Viewpoint Selection for Complete Surface Coverage of Three Dimensional Objects

D.R. Roberts and A.D. Marshall
Department of Computer Science,
Cardiff University
PO Box 916, Cardiff. CF2 3XF, U.K.
`dave@cs.cf.ac.uk`

Abstract

Many machine vision tasks, *e.g.* object recognition and object inspection, cannot be performed robustly from a single image. For certain tasks (*e.g.* 3D object recognition and automated inspection) the availability of multiple views of an object is a requirement.

This paper presents a novel approach to selecting a minimised number of views that allow each object face to be adequately viewed according to specified constraints on viewpoints and other features. The planner is generic and can be employed for a wide range of multiple view acquisition systems, ranging from camera systems mounted on the end of a robot arm, *i.e.* an eye-in-hand camera setup, to a turntable and fixed stereo cameras to allow different views of an object to be obtained. The results (both simulated and real) given focus on planning with a fixed camera and turntable.

1 Introduction

Considerable research in machine vision has been directed at object recognition and object inspection. For complete 3D recognition or inspection several distinct views of the object are usually required. However, until recently, little consideration has been applied towards identifying strategies for selecting such viewpoints. Usually fairly simplistic methods or assumptions (*e.g.* fixed angular steps) are made when selecting viewpoints. If careful *a priori* or on-line determination of viewpoints can be made then benefits may include improving the quality, efficiency and/or reliability of subsequent processing and object recognition tasks. In many instances object inspection and, often, object recognition will be performed in controlled environments where a CAD model of the object of interest may be available. It is worthwhile looking to exploit stored knowledge of the object and its environment to determine viewpoints that improve the performance of these tasks, *e.g.* reducing the amount of redundancy in data capture and the time cost of capturing this data.

This paper describes a planning system that has been developed to select viewpoints suitable for a variety of machine vision tasks, *e.g.* recognition and inspection. As input, the planner takes a description of the vision work-cell, a CAD boundary representation model of the object of interest, and a description of the visibility of each of the objects

faces. The work-cell description defines lens optical settings and other metrics that depend on the multiple view acquisition system used, *e.g.* if multiple views are obtained from a fixed camera and turntable then these metrics would include camera position and orientation relative to the turntable and turntable step size. For a given object face, its visibility is defined by a generalised cone that bounds the directions along which there is an unobstructed view of the face. Additional constraints can also be specified for the purpose of adapting the planner to search for viewpoints suitable for specific applications.

Planning is performed in two stages: (1) features are grouped into sets that are suitable for viewing from a common viewpoint (*viewpoint planning*), and (2) viewpoints are selected to view these feature groups (*viewpoint selection*).

2   Background/Related work

Two approaches to sensor and viewpoint planning are possible, these can be described as follows: (*a*) Determine the next viewpoint on the basis of information derived from the analysis of the current and previous views, or (*b*) Determine a suitable set of viewpoints prior to beginning the vision task.

The first approach is more applicable when there is no model of the viewed object available. Planning of this type has been used for the automatic generation of object models, acquiring images that can be used to represent a scene or an object, and for view selection for the purpose of visually searching a scene[1, 2, 3].

The second approach is suitable when a CAD model of the object is available. Possessing an understanding of the shape of the model allows more complex and specific plans to be generated that exploit the available knowledge. This form of sensor planning has been developed for several different purposes, for example object recognition [4, 5], general robot vision tasks [6], inspection of loose tolerance objects [7], and accurate inspection [8]. Both Cowan [4] and Tarabanis [6] have developed sensor planning systems that determine suitable sensor positions for viewing a feature by applying constraints to the set of possible viewpoints. Tarbox [7] has looked at viewpoint planning to allow the complete surface of an object to be seen.

3   Our goals: Minimal views versus best views

In general, the problem of selecting a set of viewpoints to allow the whole of an objects surface becomes a tradeoff between two rather contradictory aims — trying to minimise the number of viewpoints while trying to ensure that the viewpoints selected yield views of each face as close to the best view as possible. This is particularly important for an application where accuracy is a concern, *e.g.* visual inspection. The best view of a feature or a group of features can be defined in several ways. In general, the best view of an object face is obtained when it is viewed head on [9]. This occurs when the viewing direction is the inverse surface normal of the face. For a curved surface this can either be approximated by the average surface normal or the surface can be approximated by planar facets that are then considered separately. However, there is no guarantee that a face may be visible from this direction corresponding to the inverse surface normal. A more robust definition of the best view is the direction that lies within the faces visibility region and has the smallest angular offset from its inverse surface normal.

