Model-Based Vision using a Planar Representation of the Viewsphere

A. Rydz, G. D. Sullivan, K. D. Baker

Intelligent Systems Group, Department of Computer Science,
University of Reading, RG6 2AX.UK.
Andrew.Rydz@reading.uk.ac

Viewpoint indicates the viewing direction of an observer with respect to an object and can be described by a point on a sphere. This paper describes a planar representation of viewpoint that can be used by a model-based vision system to represent and reason about the visibility of object-features. The representation can be used for both on-line and off-line reasoning. With respect to off-line reasoning it is shown how it is possible to represent the visibility of an object's features as a quadtree and characterize an object by a set of canonical viewpoints. With respect to on-line reasoning it is shown how it is possible to reason about the presence of an object in the image and to estimate the possible viewpoints. Once a group of detected features has been hypothesized to belong to an object and a set of possible viewpoints determined, it is then possible to confirm this hypothesis and exclude unlikely viewpoints by using precomputed information about the pairwise relationships between image features.

Model-based Vision using single 2D images involves reasoning about the relationship between image-features and object-features. The search-space associated with such reasoning is potentially very large since only 2D features are available. It is therefore very difficult to formulate tight geometric constraints of the type discussed by Grimson[1] to help reduce the search space during model matching. Other ways of reducing the search space must be found.

The approach adopted here uses low-level feature groups as cues to hypothesize the presence of an object in the image and the possible set of viewpoints with respect to it. This combination of feature grouping and viewpoint reasoning can be achieved by precomputing information about the visibility of an object's features as a function of viewpoint and designing grouping operations that look for these features in the image. The visibility of each object feature is stored by using a planar representation of the viewsphere. This representation is based on a quadtree encoding of the surface of a tetrahedron.

Once an object has been hypothesized, confirmation of its presence and a pruning of the set of all possible viewpoints can be achieved by using precompiled information about the pairwise relationships between image-features. Such information will have to be stored for each viewpoint. After unlikely viewpoints have been eliminated in this way the perspective transform can be inverted[2] and iconic evaluation performed[3,4].

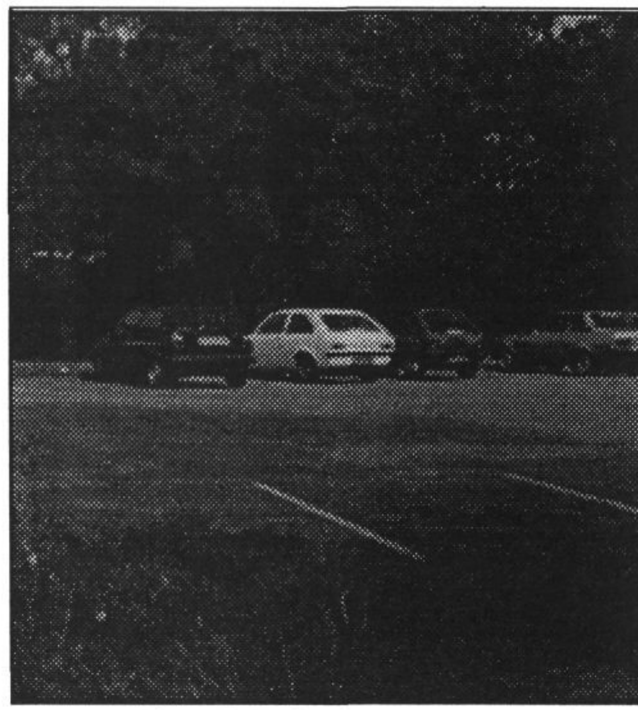


Figure 1. Typical image used

This paper is concerned with describing the above mentioned model-based vision strategy and preliminary work towards its implementation. For this purpose images of a car in an outdoor scene are being used such as that shown in Figure 1. As the work reported is strongly related to other work in this field a brief review of the relevant literature is given first.

RELATED WORK

The Model-Based Vision strategy outlined in this paper draws heavily on the work of Koenderink and Van Doorn[5,6], Chakravarty and Freeman[7], Goad[8] and Lowe[9]. Their work will be discussed in turn concentrating on those aspects important in the present context.

