REDUCING VIEWSPHERE COMPLEXITY

Robert B Fisher

DAI RESEARCH PAPER NO. 481

Proc. Eur. Conf. on AI,
Stockholm, Sweden, 1990.

Reducing Viewsphere Complexity

Robert B. Fisher
Dept. of Artificial Intelligence, University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, Scotland, United Kingdom

Abstract

This paper shows that extreme fragmentation of the viewsphere can be reduced by a model hierarchy and simplification of the relationships represented by the viewsphere. This means the viewsphere representation is then more suitable for use in model-based object recognition systems, even with non-rigid objects.

Keywords: viewsphere, hierarchical modeling, 3D scene analysis, aspect graph

1 Introduction

Characteristic view, viewsphere or aspect graph techniques have been proposed for three dimensional scene analysis[1,2,5,9]. The viewsphere is a notional sphere surrounding the modeled object, (usually at a large distance, implying orthographic projection) over which a viewer can observe the object from all directions. It is usually partitioned into regions where some set of perceptual relations, such as the appearance of given features, remain constant. From each viewpoint, one can build a graph representing the topological structure of the object's appearance. Further one can build another graph (the aspect graph) representing the structure of the regions on the viewsphere.

The aspect graph representation is useful for computer vision systems because it precompiles 3D model shape to make explicit certain aspects of 2D appearance (depending on the criteria differentiating regions on the viewsphere). Otherwise, it is necessary to transform between 3D model and 2D image representations, which can be complex and time-consuming (e.g. it may require a complete image synthesis and analysis for complex questions).

Research has considered polyhedral objects[3,6,11], where the viewsphere is partitioned according to the visibility relations between between planar surface facets and curved surfaces[2,8,10], where the viewsphere is partitioned according to the singularities of surface visibility.

It is often felt that this approach is too computationally complex to be used for practical computer vision systems, because even simple objects produce viewspheres with great complexity[7]. Chen[3] showed the complexity of the viewsphere for convex polyhedra was $O(n^2)$ in the number of faces and Gigus *et al*[6] claimed it was $O(n^6)$ in the number of vertices for general polyhedra. If we consider objects with curved surfaces, all interfeature relationships and general occlusion, then the complexity is even greater.

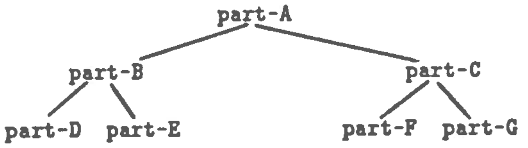
It is also unknown whether the viewsphere is useful for recognizing objects whose shape can vary. Here, one might need to develop separate viewspheres for different values (or ranges) of shape parameters, because shape changes cause changes in the relative position of features, and thus potentially causes differences in occlusion or appearance relationships.

2 Partitioning the Viewsphere

Recent work here suggests viewsphere complexity can be reduced if some simplifications are exploited: (1) objects are hierarchically represented and (2) only feature depth-order relationships between major surface patches are considered. We assume that the visibility analysis occurs at a great distance (i.e. orthographic projection), and disregard privileged viewpoints (i.e. those that become small as distance from the object increases).

2.1 Exploiting the Model Hierarchy

Instead of producing a viewsphere partitioned by all interfeature relationships, we generate a hierarchy of viewspheres that only considers the occlusion relationships between features at each level of a model hierarchy. For example, for the model hierarchy:



the viewsphere at level A only considers the occlusion relationships between parts B and C as wholes, and disregards any finer relationships arising from substructure relating to parts D, E, F and G. Another viewsphere is developed for B, reflecting the relationships between D and E, and so on.

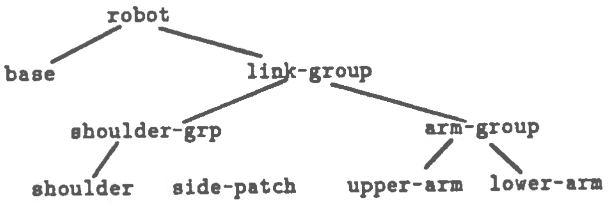
This replaces one form of complexity for another, but it results in a dramatic reduction of the magnitude of the complexity, because there are at most five distinct regions in A's viewsphere (depending on the relative sizes and positions of B and C): C-hides-B, C-occludes-B, C-side-by-side-B (or mutually occlude), B-occludes-C and B-hides-C. Assuming both objects are similarly sized, somewhat complex and well-connected, then then it is likely that only the regions where C-side-by-side-B, C-occludes-B and B-occludes-C hold will be significant in size.