A problem that may arise when the goal is to minimise the number of views is evident if three faces of a cube sharing a common vertex are considered. Two views would be capable of capturing the complete surface data. It is obvious that the best view of any of these faces cannot be obtained since it would require that the other two be at oblique glancing angles to the cameras line of sight. Therefore, for a group of faces, the best view may be defined as the direction that is simultaneously a minimum angle from each of the corresponding surface normals [9].

4   Developing a viewpoint planning strategy

In this section, a description is given of the methods that have been used to develop our viewpoint planning strategy.

4.1   Defining a search space for viewpoint planning

The viewpoint planning problem can be conveniently defined as a search problem of a type that has been comprehensively described in artificial intelligence literature [10]. Taking this approach, a search space needs to be defined in which potential solutions can be evaluated in an attempt to find the best solution.

It may be intuitive to define the search space in terms of what is being sought, *i.e.* viewpoints. Such a representation is called an *aspect graph* [11]. Nodes in this graph correspond to a set of viewpoints from which the same topological entities are visible, and arcs correspond to transitions from one aspect to another caused by movement in viewpoint that results in a change in the visible topology of a viewed object or scene. There are, however, drawbacks in this representation, the most relevant of these are the complexity in generating aspect graphs and the potentially huge search spaces (even for moderately simple objects) that may result [12]. Another disadvantage of this representation is that the aspect graph deals with the visibility of object features and, hence, further constraints that may affect viewpoint selection are *not easily* incorporated into the representation.

For these reasons an alternative representation has been investigated. Rather than deal with viewpoints, a set of faces suitable for viewing from a common viewpoint are sought. In our representation the search space is defined by a graph in which the nodes correspond to faces of the object of interest and arcs between two nodes represent both faces satisfying all necessary constraints simultaneously. Hence, if face visibility is the only constraint being used then an arc would connect two nodes if the two corresponding faces are visible from a common point (or region) in space. This representation has the advantage that it can be generated from geometric information readily held in the stored CAD model and visibility information computed at an earlier stage[13] (other relevant information relating to other constraints on a viewpoint may also be incorporated). The graph also has definite manageable size, *n* nodes for *n* faces, and knowing the exact size of the graph before construction leads to a simpler representation of the structure. The graph representation of a simple object using visibility as the only constraint on viewpoint is shown in Fig. 1.

The graph can be conveniently and efficiently represented as a 2 dimensional array. Elements of the array have values determined by the constraints imposed on pairs of faces. For example, the graph in Fig. 1 using visibility constraints and surface normal information would be stored as the array shown in Table 1.
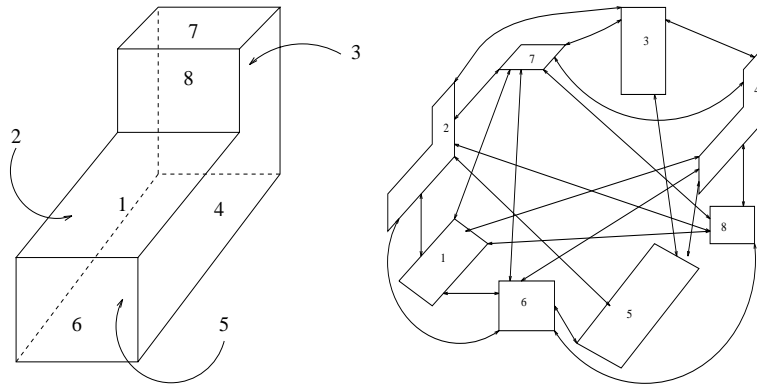
Figure 1: Search space for a simple object

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
R_1	–	90	–	90	–	90	0	90
R_2	90	–	90	–	90	90	90	90
R_3	–	90	–	90	90	–	90	–
R_4	90	–	90	–	90	90	90	90
R_5	–	90	90	90	–	90	–	–
R_6	90	90	–	90	90	–	90	0
R_7	0	90	90	90	–	90	–	90
R_8	90	90	–	90	–	0	90	–

Table 1: Connectivity matrix for L shape block.

