Probing View Plan Solution Space

Scott, William

https://doi.org/10.4224/8914055

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Probing View Plan Solution Space *

Scott, W.R.
March 2005

* published as NRC/ERB-1124. March 31, 2005. 8 pages. NRC 47447.

Canada

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

ERB-1124

NRC·CNRC

Probing View Plan Solution Space

Scott, W.R.
March 2005

Canada

NRC 47447

Probing View Plan Solution Space

William R. Scott†

† Computational Video Group,
National Research Council of Canada, Ottawa, Canada, K1A 0R6
william.scott@nrc-cnrc.gc.ca

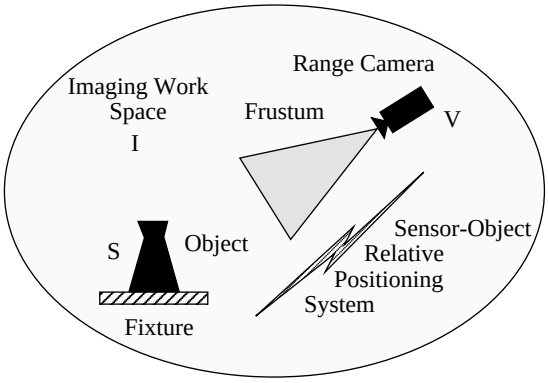

Figure 1: Imaging Environment

Abstract

The view planning problem (VPP) arises in a number of important computer vision applications such as 3D object reconstruction and inspection. The VPP is known to be isomorphic to a classical problem in combinatorial mathematics, the set covering problem (SCP). This paper characterizes the nature of the VPP and presents a set covering method exploiting knowledge of the topology of viewpoint space.

1 Introduction

The goal of object reconstruction with an optical sensor is to assembly a virtual object model, for example a 3D mesh, capturing surface geometry and possibly appearance (reflectance properties). In the inspection case, we start with a high definition virtual object model, such as a CAD representation, and wish to compare the reference model with the actual shape of an instance of the physical object, such as a production line sample. Both cases begin with specified quality objectives for measurement precision and sampling density. The imaging environment (Figure 1) consists of a range camera, positioning system, various fixtures and the target object [12].

These computer vision tasks involve planning views, physically altering the relative object-sensor pose, optimizing sensor parameters, taking scans, registering the acquired geometric data in a common co-ordinate frame of reference and finally integrating range images into a non-redundant model. The view planning component of this process remains an open problem.

Stated informally, the view planning problem (VPP) involves finding a suitably small set of sensor poses and configurations for a specified reconstruction or inspection task. The VPP involves reasoning about object surface space S, viewpoint space V and imaging workspace I. The view plan N of size $n = |N|$ is merely an optimal subset of viewpoint space.

A measurability matrix [9] $\mathbf{M} = [m_{ij}]$ is a convenient and powerful data structure capturing the essence of the problem. Rows of $\mathbf{M}$ span discretized surface space S while columns span discretized viewpoint space V. $\mathbf{M}$ has sv elements, where $s = |S|$ is the number of rows (surface points) and $v = |V|$ is the number of columns (viewpoints). In the inspection case, the set S is derived from sub-sampling the high resolution reference CAD model, while for reconstruction it is developed from an initial coarsely-sampled scene exploration stage. The set V is derived from discretizing viewpoint space in some optimal manner. Each measurability matrix element m_{ij} is a binary estimate of measurability of a single surface point $\mathbf{s}_i$ on an approximate object model from a single viewpoint $\mathbf{v}_j$. Measurability determination involves tests for frustum occupancy, visibility and specification compliance [9].

Thus, the solution to the VPP is simply to cover the rows of $\mathbf{M}$ with a minimal subset of its columns. Consequently, it is immediately apparent that the VPP is isomorphic to the set covering problem (SCP), a well-studied problem in combinatorial optimization. The complexity of deriving $\mathbf{M}$ is $\mathcal{O}(s^2v)$ [9]. The complexity of the SCP is known to be NP-Complete [7].