Koenderink and Van Doorn

Koenderink and Van Doorn carried out the first formal treatment of viewpoint in relation to general vision systems, in particular human vision. They argued "...that the internal model of any object must be in the form of a function such that for any intended action the resulting reafferent is predictable"[5, (p. 211)]. The object-function can be derived explicitly for the case of visual perception of rigid bodies by ambulant observers. Because the function depends on physical causation and not

AVC 1988 doi:10.5244/C.2.1

physiology, Koenderink and Van Doorn were able to make a priori statements about the nature of the object-function. Their characterization of this function had a quantitative and a qualitative structure. The qualitative structure (visual potential) provides a framework for the quantitative information. The object-function represents in a concise way the visual experience of an observer as he moves around an object. The visual potential (qualitative structure) of a tetrahedron is shown in Figure 2. Each aspect (node on the graph) represents a set of connected viewpoints for which the topological structure of the 2D projection of the object under view remains the same. It is the qualitative nature of the internal representation of an object, the object-function, that has influenced the model-based vision strategy described later.

Chakravarty and Freeman

Chakravarty and Freeman took viewpoint into account by representing an object in terms of canonic 2D models. The reduction in dimensionality is achieved by factoring the space of all possible perspective projections of an object into a set of characteristic views, where each such view defines a set of viewpoints over which all projections are topologically identical and related by a linear transformation. Some of the characteristic views of a polyhedral shape are shown in Figure 3. The ideas of Chakravarty and Freeman are related to those of Koenderink and Van Doorn. The major difference is that they use a geometric model to derive their object function. This approach is also used here for deriving the object-function.

Goad

Goad, like Chakravarty and Freeman, uses a multiview feature model of an object. Unlike the previously mentioned authors, Goad did not try to factor the set of all possible projections into topologically equivalent classes. For practical reasons he chooses to represent the

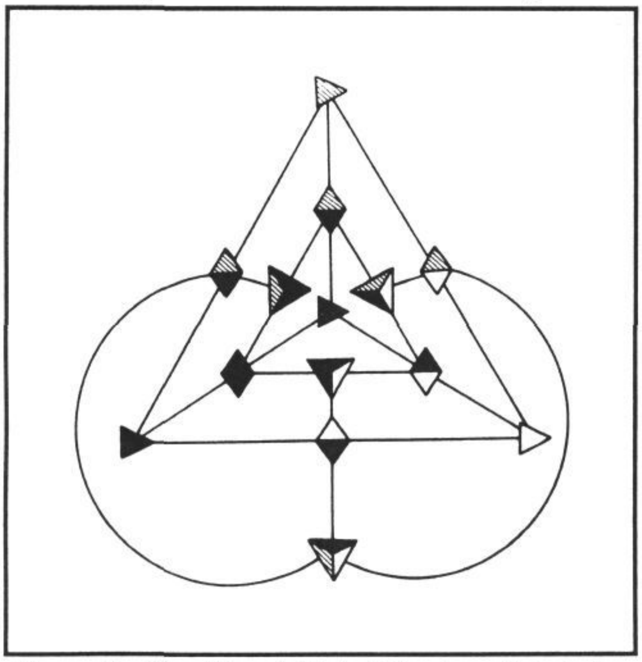


Figure 2. The Visual Potential of a Tetrahedron [Koenderinck and Van Doorn 1979].

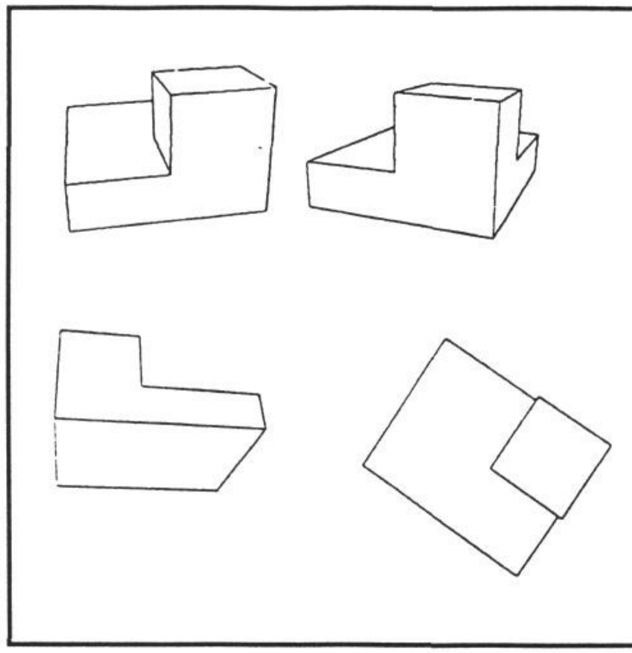


Figure 3. Some characteristic views of a polygonal object [Chakravarty and Freeman 1982].