In the array shown in Table 1, entries in the range $0 \leq \theta \leq 360$ correspond to the angle θ between the surface normals of two compatible faces. Hence, entries in this range correspond to two faces that satisfy all necessary constraints simultaneously and consequently to two connected nodes in the graph. Incompatible faces, *i.e.* those that do not satisfy all constraints simultaneously and hence relate to unconnected nodes in the graph are represented by entries with a null (–) value. When other constraints are imposed for which there are values that also have properties that are suitable to distinguish faces suitable for viewing from a given viewpoint, these can be combined with the surface normal values and any other meaningful values into a cost function. The cost function is then used as a basis for determining faces to be viewed from a particular viewpoint. The main aim here is to highlight that a flexible, generic cost function has been developed — space limitations prevent further discussion and illustration of this matter here, for further results refer to [14].

4.2 Formulating a viewpoint planning strategy

Given the representation that has been chosen, the first stage of the planning problem can be stated as trying to group faces together into sets that can be viewed from a common viewpoint in such a way that a minimum number of viewpoints is required to view all faces. Described in this way, the problem is essentially one of set partitioning. If a minimum number of views is desired then these partitions are disjoint subsets. This is ignoring the possibility that disjoint sets might not be desirable for object inspection where feature inter-relationships may be needed. However, if the views are registered sufficiently well the relationships may be determined anyway. A solution to finding the minimum number of viewpoints is to generate all possible partitions and by applying some heuristic to each, determine which forms the best plan. Looking at all combinations, *i.e.* all viewpoint

Repeat
  Repeat
    Start with the row having the lowest total - call this *current*.
    Search *current* row and find the lowest next value - this is in position *i*,
     (this corresponds to the row *i* with the lowest total).
    Add *current* face to viewpoint list.
    Blank out entries in row *i* corresponding to those blanked out in row *current*.
    current = i.
  Until all entries in current are blank.
Until all faces are visited.

Table 2: The Viewpoint Planning algorithm.

plans, would guarantee the minimum solution is found. For simple objects, analysing all possible combinations is possible. However, for objects with even a moderate number of faces, the complexity increases dramatically[15].

A better approach is to consider all possibilities by searching the graph in a breadth first fashion. However, the computation time to perform a complete search is not tractable. Therefore, rather than attempt to find the best possible plan it is more logical to look for an acceptable plan that can be computed in a reasonable time. Our planning method computes an approximation to the best viewpoint plan. First, the largest set of faces suitable for imaging from a single common viewpoint is sought. These faces are then removed from the set of all faces and the largest set of suitable faces is found from the remaining faces. This procedure is continued until all faces have been considered. In the graph representation used, two faces are suitable for viewing from a common viewpoint if their corresponding nodes are linked by an arc. For a set of *n* faces to be suitable for viewing from a common viewpoint, the node corresponding to each face in the graph must be connected to all *n-1* other nodes corresponding to the other *n-1* faces. Therefore, the search is for the largest fully connected subgraphs, or *cliques*. Determining the maximum clique of a graph is an NP-hard problem.

A number of heuristics can be applied to aid solution to this problem[16]. The approach taken to our viewpoint planning problem is one of constraint satisfaction. Instead of minimising the number of views, it attempts to maximise the number of faces satisfying all constraints. Therefore, the search is for the largest set of faces simultaneously satisfying all constraints.

Our solution allocates object faces into groups that satisfy all constraints simultaneously. We use the concept of a clique to find a set of object faces suitable for viewing from a common viewpoint. Our algorithm (Table 2) maintains a current viewpoint list holding the tags of those faces that are simultaneously visible and have been deemed by the algorithm as suitable for imaging from a common viewpoint. Face tags are added to the current viewpoint list on the basis of which has the lowest row sum in the array representation (see, for example, Table 1). When the longest possible list has been found, (this occurs when the row in the array representation corresponding to the current face has all blank entries), the nodes in the list are removed from the graph. The faces in the list at this point correspond to a set of faces that will be viewed from a common viewpoint. The search is then repeated on the graph to find the next viewpoint, this repetition continues until all nodes have been removed from the graph. Each viewpoint list found corresponds to an approximation to the maximum clique in the current state of the graph.

The rationale behind the strategy for choosing the first (and successive) face is that

blanked entries, corresponding to unconnected nodes in the graph, are given a high score. Hence, the row with the lowest total will have the least blanked entries (*i.e.* has common visibility with most other faces) and, depending on constraints set, also has closest surface orientation to all other faces. For these reasons, using this selection strategy is likely to choose a good first (next) node.