It is evident then, that solving the VPP involves two computational challenges: (1) efficient algorithms for acquiring a rough scene model (in the reconstruction case), discretizing S and V and computing the measurability matrix $\mathbf{M}$ [9] and (2) solving the set covering problem. This paper presents a solution method for the VPP version of the SCP by exploiting knowledge of the topology of viewpoint space.

The paper is organized as follows. In Section 2, we

characterize the VPP, in particular the phenomenon of viewpoint correlation. Section 3 characterizes the SCP and briefly reviews conventional set covering methods. Section 4 describes the probe method, while Section 5 presents experimental results. Finally, Section 6 summarizes the results and concludes the paper.

2 View Planning Problem

2.1 Topology of Viewpoint Space

While the VPP and SCP are isomorphic in terms of computational complexity, there are subtle differences. One of these concerns the covering sets. In many SCP applications, the covering sets have no known relationship to one another. With a generate and test formulation of the VPP, however, V is discretized by a known algorithm and therefore the topology of the covering sets (viewpoints) is known a priori.

The most common strategy for discretizing V is the object-centered "view sphere" approach, of which there are several variants [12]. The "optimal scanning zone algorithm" and its modifications [9] better matches candidate viewpoints with sensor capabilities and object geometry. In its simplest form, the algorithm creates one optimally-configured viewpoint per vertex on the approximate object model, creating a 1:1 mapping between points in S and V.

All such viewpoint generation algorithms overlay a complex, multi-dimensional web of neighbour relationships on the 1D layout of columns (viewpoints) in the measurability matrix $\mathbf{M}$. Effectively, this viewpoint mesh is an aspect graph.

We represent the object surface as a 3D triangular mesh based on simplification of Woo's symmetric boundary data schema [17]. Our mesh implementation explicitly embeds 4 of 9 possible topological relationships: $E \rightarrow V$, $E \rightarrow T$, $T \rightarrow E$ and $V \rightarrow T$. In this context, E,V and T refer to mesh edges, vertices and triangles and the notation $X \rightarrow Y$ means "given X, find Y". The modest storage requirements for the embedded connectivity allows us to rapidly and efficiently compute other topological relationships. Given the previously noted mapping between S and V, we can also rapidly compute the viewpoint neighbour function $\mathbf{v}_j \rightarrow N(\mathbf{v}_j)$. Such a viewpoint neighbour function can be defined for all deterministic schemes for discretizing viewpoint space.

2.2 Viewpoint Correlation

In our usage, a generalized viewpoint [14] $(\mathbf{v}, \lambda_\mathbf{s})$ defines a discrete sensor pose $\mathbf{v}$ with a defined frustum in 3D space plus a given setting of configurable sensor parameters $\lambda_\mathbf{s}$. There is a direct correspondence between viewpoints and range images and we will use the terms interchangeably. Topologically-adjacent viewpoints will be correlated and have overlapping images.

A degree of image overlap is both inevitable and desired. It is inevitable due to (1) the irregular shape of the intersection of the sensor frustum cross-section with complex object surface geometry and (2) sensing physics which are sensitive to standoff range, grazing angle and surface reflectance properties. The VPP also requires a degree of image overlap for range image registration and integration [12], the former a consequence of system pose error and the later a necessity to seamlessly stitch adjacent range images together.

Viewpoint correlation also influences the scheme for discretization of viewpoint space. There is a push-pull dynamic involved. We desire candidate viewpoints sufficiently nearby for registration and integration requirements and to ensure a high probability of imaging difficult-to-image surface regions, yet far enough apart for efficient sampling of viewpoint space and acceptable computational burden in constructing $\mathbf{M}$.

Image overlap, and consequently image registration, can be determined from the degree of viewpoint correlation. For the purposes of view planning, we define the cross-correlation σ_{kj} of two viewpoints v_k and v_j as the dot product of the respective column vectors $\mathbf{M}_{S,k}$ and $\mathbf{M}_{S,j}$ of the measurability matrix, normalized by the maximum viewpoint coverage of any viewpoint in the candidate viewpoint set - that is, $m_S = max|\mathbf{M}_{S,k}| \ \forall k \in V$, so

$$\sigma_{kj} = \frac{\mathbf{M}_{S,k} \cdot \mathbf{M}_{S,j}}{m_S}. \qquad (1)$$

Graphical depictions of viewpoint correlation for simple shapes are shown at [10, 11]. Whereas viewpoint correlation is easily computed, visualization of the function for complex surface geometry is best done with a 3D viewing tool and is not readily displayed with the limitations of a 2D paper medium. Correlation data shows that neighbouring viewpoints are highly correlated and that correlation falls off with displacement (not strictly monotonically) and is modified by surface shape. Viewpoints which are geodesically well separated are uncorrelated.

