem admits a subsystem oracle of finite degree. Afterwards, it is straightforward to prove the main result of this section.

THEOREM 5.2. (SUBSYSTEM ORACLE FOR SAMPLED ART-GALLERIES). *Let Σ' be the dual of the set system representing the sampled instance of Problem 2.1. Σ' admits a subsystem oracle with running time $O((n+h)|A| \log |A|)$.*

The proof for this theorem is analogous to that of Theorem 5.1. X' can be seen as an arrangement of k-intervals, and any $A \subseteq X'$ creates a sub-arrangement of at most $2k|A|$ subdivisions (recall Figure 8). To compute $\mathcal{R}_{|A}$ we merely have to sort and sweep these subdivisions. This has cost $O((n+h)|A| \log |A|)$ because k is bounded by $O(n+h)$, and the sorting operation is done over $|A|$ *pre-sorted* lists of size $O(n+h)$.

THEOREM 5.3. (NEAR OPTIMAL COVERS FOR SAMPLED ART-GALLERIES). *For the sampled instance of Problem 2.1, a set of guards of size at most $O(c \log(n+h) \cdot \log(c \log(n+h)))$ can be found in polynomial time, where c is the optimal size.*

Proof. Finding the optimal cover for the sampled art-gallery problem is equivalent to that of computing the optimal hitting set for the dual problem. Theorem 5.1 states that the dual VC-dimension of the problem is $O(\log(n+h))$, while Theorem 5.2 states that our problem admits a subsystem oracle of finite degree. We satisfy all the conditions stated in [4], and is therefore possible to compute a set of guards of size at most $O(c \log(n+h) \cdot \log(c \log(n+h)))$. $\square$

Remarks. The quality of the approximation obtained here is not a function of m. This number only affects the runtime of the algorithm. Of course, as discussed in Section 4, the value of m also affects the probability that the near-optimal solution to the sampled problem is in fact a near-optimal solution to Problem 2.1.

6. CONCLUSION AND EXTENSIONS

The computation of efficient motions for high-level vision-oriented operations is a problem that has received little attention so far. This paper presents a novel approach to reduce the number of sensing operations during the construction of large models. Specifically, we incorporated sensor limitations into a practical sensor placement algorithm. This algorithm is based on a random-sampling strategy that transforms the art-gallery problem into an instance of the set cover problem. This paper also shows how to solve this set cover problem efficiently.

An interesting topic for future research is to study placement strategies that address image registration issues. Due to errors in robot localization, views captured at different locations must be aligned prior to the construction of a consistent model. Reliable alignment techniques are essential in this context, which by necessity require some degree of overlap between images [18]. This fact further constrains the placement of guards by demanding a minimum overlap between their views. Adding this new constraint to the randomized strategy presented here is straightforward. But the problem of computing the image sequence (i.e., the sensing *order*) that minimizes registration errors remains open.

An obvious extension to the sensor placement problem is to generate *routes* instead of positions for tasks involving mobile sensors. This is a simple statement that has profound consequences. If the cost of sensing is very expensive relative to the cost of motion — in time, resources, or computation — then motion costs can be neglected and the problem becomes identical to the one addressed in Section 2. Conversely, the cost of sensing can be neglected when sensing is cheap compared to the cost of motion. In this scenario, the mobile device is assumed to be capable of almost continuous sensing. The planning problem now becomes the *watchman route problem* [15]: Find the shortest closed path from which the entire workspace is visible. Extending the randomized techniques presented in this paper to compute watchman routes is an interesting topic for future research.

A more difficult problem is when neither the cost of sensing nor the cost of motion can be neglected. This is a more general problem, and the art-gallery and watchman-route problems become special cases. In practice, the problem is complicated by other considerations such as fuel and total distance traveled. This topic remains largely unexplored, but some preliminary work exists [8].

The algorithm presented in this paper can be embedded into a larger autonomous model-building system as follows. Once the set of guard locations is computed, a graph is constructed using these locations as nodes. A pair of nodes (g_1, g_2) in this graph is connected if two conditions hold true: (a) The line segment connecting g_1 with g_2 does not intersect $\partial\mathcal{W}$, and (b) there is a minimum overlap between the visible regions at g_1 and g_2. Afterwards, the graph can be processed using a *traveling-salesman* algorithm in order to compute the shortest route (under some metric) connecting all of these locations. The resulting system produces a tour that a robot may follow in order to build a large model of a building, in spite of image registration constraints. The computed tour will not be optimal, but from the engineering perspective it represents a practical solution. Generating optimal routes for mobile sensors under image-registration constraints is an open problem, and will probably remain so for some time.

7. ACKNOWLEDGEMENTS

We would like to thank Alon Efrat for pointing out the potential of using the concept of VC-dimension to analyze the performance of the randomized art-gallery algorithm; and Dan Halperin, T.M. Murali and David Hsu for their observations on how to improve the bound in Theorem 5.1. We also thank Leo Guibas for his comments and suggestions.

This work was funded by DARPA contract DAAE07-98-L027, ARO MURI grant DAAH04-96-1-007 and NSF grant IIS-9619625.

8. REFERENCES

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

[2] M. Bern. Triangulations. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 413–428. CRC Press, Boca Raton, FL, 1997.

[3] H. Brönnimann, B. Chazelle, and J. Matoušek. Product range spaces, sensitive sampling, and derandomization. In *Proc. of 34th IEEE Symposium on Foundations of Computer Science*, pages 400–409, 1993.

[4] H. Brönnimann and M.T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete and Computational Geometry*, 14:463–479, 1995.

[5] C. I. Conolly. The determination of next best views. In *IEEE Int. Conf. on Robotics and Automation*, pages 432–435, 1985.

[6] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 1990.

[7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. ACM SIGGRAPH*, pages 303–312, August 1996.

[8] T. Danner and L. Kavraki. Randomized planning for short inspection paths. In *Proc. 2000 IEEE Int'l Conf. Robotics & and Automation*, pages 971–976, April 2000.

[9] H. González-Banos, A. Efrat, J.C. Latombe, E. Mao, and T.M. Murali. Planning robot motion strategies for efficient model construction. In J. Hollerbach and D. Koditschek, editors, *Robotics Research - The Ninth Int. Symp.*, Salt Lake City, UT, 1999. Springer-Verlag.

[10] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. ACM SIGGRAPH*, pages 31–42, 1996.

[11] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(5):417–433, May 1993.

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

[13] J. O'Rourke. Visibility. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 467–479. CRC Press, Boca Raton, FL, 1997.

[14] 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.

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

[16] Petr Slavik. A tight analysis of the greedy algorithm for set cover. *Journal of Algorithms*, 25(2):237–254, November 1997.

[17] J. Snoeyink. Point location. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 559–574. CRC Press, Boca Raton, FL, 1997.

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

[19] Jorge Urrutia. Art gallery and illumination problems. In J. R. Sack and J. Urrutia, editors, *Hanbook on Computational Geometry*, pages 387–434. Elsevier Science Publishers, 1997.

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