When visibility is a constraint on viewpoint selection, applying the algorithm as it is given may compute cliques that do not correspond to acceptable face groupings for viewing from a common viewpoint. An example of such an instance is shown in Fig. 2, where faces A, B and C do not have a common visibility region. However, in the graph corresponding to this object, {A, B, C} is a clique. To avoid occurrences of these cliques, a single intersection test on the corresponding visibility regions must be performed by the *underlying* geometric modeller for each node being added to a clique (face being added to a viewpoint list). This ensures that all corresponding faces have a common visibility region. This operation is a basic geometric modeller operation and introduces no significant overheads to our method as many more general intersections are being performed already.
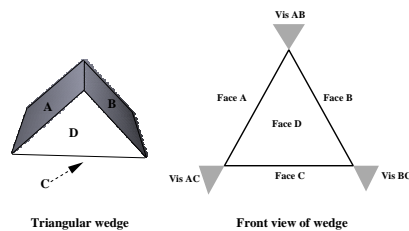
Figure 2: An object and the common visibility regions of faces A, B and C

5 Viewpoint Selection

In this section, consideration is given to the second phase of planning, selecting viewpoints to view face groups selected by the algorithm given in the previous section. Methods for viewpoint selection are given for two different view acquisition setups.

5.1 View acquisition using an eye-in-hand camera

In this configuration, it is assumed that a camera is mounted on a robot arm that has sufficient dexterity to position the camera at any viewpoint relative to the object. Potential viewpoints surrounding the object define a *viewsphere*. The camera remains at a fixed distance to the object, determined at a preliminary calibration stage [13]. The distance between the object and camera defines the radius of the viewpshere.

Output from the planning algorithm is in the form of a list of face groups that are suitable for viewing from a common viewpoint. To determine the region of acceptable viewpoints for viewing all faces in a group, the visibility regions of the faces in the list are intersected. The result is a generalised cone from which any direction inside is capable of viewing all the corresponding faces. Intersecting this cone of view directions with the viewsphere results in a region of viewpoints on the viewsphere that are suitable for viewing the face group. This region of viewpoints can be considered as representing

a set of candidate viewpoints, from which the best is chosen, *i.e.* the viewpoint that has a minimum angle to all corresponding face normals (or where necessary, the closest direction to this lying inside the intersected visibility regions). The chosen viewpoint is then returned as a point on the viewsphere.

5.2   View acquisition using a fixed camera and a turntable

Consideration is now given to selecting viewpoints that are suitable for viewing the face groups selected by the planner using a fixed camera and a turntable to vary the viewpoint relative to the viewed object. The turntable only has one degree of freedom, an angular rotation about its centre, hence the range of possible viewpoints is also restricted to a single degree of freedom about the object. It is assumed that the camera has been configured so that its optical axis is directed at the centre of the turntable.

In this setup, a viewsphere (Fig. 3) is defined by the turntable centre and the camera position. Essentially, the algorithm given made no assumption on possible viewpoints and so a viewpoint required to view a chosen set of faces could potentially lie at any point on the viewsphere. In this setup possible viewpoints that can be achieved are restricted to those lying on a circle on the viewpshere determined by the pose of the cameras.
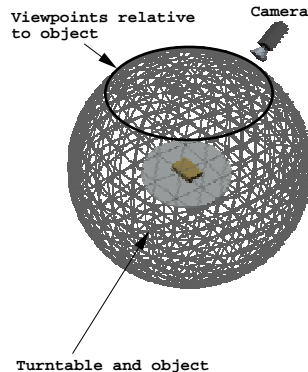
Figure 3: Circle defining possible viewpoints relative to the viewed object

The method given previously for finding the region of acceptable viewpoints to view a group of faces (intersecting the visibility regions of the corresponding faces to find viewpoints capable of viewing all faces simultaneously) is insufficient in the setup now being considered. For two features to be seen from the same sensing position it is necessary that a common viewpoint found as previously, *i.e.* lying in the intersection of the respective visibility cones, also lies on the circle defined by the work-cell configuration. For practical purposes, (to avoid issues of accuracy during earlier computation), the circle defined by the camera and the centre of the turntable are used to create a conical surface. This conical surface forms the locus of possible optical axis directions relative to the object. In the simplest case, when the cameras lie at zero degrees to the turntable, this surface becomes a plane. A valid viewpoint is one that lies in the intersection of the corresponding faces visibility regions and also lies on the conical surface. This additional constraint requires an extra solid modelling intersection test to be performed at Step 4 of the algorithm to ensure that a face being added to the list can be viewed with the other faces already in the list, given the camera position relative to the object. The result of this is a set of possible viewpoints lying on a curve. The best viewpoint is chosen as previously and then transformed to a turntable angular rotation from an initial reference position.