3 Set Covering Problem

3.1 IP Formulation

We can formally express the VPP as the following integer programming (IP) problem, where we use IP

notation conventions[1]. This simplified formulation is equivalent to the unicost SCP.

$$\text{Minimize } Z = \sum_{j=1}^{v} x_j \text{ , subject to} \tag{2}$$

$$\sum_{j=1}^{v} m_{ij} x_j \geq 1 \,;\; i = 1, \ldots, s \;\; , \tag{3}$$

$$x_j \in \{0, 1\} \,;\; j = 1, \ldots, v \;. \tag{4}$$

Equation 2 is the objective function. Equation 3 ensures each row of $\mathbf{M}$ (each surface point) is covered by at least one viewpoint. Equation 4 applies an integer constraint on the viewpoint variable x_j. $\mathbf{X} = [x_j]$ spans viewpoint space V as sampled by the viewpoint generation stage. The optimal view plan $\mathbf{X}$ is the minimal set of viewpoints covering surface space S.

Two measures of a solution's merit are cost (which is the value of the objective function in Equation 2 and in our case equals view plan length) and measurability. The measurability of a view plan N is the ratio of the joint coverage of its viewpoints to the size of discretized surface space S [9]. That is,

$$m(N) = \frac{|\bigcup_{i \in N} \mathbf{M}_{S,i}|}{s}. \tag{5}$$

3.2 VPP Solution Space

The size of view plan solution space is 2^v, where v is the number of columns (viewpoints) in $\mathbf{M}$. Similarly, the number of view plans of length n is ${}_vC_n$, which we can approximate by Sterling's Formula

$${}_vC_n \approx \frac{1}{\sqrt{2\pi}} \frac{v^{v+\frac{1}{2}}}{(v-n)^{v-n+\frac{1}{2}} \; n^{n+\frac{1}{2}}}. \tag{6}$$

Some quantitative examples will put these numbers in context. Consider a typical case where $v = 400$. Then, the size of view plan solution space is

$$2^v = 2^{400} \approx 10^{120}. \tag{7}$$

This is a rather large number, even for the modest number of viewpoints. Now, consider a slice through VPP solution space limited to view plans of length $n = 10$. From Equation 6, the number is approximately

$${}_{400}C_{10} \approx 10^{19}. \tag{8}$$

The number of potential solutions is still huge. Given a processor capable for examining a potential solution every nanosecond, it would take approximately 825 years to exhaustively search just this portion of VPP solution space! Fortunately, it is a characteristic of the VPP that, while the search space may be large, solutions are typically small.

For any given view plan length n, VPP solution space has a large number of local maxima. We can visualize this as an n-dimensional sea urchin with spines in the range [0,1]. "Feasible" solutions have $m(N) = 1.0$, whereas the measurability of "infeasible" solutions falls short of that objective. Above some minimum view plan length, there will exist multiple feasible solutions of any given view plan length. The density of feasible solutions decreases as view plan length approaches the optimum.

From the binomial theorem, we know VPP solution space has the topology of a hypercube. Unfortunately, we do not know the topology of feasible solutions. That is, there is no known algorithm permitting us to move directly from one feasible solution to its nearest feasible neighbour.