set of all viewpoint positions by partitioning the surface of a unit sphere into 218 small patches. Each patch on the sphere is associated with a single perspective projection of an object. Goad then precompiled information about the spatial relationship between projected object-features. This information was used during the model matching stage to eliminate unlikely viewpoints.

The matching strategy of Goad is related to the work of Grimson and Lozano-Perez[10] who have experimented with the recognition of simple industrial parts using a constrained viewpoint.

The aspects of Goad's work that are of interest here are (a) his approximation of the viewphere which he used to represent feature visibility and (b) his matching strategy. Both of these aspects have influenced the model-based vision strategy discussed later.

Lowe

Lowe has also proposed a model-based vision strategy, similar to the strategy that is put forward here, which uses feature grouping methods to help solve for viewpoint parameters. The major concept that unifies the work of Lowe is the viewpoint consistency constraint which is defined as follows:

" The location of all projected model features in an image must be consistent with projection from a single view"[11 (p. 57)].

Lowe's algorithm works by carrying out grouping operations on an edge segmented image to extract features unlikely to have occurred by chance. There groupings are then matched one at a time to components of the object model that could give rise to them. Once potential matches have been found, the perspective transform is inverted and viewpoint determined. The computed viewpoint is then used to apply the viewpoint

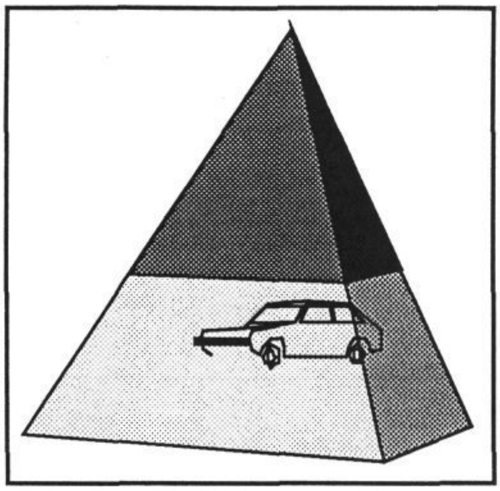


Figure 4.

consistency constraint by extending the match to incorporate predicted features.

Using grouping processes can considerably reduce the search space during model matching. It is this aspect of Lowe's work that is used later.

A PLANAR REPRESENTATION OF FEATURE VISIBILITY

A planar representation of a feature's visibility is described here, based on the tetrahedron and the quadtree representation of spatial occupancy.

Quadtrees

Quadtrees[12] are a useful encoding of spatial occupancy and have been used widely in computer vision and image processing. The quadtree representation is a scale based description and is constructed by a process of cellular decomposition. A rectangular decomposition of the plane is usually used.

Alternative decompositions of the plane are possible[12], for instance the equilateral triangular tessellation of the plane. The equilateral triangular tessellation of the plane has not been widely used because spatial indexing is normally implemented by using a cartesian coordinate frame.

Quadtree representations are useful for representing region information since set operations may be performed on quadtrees with ease. Thus it is very easy to combine regions and find their intersection. There has also been a number of efficient codes developed for representing quadtrees which greatly reduce the amount of storage[13] and the computational requirements for operating on quadtrees.

The View-Tetrahedron

Because of its simplicity it has been decided to use a spherical tetrahedron as the basis for forming a discrete approximation of the viewsphere. The spherical tetrahedron can be constructed by projecting a tetrahedron onto the viewsphere. A denser tessellation of the viewsphere can be achieved by subdividing the faces of the tetrahedron either before or after projection. As the method of approximation and subdivision is of secondary importance here, the reader is referred to Gasson[14] for a discussion of methods for approximating spheres.

An interesting property of the tetrahedron is that when unfolded an equilateral triangle is formed. This property can be exploited to produce a compact 2D representation of the viewsphere based on quadtrees using a triangular tessellation of the plane. This representation has been named the view-tetrahedron.