The planner is also being extended to deal with viewpoint planning for a stereo camera and laser acquisition system. Essentially additional constraints are required to account for viewpoint intersections from two cameras and a laser.

6 Results and Discussion

In this section, examples are given that demonstrate the practical capability of the viewpoint planner. Results are given in two forms, graphical output from the solid modeller and real results from a vision acquisition system that uses a turntable and fixed camera to achieve multiple views. Space limitations prevent further discussion of results here. Results presented here demeonstrate that our planner can operate in a variety of domains — for further results refer to [14].

6.1 Example 1: The L-shaped block

Results are given for the L-shaped block shown in Fig. 1. Table 3 shows part of the output from the viewpoint planner when no restrictions are applied to potential viewpoints. The object in this case hasn't been specified as lying on any surface and, hence, none of its faces are considered to be obscured by any part of its environment. Fig. 4 shows the two resultant views produced when the viewpoints are fed back into the solid modeller. By referring to Fig. 1, it can be seen that the viewpoints chosen give clear views of the faces indicated.

```
General viewpoint plan.
Faces in current viewlist: 1,7,2,6,8
DIRECTION: (23.092810 -18.815058 21.343947)
Faces in current viewlist: 3,4,5
DIRECTION: (-21.845330 19.320590 -20.149477)
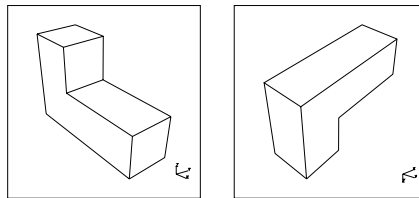```

Table 3: Sample output from the planner



Figure 4: View 1. To view faces 1,7,2,6,8. View 2. To view faces 3,4,5.

Fig. 5 shows the views chosen by the planner when possible viewpoints are restricted to a single degree of freedom relative to the object, *i.e.* suitable when a turntable is used to enable the acquisition of multiple views. In this instance, the object has been specified as resting on face 5, (*i.e.* the face with surface normal in the negative *z* direction), and the cameras pose is specified as being parallel and level with the turntable.

## 6.2 Example 2: A more complex test object

Fig. 6 shows the six views that were computed to view all possible faces of the object given the setup of the acquisition system used and a specified object pose. In this case, two
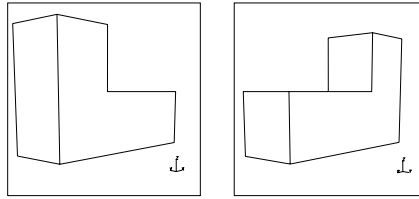
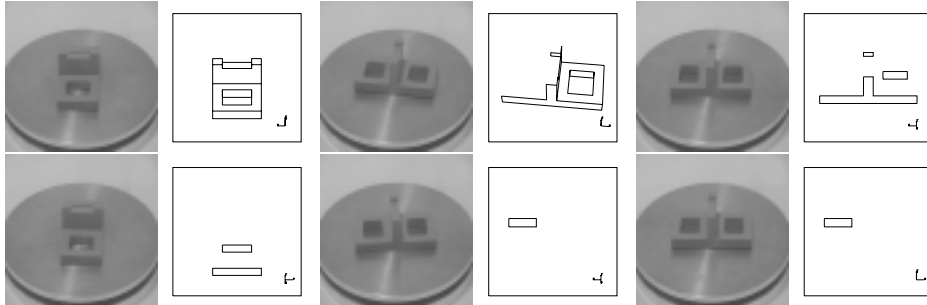Figure 5: View 1. To view faces 2,3. View 2. To view faces 4,6,8



Figure 6: Real views and predicted views to view each face group. Only the faces to be viewed are shown in the predicted views.
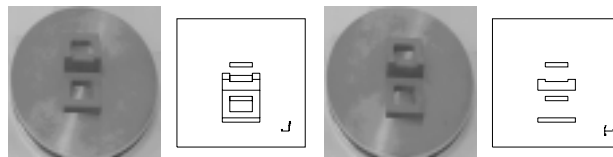


Figure 7: Change in camera position that allows previously non-imageable faces to be seen in their entirety from a single view.

faces besides the face obscured by the turntable are deemed to be impossible to see as a whole from any viewpoint given the current object and camera poses. Fig. 7 demonstrates how the problem of hidden faces can sometimes be overcome by changing camera pose. Here, the camera angle relative to the turntable has been increased and the planner now specifies the previously hidden faces to be suitable for viewing given the new camera pose and the pose of the object on the turntable.

In the experiment shown in Fig. 6 no constraint was imposed on the range of surface orientations that were chosen for viewing in a single image. Hence, in the second image a face has been chosen for viewing at a very oblique angle. However, in the second experiment, shown partly in Fig. 7, such a constraint was imposed. The face previously viewed at an oblique angle is now considered in a different view, shown in the second image, which results in a far better orientation relative to the camera.