3.3 SCP Solution Methods

Expressing the VPP as an integer programming problem provides a compact mathematical formulation of the problem, opening up the rich research base in discrete combinatorial optimization, in particular techniques developed for the SCP. We have seen from the previous section that the solution space is huge, even for modest problems and we know that the search for an optimal solution is NP-Complete. While guaranteeing optimal results, exact methods such as branch-and-bound and cutting-plane techniques can be computationally prohibitive for even modestly sized IPs. For most medium-to-large IPs, this leaves a choice of approximate and heuristic algorithms [8], including greedy search [6], simulated annealing [13], genetic algorithms [2], Lagrangian relaxation [1, 5] and neural network [7] methods. Recent surveys of SCP solution techniques [3, 4] have shown that substantial progress has been made with Lagrangian techniques. Most published performance results [1, 7] deal with random, low density data sets. The VPP falls into the category of a medium-to-large IP with non-random data and moderate density [10].

We have found that greedy search works well for many object reconstruction problems. It is fast, essentially instantaneous, and the modest degree of view

[1] Here, we have used unitary viewpoint movement costs and removed the registration constraint as it is frequently rendered moot due to inherent view plan redundancy for complex objects. For a complete IP formulation, refer to [9].

```
INITIALIZE
n_u = n_gs
n_l = ⌈A / F_s⌉
N_Best = ∅
count = 1

PROBE VPP SOLUTION SPACE
while (1)
{
  N = createRandomViewPlan(size n_u)
  N = optimize(N)

  count++

  TERMINATION CRITERIA
  If (m(N) > m(N_Best))
    N_Best = N
  If (m(N_Best) = 1.0)
    n_u = n_u − 1
  If ((n_u = n_l)||(count > limit))
    break
}
```

Table 1: Pseudo Code - Probe VPP Solution Space

```
INITIALIZE
m_b = m(N)

while (improvement)
{
  CYCLE THROUGH THE VIEW PLAN
  For ( each v_j, j ∈ N )
  {
    GET THE NEIGHBOURING VIEWPOINTS
    v_j → N(v_j)

    CYCLE THROUGH THE NEIGHBOURS
    For ( each v_i, i ∈ N(v_j) )
    {
      m_t = m(N − v_j + v_i)

      REPLACE IF AN IMPROVEMENT
      If (m_t > m_b)
      {
        m_b = m_t
        N = N − v_j + v_i
      }
    }
  }
}
```

Table 2: Pseudo Code - Optimize View Plan

plan inefficiency improves robustness to system errors, in particular pose error. For applications requiring greater view plan efficiency, such as inspection and some reconstruction problems, the developer has a range of solution techniques available, including those noted above and the probe method to be described in the next section.

4 Probe Method

4.1 Probe Algorithm

Bounds on view plan length can be quickly computed, limiting the region of VPP solution space to be searched. For the unicost SCP, greedy search provides a reasonably tight upper bound n_u. The ratio $\lceil A/F_s \rceil$ of rough model surface area to the frustum cross-sectional area at the sensor's optimal scanning range provides a valid, if rather loose, lower bound n_l.

The probe algorithm (Table 1) uses a step-down search, starting from the computed upper bound and terminating after a user-specified number of probes. At each step, VPP solution space is randomly probed at the given view plan length. Most probes initially find infeasible solutions i.e. a view plan N such that $m(N) < 1.0$. The result of each probe is optimized (i.e. view plan measurability is maximized) by a local hill climbing operation (Table 2) using knowledge of the topology of viewpoint space V. Hill climbing quickly finds the nearest local optimum but not necessarily the global optimum. When a feasible solution is found, target view plan size is decremented. The best feasible solution is retained.

We have examined two variants of the probe algorithm: uniform random probe and viewability-weighted random probe. Uniform probing selects new candidate view plans of a given length by sampling viewpoints with equal probability. Viewability-weighted random probing favours viewpoints imaging difficult-to-view surface regions. The rationale is to assign higher weights to those viewpoints covering the largest number of sparsely measured surface points. Both benefit from local view plan optimization. Presently, neither exploits the cumulative knowledge gained from prior probing.