Representing feature visibility using the view-tetrahedron

Figure 4 shows a car within a tetrahedron. The visibility of the roof is represented by the shaded upper portions on the tetrahedron and was computed by calculating the intersection of the plane of the roof with the sides of the tetrahedron (the surface normal information necessary to do this was obtained from the car model Worrall[15]). The tetrahedron may be unfolded as in Figure 5a. A point on the tetrahedron is then represented by a point in the plane. By using a quadtree to encode Figure 5a it is possible to produce a compact representation of the roof's visibility as it is shown in Figure 5b. A subdivision of the tetrahedrons surface is called a viewpatch.

A viewpatch may not be a connected set of viewpoints as Figure 8h shows. Whether it is or not will depend on the shape of the object and the type of feature chosen. As the connectivity of a viewpatch is important an algorithm for carrying out connected component analysis has been developed.

The viewpatches of other types of object-features can be computed in the same way. As the image-features we detect are related to object surfaces, the visibility of image-features can be defined by reasoning about object surfaces. Therefore the visibility of object surfaces may be used to reason about viewpoint during the feature grouping stage by using logical operations on viewpatches. Viewpatch reasoning, as it has been called, is described in more detail later.

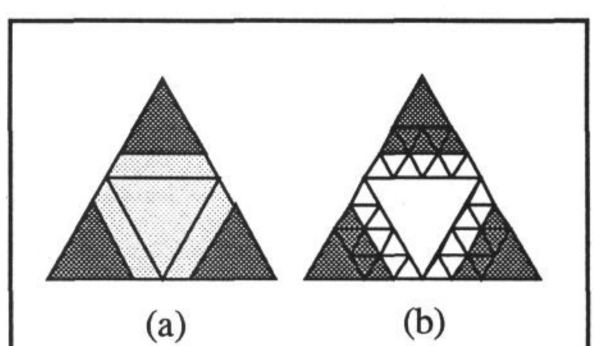


Figure 5. The roof of a car.

CHARACTERIZING AN OBJECT BY A SET OF CANONICAL VIEWS

By reasoning about how the visibility of an object's surfaces change with changes in viewpoint it is possible to identify a viewpatch in which surface visibility does not change. Such viewpatches define equivalence classes on the viewtetrahedron which we refer to as canonical views (CVs). Typically an object will be described by a number of these. Figure 6 shows the 2D projection of a car from 8 different viewpoints each a member of a different CV. Different surfaces are visible from each of these viewpoints

Characterizing an object by a number of CVs is not a simple task as one has to specify in advance the criteria for grouping together different views. The criteria chosen will depend on the nature of the low-level processes. For instance Chakravarty and Freeman characterize a polyhedral object with a number of CVs by grouping together viewpoints for which the junction labelling of the 2D projection of the object are the same.

Information about the size of a viewpatch that represents a CV can be used to define the saliency of a feature. Those features that are most likely to be visible are worth looking for first.

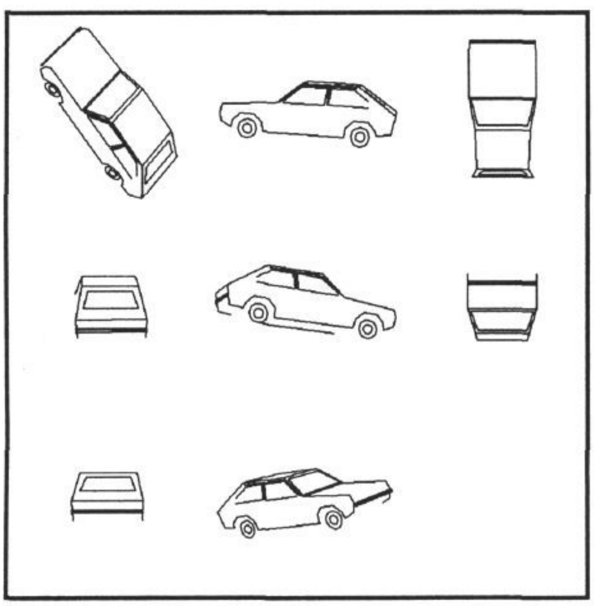


Figure 6. Eight different viewpoints each a member of a different CV.