# 7   Summary and Conclusion

An algorithm has been presented that computes viewpoints suitable for viewing the complete surface area of an object. The approach taken to finding these viewpoints is based on constraint satisfaction. A graph representation is used to define the search space of solutions in which maximally connected subgraphs are sought since these correspond to

faces that satisfy all constraints simultaneously and hence are suitable for viewing from a common viewpoint. The algorithm was first given in a more general form where viewing directions on the Gaussian sphere are sought. These are suitable for a work-cell in which a camera has been mounted on the end of a robot arm. An augmented algorithm was then given that is suitable for computing viewpoints in more restricted space, this would used when multiple views are obtained using a turntable. Practical results have been given that demonstrate the capability of the planner. Although not demonstarted here (due to space limitations) we can show that our planner is efficient in its operation ([14]). Our formulation of the search space is convenient and efficient, in comparison to other techniques, and has the advantage of manageable and known size of the search space prior to viewpoint deteremniation.

# References

[1] J Maver and R Bajcsy. Occlusions as a guide for planning the next view. *IEEE transactions on pattern analysis and machine intelligence*, 15(5):417–433, May 1993.

[2] V. Hlavac, A. Leonardis, and T. Werner. Automatic selection of reference views for image-based scene representations. In Buxton B. and Cipolla R., editors, *Proceedings of the European Conference on Computer Vision*, number 1064 in Lecture Notes in Computer Science, pages 526–535. Springer Verlag, April 1996.

[3] Lambert Ernest Wixson. *Gaze Selection for Visual Search*. PhD thesis, University of Rochester, Dept of Computer Science, University of Rochester, Rochester, New York, 1994.

[4] C K Cowan and P D Kovesi. Automatic sensor placement from vision task requirements. *IEEE transactions on pattern analysis and machine intelligence*, 10(3):407–416, May 1988.

[5] K D Gremban and Ikeuchi K. Planning multiple observations for object recognition. *International journal of computer vision*, 12(2/3):137–172, 1994.

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

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

[8] S O Mason and A Grun. Automatic sensor placement for accurate dimensional inspection. *Computer Vision and Image Understanding*, 61(3):454–467, May 1995.

[9] A.D. Marshall and R.R. Martin. *Computer Vision, Models and Inspection*. World Scientific, River Edge, NJ, 1992.

[10] G. F. Luger and W. A. Stubblefield. *Artificial Intelligence : structures and strategies for complex problem solving*. Benjamin Cummings, 390 Bridge Parkway, Redwood City, CA 94065, 1992.

[11] J.J. Koenderink and A.J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[12] Harry Plantinga and Charles R. Dyer. Visibility, occlusion and the aspect graph. Technical Report 736, University of Wisconsin, Madison, Dept. of Computer Science, University of Wisconsin, Madison, December 1987.

[13] D.R. Roberts and A.D. Marshall. A strategy for complete inspection of three-dimensional objects by active stereo computer vision. In *International Workshop on Advanced Robotics and Intelligent Machines*, April 1996.

[14] D.R. Roberts and A.D. Marshall. Viewpoint Selection for Complete Surface Coverage of Three Dimensional Objects Dept of Computer Science, Cardiff University Technical Report No. 97008, *http://www.cs.cf.ac.uk/Research/Rrs/1997/detail008.html*, 1997. *Also available* as EU Copernicus Grant No 1068 "RECCAD", Deliverable Document 3, Article 13, 1997.

[15] T.A. Dowling. Stirling numbers. In J.G. Michaels and K.H. Rosen, editors, *Applications of Discrete Mathematics*, chapter 6, pages 95–12. McGraw-Hill, 1991.

[16] Arun Jagota. Approximating maximum clique with a hopfield network. Technical Report 92-33, State University of New York at Buffalo, Dept. of Computer Science, State University of New York at Buffalo, USA, 1992.