Figure 2: Tsimshian Stone Mask

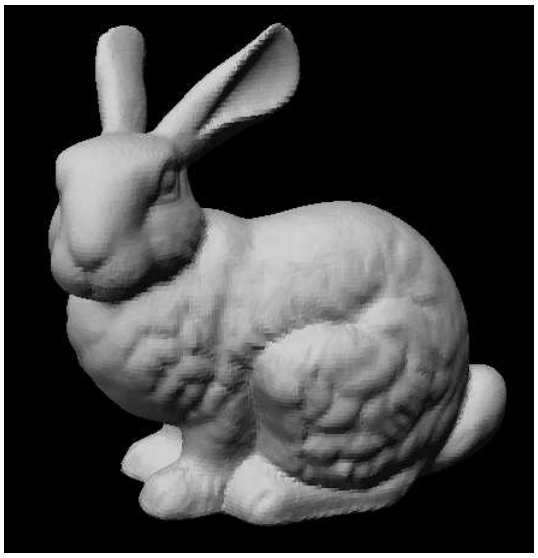

Figure 3: Bunny

4.2 Viewability-Weighted Random Probing

The measurability $m(\mathbf{v}_j)$ of a viewpoint $\mathbf{v}_j$ is the relative portion of the surface measured by that viewpoint [9]. $m(\mathbf{v}_j)$ is defined as

$$m(\mathbf{v}_j) = \frac{1}{s}\sum_{i=1}^{s} m_{ij}. \tag{9}$$

The viewability $v(\mathbf{s}_i)$ of a surface point $\mathbf{s}_i$ is the relative portion of viewpoint space which measures that surface point. $v(\mathbf{s}_i)$ is defined as

$$v(\mathbf{s}_i) = \frac{1}{v}\sum_{j=1}^{v} m_{ij}. \tag{10}$$

It seems reasonable to focus view planning on the difficult-to-measure surface regions i.e. those surface points with low viewability. Thus, we define weighted measurability $m_w(\mathbf{v}_j)$ as

$$\begin{aligned} m_w(\mathbf{v}_j) &= \frac{1}{s}\sum_{i=1}^{s}\left[m_{ij}\frac{1}{v(\mathbf{s}_i)}\right] \\ &= \frac{v}{s}\sum_{i=1}^{s}\left[\frac{m_{ij}}{\sum_{l=1}^{v} m_{il}}\right] \end{aligned} \tag{11}$$

The range of both $m(\mathbf{v}_j)$ and $v(\mathbf{s}_i)$ is $[0,1]$. The range of weighted measurability[2] $m_w(\mathbf{v}_j)$ is $[1,v]$.

4.3 The Literature

In writing this paper, we discovered an aspect of Tarbox's [15] work which is somewhat similar to our probe technique. Tarbox used simulated annealing (SA) in his algorithm "C", exploiting the geometry of the viewpoint sphere to access the neighbours of "sensing operations". Without detracting from the overall importance of Tarbox's pioneering view planning research, there are a number of weaknesses with his approach to view plan optimization and solving the SCP. His "linear" SA search followed a step-up operation which had the disadvantage of not starting from a firm lower limit[3]. There appears to be no rational basis for determining how much computational resource should be spent at the lower limit before incrementing the size of trial view plans and continuing the search, and so on until a feasible solution was found. In contrast, our step-down approach operates from a firm upper limit that is close to the optimal. We only decrement view plan length once a compliant solution is found, which becomes the new firm upper bound. His "binary search" approach also appears suspect because the method requires definitive tests of movable upper and lower bounds, neither of which existed. Finally, examination of the simulated annealing implementation [16] indicates that very small perturbations were

[2]Note that for any i for which the denominator $v(\mathbf{s}_i) = 0$ in the expression for $m_w(\mathbf{v}_j)$, numerator m_{ij} is also 0.

[3]However, if one could determine a firm lower limit there would be no need to proceed further except to find a specific instance of a solution at that lower limit.

employed in which a randomly-selected viewpoint in the trial view plan would be perturbed to its neighbour in position or boresight twist angle. Given the small perturbation step and the exponential probability selection function, it is unlikely that even a series of higher energy perturbations would be sufficient to break out of the current energy well. In other words, the annealing portion of the algorithm appears weak, such that the approach becomes primarily a computationally expensive method of local hill climbing.

5 Experimental Results

5.1 Viewability

For experimental subjects, we again use the Tsimshian stone mask (Figure 2) and bunny (Figure 3) objects previously reported at [9]. Both objects pose difficult view planning challenges for triangulation-based range sensors due to shadowing and self-occlusion problems. The mask object was segmented into front and back segments for view planning purposes. In this paper, we consider only the more difficult rear segment. For the bunny object, view plans were computed for the object as a whole.