2.2 Simpler Visibility Relationships

At each level of the hierarchy, rather than consider all occlusion relationships between all features specified at that level, we only consider depth-ordering relationships between large visible surfaces and groups of surfaces. This eliminates a large amount of minor view-potential detail created by occlusion relationships between point and boundary segments, and interactions with non-visible back-facing features. Small regions of the viewsphere are also eliminated. Thus, the complexity of representation at a single level is similar to that of the polyhedral case, except that usually there are only a few large surface features (i.e. N remains small).

3 Example

We illustrate the points discussed in the last section with an example based on the robot model shown in Figure 1. This model has the following subcomponent hierarchy:



For this model, there are nine viewspheres for the nine subcomponents. The number of regions in the viewspheres for each subcomponent is:

Model	View Regions	Subcomp	Surfaces
robot	3	2	-
base	1	-	1
link-group	3	2	-
shoulder-grp	3	2	-
arm-group	3	2	-
shoulder	6	-	2
side-patch	3	-	2
upper-arm	6	-	8
lower-arm	5	-	10

The table lists the number of significant regions in the viewsphere for each level in the viewsphere/model hierarchy. The number of subcomponents and surfaces is given by the model. Each level consists of only subcomponents or surfaces. In the former case, only occlusion (i.e. depth ordering) relations between the subcomponents contributes to the different viewpoints; hence the robot model has only 3 views, depending on whether the base is in-front-of, behind or beside the link_group. Subcomponents composed of only surfaces have their full partitioning of the viewsphere, but, while the size of some subcomponents might suggest a complex viewsphere, the number of distinct visibility regions is reduced because of parallel surfaces. Additionally, only one instance of a symmetric component is represented.

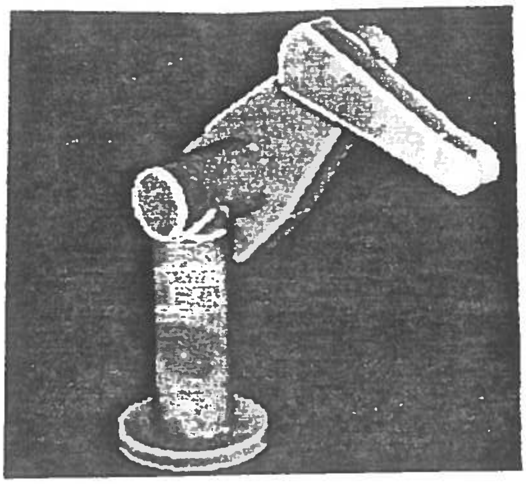


Figure 1: Robot Model

The two simplifications produce a remarkable reduction of the complexity, as the model contains approximately 23 surfaces, reducing something like $2^{23-1} \approx 4,000,000$ visibility regions in a non-hierarchical visibility sphere to 33 in the simplified viewsphere hierarchy. It is noteworthy that this visibility analysis applies for all orientations of the three robot joints, assuming non-degenerate positions, whereas conventional analysis could not cope with the range of shapes. This reduction is also present in all other objects we have modeled so far.

4 Implementation

To represent visibility information, the SMS modeling system[4], links *visibility groups* to the geometric models. Each visibility group describes the appearance of the object from a characteristic viewpoint and its representation consists of three main descriptions: (1) a list of visible and tangential (possibly visible) features, (2) the occlusion relationships between adjacent subcomponents and (3) the geometric position constraints over the positions from which the viewer can see the given group.

The SMS visibility groups for the robot link-group are:

```
(VDFG link_group (
    /* arm fully in front of shoulder */
    (VIS_GROUP (arm_group)
     TAN_GROUP ( NONE )
     CONNECT_CONSTRAINTS ( NONE )
     NEW_FEAT_CONSTRAINTS ( NONE )
     POSITION_CONSTRAINTS (
       ((VIEWER DOT MAP (0,0,-1)>0.9))))

   /* shoulder visible behind arm group */
    (VIS_GROUP (arm_group shoulder_grp)
     TAN_GROUP ( NONE )
     CONNECT_CONSTRAINTS ( NONE )
     NEW_FEAT_CONSTRAINTS (
       (VPD_POFEAT shoulder_grp BY arm_group))
     POSITION_CONSTRAINTS (
```

```
        ((VIEWER DOT MAP (0,0,-1)<0.9))
        ((VIEWER DOT MAP (0,0,1)<0.0))))

    /* shoulder in front of arm */
    (VIS_GROUP (arm_group shoulder_grp)
     TAN_GROUP ( NONE )
     CONNECT_CONSTRAINTS ( NONE )
     NEW_FEAT_CONSTRAINTS (
        (VPD_POFEAT arm_group BY shoulder_grp))
     POSITION_CONSTRAINTS (
        (VIEWER DOT MAP (0,0,1)>0))))))
```

Figure 2 shows the three visibility groups. VIS_GROUP states which subcomponents at the current level of the geometric model are in this visibility group. VPD_POFEAT states that one feature is partially obscured by another. Position constraints are expressed as bounds on dot products between the given feature vectors (transformed by the estimated object position) and the observer direction.