VIEWPATCH REASONING AND THE RECOGNITION PROCESS

When only 2D cues are available about scene objects it is useful to carry out perceptual organization prior to model matching to reduce the search space. Perceptual organization can reduce the search space in two ways.

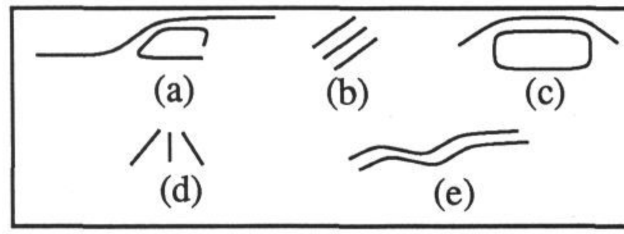
(i) By reducing the possible number of object-feature/image-feature matchings.

(ii) By providing stronger constraints for viewpoint than a single feature.

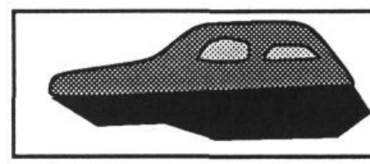
It has proved very difficult to interpret the output of general purpose grouping processes but as the work of Lowe demonstrates it is possible for restricted application domains. This is probably because the role of knowledge in perceptual grouping has largely been ignored[16].

We obtain the following types of feature groups from our low-level grouping processes.

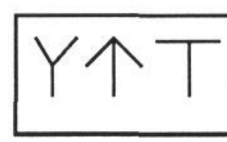
(1) Groups of related open or closed curves:



(2) Groups of related regions.



(3) Junctions:



The exact nature of the low-level grouping processes used will not be discussed here as their performance is well documented in the literature.

The features listed above are extracted during the first stage of the recognition process. The other stages involved are:

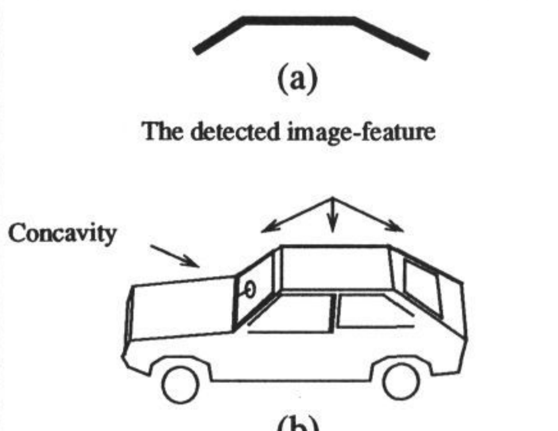
(1) Known constraints on camera position and the orientation of objects in the environment are encoded as viewpatches.

(2) The objects that could give rise to the feature groups found in the image are hypothesized.

(3) Reasoning is performed about the relationship between image-features and object-features. For each possible labelling a set of viewpoints will be generated. Precompiled information about the pairwise relationships between features is used to rule out unlikely viewpoints.

(4) For the remaining viewpoints the perspective transform is inverted and iconic evaluation performed[2,3].

The main difference between the matching strategy described here and that of Goad's is (a) the nature of the representation of feature visibility (b) the fact that grouping operations are used to reduce the possible number of viewpoints that need to be considered prior to



(a)
The detected image-feature

(b)
A possible set of model lines that could give rise to this feature

Figure 7.

matching. This matching strategy and others like it are under development.

To illustrate the technique a simple example has been chosen based on edge information. For the example it is assumed that a feature group has been identified and one possible image-feature/object-feature pairing made. The feature group used is shown in Figure 7a and the possible pairing in Figure 7b. At this stage of the matching process it has been hypothesized that these model lines form part of the occluding contour of the roof of a car which is travelling from right to left. If this hypothesized object-feature/image-feature pairing is correct then the right window of the car must not be visible and the front windscreen, roof and rear window must be visible. A constraint on viewing direction must also be imposed otherwise the projected shape of the car's edges would not be correct. This constraint is derived by precomputing information about the visibility of this particular class of feature .