Viewability distributions for the mask and bunny objects are shown at Figures 4 and 5. As the mask object is segmented, we observe a wider dispersion in viewability. The distribution is also bimodal. Regions near the bottom of the main cavity are observed by a high percentage of viewpoints. There is also a band around the steep cavity walls observable by only a small percentage of viewpoints. As an unsegmented object that is somewhat spherical in shape, viewability of the bunny is concentrated in the lower range with some dispersion due to shape complexity. Again, a small number of surface points are measurable by only a small portion of viewpoint space.

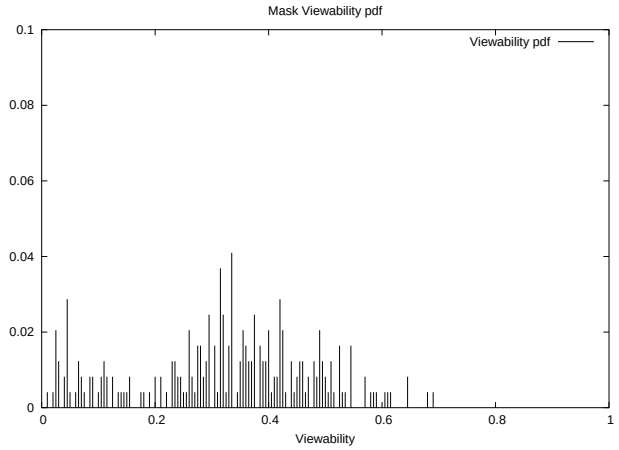

Figure 4: Mask Rear Segment Viewability pdf

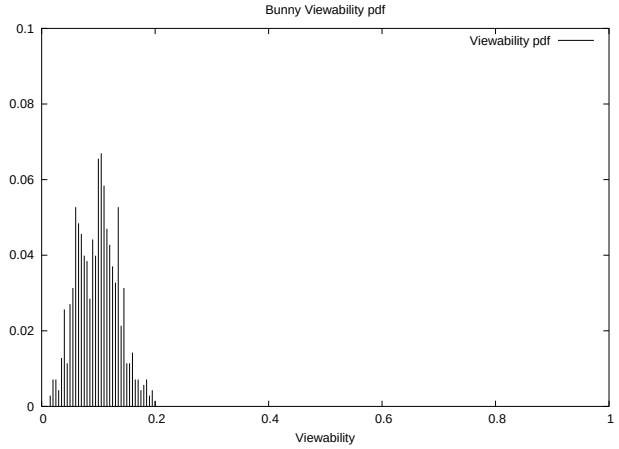

Figure 5: Bunny Viewability pdf

5.2 Probe Initialization

Experiments were conducted with the bunny object to compare viewability-weighted versus uniform-weighted random probing. The measurability matrix corresponds to one instance in a series of experiments involving surface sampling error, pose error and pose error compensation. The matrix's size was $s = 702$ rows by $v = 726$ columns, with density $\rho_M = 0.09579$. Table 3 summarizes the results of uniform- and viewability-weighted probing for view plans of size $n = 16, 12, 8$. One thousand trials were conducted at each view plan length. Columns 3 and 4 present statistics for the average measurability $\overline{m_{int}}$

n	Weighting	$\overline{m_{int}}$	σm_{int}	$\overline{m_{opt}}$	σm_{opt}
16	Uniform	.7658	.06449	.9873	.00538
	Viewability	.8090	.05384	.9881	.00463
12	Uniform	.6720	.06896	.9496	.01461
	Viewability	.7175	.06206	.9525	.01106
8	Uniform	.5333	.07428	.8243	.02308
	Viewability	.5805	.06414	.8271	.01987

Table 3: Random View Plan Initialization

n	$\Delta\overline{m_{int}}$	$\Delta\sigma_{m_{int}}$	$\Delta\overline{m_{opt}}$	$\Delta\sigma_{m_{opt}}$
16	+5.64%	-16.5%	+0.08%	-13.9%
12	+6.77%	-10.0%	+0.31%	-24.3%
8	+8.85%	-13.7%	+0.34%	-13.9%