In the IMAGINE II system[5(pp255–260)], SMS's visibility groups are used to specify: (1) which groups of features simultaneously accumulate evidence for when invoking models, and (2) which subcomponents should be instantiated with image evidence when doing model-based verification, for a given viewpoint. If there is strong evidence for one set of features being present, then the full geometric model is invoked (plus the knowledge of which viewpoint). At the given level in the model hierarchy, the visibility group information then helps the matching process by reducing the range of orientations and eliminating occluded subcomponents. The occlusion relationships help reject invalid hypotheses.

Once analysis is complete at one level, the results are used at the next. There, visibility analysis is not always necessary, because previously recognized (bottom-up) subcomponents largely entail reference frame consistency checking. On the other hand, the visibility information can be used to guide top-down search for missing subcomponents.

5 Discussion

From the results of the previous section, we can see that there is a dramatic reduction in viewsphere complexity, at the cost of some loss of detail in the visibility information, and simplified representations of the visibility regions. This has not affected our object recognition work so far.

There is still potential for exponential growth of the number of visibility regions, but the hierarchy and simplification techniques have kept it under control for the objects that we have modeled (including: oilcan, pyramid, ashtray, widget, telephone handset, face).

Acknowledgements

This work was funded by the University of Edinburgh. I'd like to thank M. Cameron-Jones and J. Hallam for helpful advice.

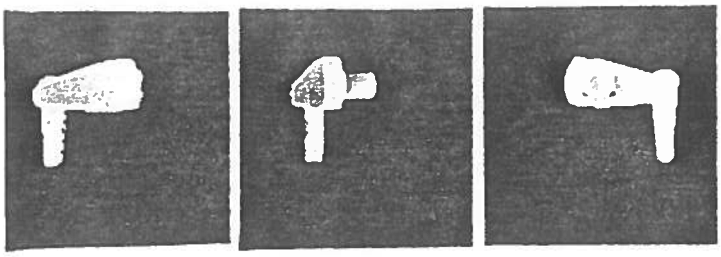


Figure 2: Link Visibility Groups

Bibliography

1. Callahan, J., Weiss, R., "A model for describing surface shape", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp 240-245, 1985.
2. Chakravarty, I., "The use of characteristic views as a basis for recognition of three dimensional objects", IPL-TR-034, Image Processing Laboratory, Rensselaer Polytechnic Institute, 1982.
3. Chen, G., "Appearance analysis from CAD models", LIFIA report, RR 756-I-IMAG-88, 1988.
4. R. B. Fisher, "SMS: A Suggestive Modeling System For Object Recognition", Image and Vision Computing, Vol 5, pp. 98-104, 1987.
5. R. B. Fisher, From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis, John Wiley and Sons Ltd, Chichester, 1989.
6. Gigus, Z., Canny, J., Seidel, R., "Efficiently computing and representing aspect graphs of polyhedral objects", Proc. 2nd Int. Conf. on Comp. Vis., pp30-39, 1988.
7. Koenderink, J. J., van Doorn, A. J., "How an ambulant observer can construct a model of the environment from the geometrical structure of the visual inflow", in Hauske and Butenandt (eds), KYBERNETIK, pp224-247, 1977.
8. Koenderink, J. J., van Doorn, A. J., "The internal representation of solid shape with respect to vision", Biological Cybernetics, Vol 32, pp 211-216, 1979.
9. Minsky, M. "A Framework for Representing Knowledge", in Winston (ed), The Psychology of Computer Vision, pp 211-277, McGraw-Hill, New York, 1975.
10. Reiger, J. H., "On the classification of views of piecewise smooth objects", Image and Vision Computing, Vol 5, No 2, pp91-97, May 1987.
11. Stewman, J., Bowyer, K., "Creating the perspective projection aspect graph of polyhedral objects", Proc. 2nd Int. Conf. on Computer Vision, pp494-500, 1988.