Using this information to constrain the possible viewpoints from which these lines can be seen, likely viewpoints can be calculated as follows. Firstly, the viewpatches of the bonnet, front windscreen, roof and rear window are ***anded*** with the viewpatch of the extracted feature. The resultant viewpatches are shown in Figures 8a, 8b, 8c and 8d respectively. Secondly, the visibility of the concavity is computed by ***anding*** the viewpatches of the bonnet and windscreen, Figure 8e. Thirdly, the visibility of the right window of the car is computed, Figure 8f, and because we are interested in those viewpoints from which the right side is not visible the ***complement*** of its viewpatch is taken, Figure 8g. Finally, this is ***anded*** with the viewpatch for the concavity, roof and rear window, Figure 8h, to find the visibility of these lines as part of the occluding contour. Figure 8i , represents the viewpatch corresponding to this particular image-feature/object-feature pairing. A

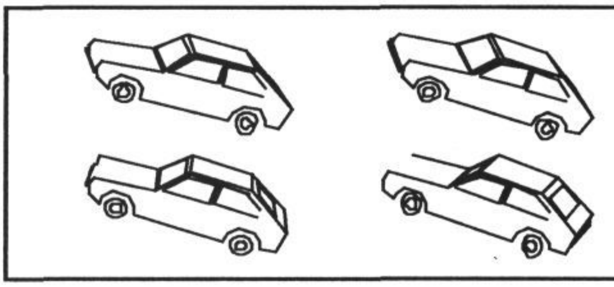
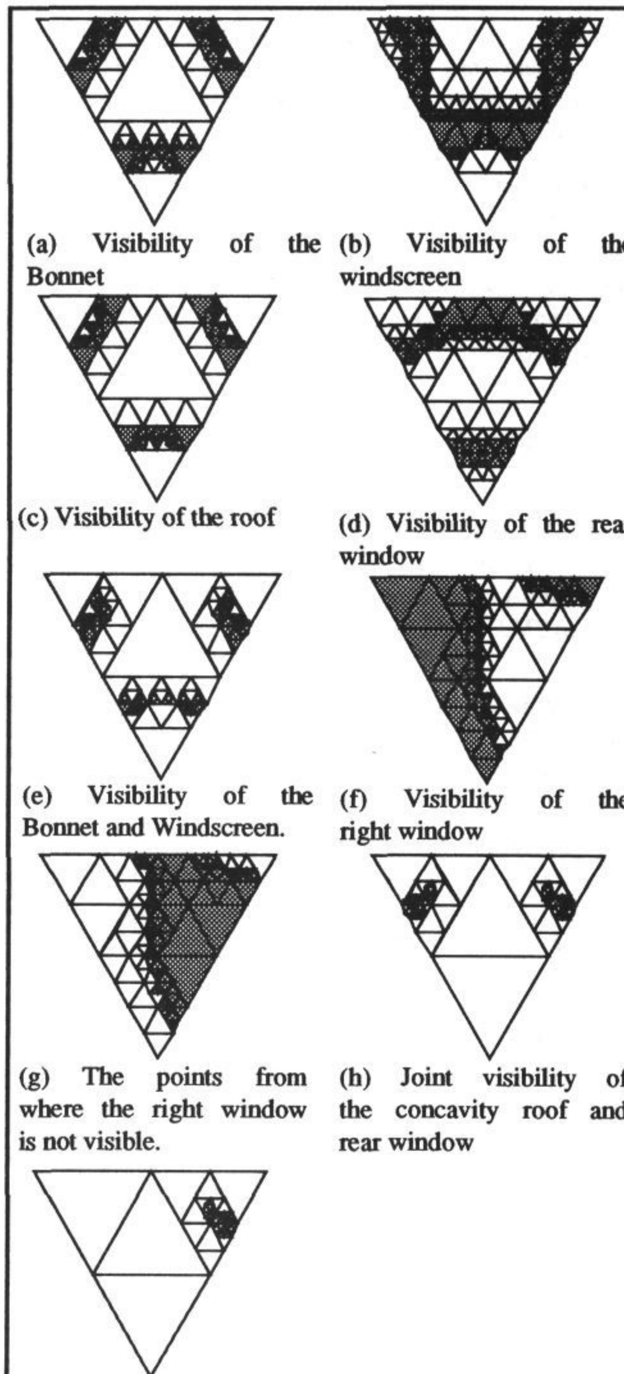


Figure 9.

number of viewpoints from within this viewpatch are shown in Figure 9. The resulting viewpatch can now be used as a basis for matching and represents a considerable reduction in the search. The visibility of other features can also be predicted.