Table 4: Viewability- vs Uniform-Weighted Initialization

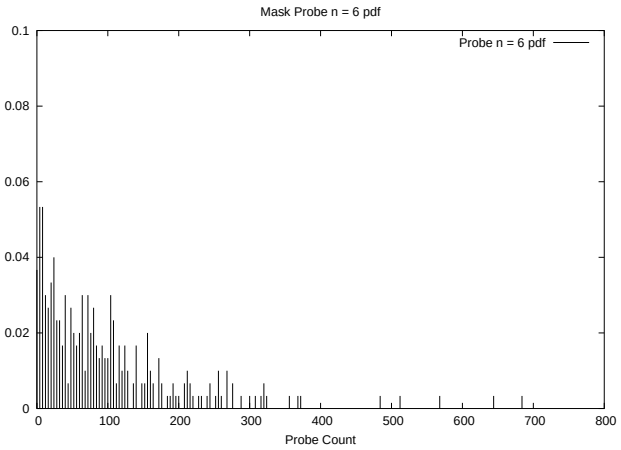

Figure 6: Mask Probe n = 6 pdf

and standard deviation measurability $\sigma_{m_{int}}$ of the initial, unoptimized probes. Columns 5 and 6 show statistics for the average measurability $\overline{m_{opt}}$ and standard deviation measurability $\sigma_{m_{opt}}$ of probes after local optimization.

Initialization methods are compared at Table 4. The results show that viewability-weighted probe initialization provides consistent improvement in both the expected value and variance of initial measurability across a range of view plan solution space. After probe optimization, the distribution of optimized measurability shows a slight improvement in average values with a marked decrease in standard deviation.

5.3 Set Covering Performance

Next, view plans for the mask and bunny objects were optimized using the probe method with uniform-weighted initialization.

The measurability matrix for the mask rear segment had dimensions $(s, v) = (244, 248)$, with density $\rho_M = 0.32043$. It corresponds to one instance in a series of experiments involving pose error and pose error compensation. The upper bound on view plan length was determined by $n_{gs} = 8$. The best solution found

n	$\widetilde{p}$	$\overline{p}$	σp
7	17	24.0	23.8
6	72	98.0	103.1
5	2115.5	2935.5	2607.4

Table 5: Probe Set Covering - Mask

n	$\widetilde{p}$	$\overline{p}$	σp
18	42	51.0	44.1
17	177	252.9	233.9
16	1822	3198.2	3502.9

Table 6: Probe Set Covering - Bunny

in these and other experiments had length $n_{Best} = 5$. Three hundred trials were conducted at each stage to determine the statistics of probing to find a feasible solution of the specified length. Table 5 presents the median $\widetilde{p}$, average $\overline{p}$ and standard deviation σ_p statistics for the number of probes required to find a feasible solution at each view plan length. Figure 6 shows the distribution of the number of probes required at length $n = 6$. The distributions typically have an irregular exponential form.

Similar view plan optimization experiments were conducted with the bunny object previously described. In this case, 101 trials were conducted at each stage to determine the statistics of probing to find a feasible solution of the specified length. The upper bound on view plan length was determined by $n_{gs} = 19$. Table 6 presents probe statistics for this object. The best solution found in these and other experiments had length $n_{Best} = 16$.

6 Conclusion

The view planning problem(VPP) is isomorphic to the set covering problem (SCP), a well-studied problem in combinatorial optimization known to be NP-Complete. Mathematical techniques, notably integer programming methods, can provide provably optimal solutions for small problems. However, the solution space for real VPP/SCP applications of quite modest dimensions is impossibly huge for current exact mathematical techniques. Therefore, approximation methods must be employed.

Two classical methods for solving the SCP, simulated annealing (SA) and genetic algorithms(GA) involve randomly probing solution space in a manner analogous to processes observed in the natural world.

Both techniques produce progressively more optimal solutions over time, yet both are known to be slow.

In many SCP applications, little is known about the data or the covering sets and both are taken to be random. However, in the view planning case, the data is not purely random. Neighbouring covering sets (viewpoints) exhibit strong correlation. By suitable sampling, the topology of viewpoint space is known. Consequently, view plans can be locally optimized by a hill climbing operation, which is the basis for the probe method reported in this paper. It could also be used in conjunction with conventional simulated annealing or genetic algorithms methods to provide good initial seed solutions and to locally optimize new candidate solutions derived by the SA or GA algorithms.

The probe method is applicable to any view planning scheme where the topology of viewpoint space V is known and where one can define an efficient viewpoint neighbour function $\mathbf{v}_j \rightarrow N(\mathbf{v}_j)$. It relies on the ability to bound solution space by setting an upper bound by greedy search and follows a step-down search strategy, knowing multiple feasible solutions of a given length exist above some lower limit. An important limitation to the method is that it is applicable only to the unicost SCP. Also, while optimization is initially fast, progress rapidly slows down as the search approaches the optimal, a characteristic of all probing methods, including SA and GA algorithms.

Experiments have shown that viewability-weighted view plan initialization has benefit in random sampling of view plan solution space if local optimization is not performed. This could apply to traditional GA and SA algorithms. For the probe set covering algorithm, it is debatable whether this improvement merits the extra preprocessing step. Regardless of initialization weighting, the greatest improvement comes from local view plan optimization.

This paper has documented a set covering algorithm for the view planning problem exploiting the known topology of viewpoint space. We have productively used the method for some time. While we have not yet implemented competing techniques for direct comparison, we believe the probe technique is competitive for the view planning application with conventional simulated annealing and genetic algorithms. However, it is likely not as computationally efficient as some other mathematically-based advances in the SCP literature. Recent advances in Lagrangian relaxation methods [3, 4] appear particularly promising.

References

[1] J. Beasley. A lagrangian heuristic for set covering problems. *Naval Research Logistics*, 37:151–164, 1990.

[2] J. Beasley and P. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94:392–404, 1996.

[3] A. Caprara, M. Fischetti, and P. Toth. Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371, 2000.

[4] S. Ceria, P. Nobili, and A. Sassano. Set covering problem. In M. Dell'Amico et al, editor, *Annotated Bibliographies in Combinatorial Optimization*, pages 415–428. John Wiley and Sons, 1997.

[5] S. Ceria, P. Nobili, and A. Sassano. A lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programming*, 81:215–228, 1998.

[6] M. Fisher and L. Wolsey. On the greedy heuristic for covering and packing problems. *SIAM Journal of Algebraic and Discrete Methods*, 3(4):584–591, Dec 1982.

[7] T. Grossman and A. Wool. Computational experience with approximation algorithms for the set covering problem. *European Journal of Operational Research*, 101:81–92, 1997.

[8] C. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications, Oxford, 1993.

[9] W. Scott. Model-based view planning. Technical Report NRC-47448, National Research Council of Canada, Institute for Information Technology, April 2005.

[10] W. Scott, G. Roth, and J.-F. Rivest. View planning as a set covering problem. Technical Report NRC-44892, National Research Council of Canada, Institute for Information Technology, August 2001.

[11] W. Scott, G. Roth, and J.-F. Rivest. View planning with a registration constraint. In *3rd Int. Conf. on 3D Digital Imaging and Modeling, Quebec City*, pages 127–134, May 2001.

[12] W. Scott, G. Roth, and J.-F. Rivest. View planning for automated 3D object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96, March 2003.

[13] S. Sen. Minimal cost set covering using probabilistic methods. In *ACM Sym. Applied Computing, Indianapolis*, pages 157–164, 1993.

[14] K. Tarabanis, R. Tsai, and P. Allen. The MVP sensor planning system for robotic vision tasks. *IEEE Trans. Robotics and Automation*, 11(1):72–85, February 1995.

[15] G. Tarbox and S. Gottschlich. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61(1):84–111, January 1995.

[16] G. H. Tarbox. *A Volumetric Approach to Automated 3D Inspection.* PhD thesis, Rensselaer Polytechnic Institute, 1993.

[17] T.C.Woo. A combinatorial analysis of boundary data structure schemata. *IEEE Computer Graphics and Applications*, pages 19–27, March 1985.