(a) Visibility of the Bonnet

(b) Visibility of the windscreen

(c) Visibility of the roof

(d) Visibility of the rear window

(e) Visibility of the Bonnet and Windscreen.

(f) Visibility of the right window

(g) The points from where the right window is not visible.

(h) Joint visibility of the concavity roof and rear window

(i) The visibility of the Bucket Shape.

Figure 8.

The number of possible image-feature/object-feature pairings possible will be dependent on the shape of the object and the amount of feature grouping that has taken place. The search problems associated with labelling feature groups is reported by Bodington [17]. He also describes how viewpatch reasoning can be used to check the consistency of a labelling.

SUMMARY AND CONCLUSIONS

This paper has described a method for representing viewpoint dependent information such as the visibility of an object's features. Examples of how this technique could be used were also given. It is clear that the ability to carry out viewpoint reasoning in the manner described here will be dependent on the shape of the object and the quality of features output by low-level grouping processes. When the grouping processes fail, and only line segments are available, then it will still be possible to estimate viewpoint using a matching strategy similar to that of Goad's.

Future work will be concerned with the development of matching strategies based on this technique and an assessment of when it is appropriate to use it. An important aspect of this assessment will be to analyze the effect of viewpoint reasoning on the size of the search space.

REFERENCE

1. **Grimson, W. E. L.** (1986), " *The Combinatorics of Local Constraints in Model-Based recognition and Localisation from Sparse Data"*. J. of the ACM. Vol 33, No. 4.

2. **Worrall, A., Sullivan, G. D. & Baker, K. D.** (1988). "Model Based Perspective Inversion," Submitted to AVC88.

3. **Brisdon, K.** (1987), *"Alvey MMI-007 Vehicle Exemplar: Evaluation and Verification of Model Instances"*. In Proceedings of the ACV87, University of Cambridge, 15-17 September 1987.

4. **Brisdon, K., Sullivan, G. D. & Baker, K. D.** (1988),. *"Feature Aggregation in Iconic Model Evaluation"*. Submitted to AVC88.

5. **Koenderink, J. J. & Van Doorn, A. J.,** (1976), *"The Singularities of the Visual Mapping"*. Biological Cybernetics, Vol. 32, No. 1.

6. **Koenderink, J. J. & Van Doorn, A. J.,** (1979), *"The Internal Representation of Solid Shape with Respect to Vision"*. Biological Cybernetics, Vol 32, No. 4.

7. **Chakravarty, I, & Freeman, H.** (1982), *"Characteristic Views as a Basis for 3D Object Recognition"*. In Proceedings of the Society of Photo-Optic instrumentation Engineers Conference on Robot Vision. Vol 336. (Arlington Va., May 6-7). SPIE, Bellingham, Wah.

8. **Goad, C.** (1983), *"Special Purpose Automatic Programming for 3D Model-Based Vision"*. In Proceedings of the DARPA Image Understanding Workshop.

9. **Lowe, D. G.** (1985), *"Perceptual Organization and Visual Recognition"*. Kluwer Academic publishers.

10. **Grimson, W. E. L. & Lozano-Perez, T.** (1987), *"Localizing Overlapping Parts by Searching the Interpretation Tree"*. PAMI-9, No. 4, July.

11. **Lowe, D. G.** (1987), *"The Viewpoint Consistency Constraint"*. Int. J. of Computer Vision, Vol. 1.

12. **Samet, H.** (1984) "T*he Quadtree and Related Data Structures.*" Computing Surveys. Vol. 16, No. 2, June.

13. **Gargantini, I.** (1982), *"An effective way to represent Quadtrees"*. Communications of the ACM. Vol. 25, No. 12, December.

14. **Gasson, P C.** (1983), *"Geometry of Spatial Forms"*. Ellis Harwood Limited.

15. **Worrall, A.** (1987), *"Facet Based Model for Object identification"*. University of Reading, Intelligent Systems Group, AOI/TR/RDGU/870322.

16. **Rydz, A. E.** *"Feature extraction"* (1985). A paper presented at the IEE Colloquium on Vision Modelling. Digest No: 1985/52.

17. **Bodington, R., Sullivan, G. D. & Baker, K. D.** (1988) *"On an ATMS Approach to the Consistent Labelling of Image Features."* Submitted to AVC88.