

# The Roll Angle Consistency Constraint

A. D. Worrall, K. D. Baker & G. D. Sullivan

Intelligent Systems Group, Department of Computer Science,  
University of Reading, RG6 2AX, UK.

Anthony.Worrall@Reading.ac.uk

---

*A new method is presented for reasoning with constraints on the three orientation parameters of an object which arise from the identification of features discovered in a single image. The method uses a geometrical model of the object and a tetrahedral representation of view-direction. The final angle parameter, the roll angle, is computed for each view direction. This method is subject to errors from the data and the assumptions inherent in the use of view directions. These errors are taken into account by considering upper and lower bounds on the roll angle. A method is presented for using constraints on the orientation parameters to prune the search for a consistent set of labelled image lines and to guide an iterative inversion of the view perspective.*

---

One of the major problems with the recognition of a three dimensional object from a single view is that the geometry of the image of the object can change radically as the object rotates with respect to the camera. This means that it is impossible in general to specify simple geometric measures which are independent of the view. There are of course exceptions to this general rule where reliable view-independent features exist. For example, a spheroid can be distinguished from a polyhedron because no matter what the view, the former generates curves in the image and the latter straight lines.

But in the general case, the recognition of an object must proceed by the use of view-dependent measures of the object and this leads to a classical conundrum: the lack of truly view-independent measures means that some approximation for an object's orientation must be determined before we can apply strong geometrical methods to classify and recognise it.

It has been shown before that it is possible to invert the view perspective on the basis of a set of lines labelled with respect to a model [1-2]. Perspective inversion uses iterative techniques to solve a set of non-linear orientation constraints. They require a good initial position for the model or the iteration may fail [2]. Because the translations are linear the iterative solution of the view perspective is far more sensitive to errors in rotation than translation.

The viewpoint consistency constraint of Lowe [3] exploits the fact that any legitimate interpretation of image lines as particular model features must be consistent with the observation of the model from a single camera position. Our work has been concerned

with methods of representing these constraints in order to reason about legitimate views. We have previously reported the use of the view-tetrahedron to determine a set of possible view-directions which are consistent with multiple constraints, which we call a viewpatch [4].

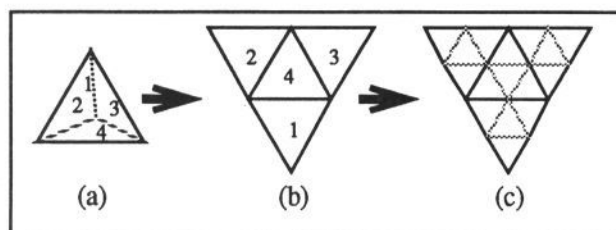
However, the determination of the view-direction constrains only two of the three rotational parameters. This leave the third parameter to be constrained: rotation of the camera about the view-direction, or roll. The roll angle consistency constraint captures the fact that in any legitimate view-direction the interpretation of image lines as particular model features must be consistent with a single roll angle.

We use the constraint for two purposes: (i) to help constrain the roll parameter itself and (ii) to eliminate areas of the viewpatch, where no consistent value for the roll parameter exists. This means that the roll angle constraint can be used to prune the search for a consistent sets of labelled lines and to limit the initial orientation used in the perspective inversion.

The main concern of this paper is to report a representation of the roll parameter, so that the roll angle consistency constraint can be enforced.

## View-Tetrahedron

There are many ways of characterising the view-direction. The method used here is based on a quadtree decomposition of the surface of a tetrahedron [4], see figure 1. The surface of the tetrahedron can be unfolded to lie on a flat plane by cutting along the three edges from one of the vertices. This gives an equilateral triangle divided into four smaller equilateral subordinate triangles which are the faces of the tetrahedron. Each of these faces can be further subdivided, and so on to any desired degree of resolution.



**Figure 1. The view tetrahedron,**  
(a) *The tetrahedron with face 1 at the front,*  
(b) *The tetrahedron unfolded,*  
(c) *A quadtree decomposition to depth of 2.*

This decomposition of the tetrahedron can be represented as a quadtree with a branching ratio of four.

A set of view-directions can now be represented as the union of triangles on the surface of the tetrahedron, which we call a viewpatch. This viewpatch can be made arbitrarily dense by increasing the depth of the quadtree decomposition. It is also very efficient to represent and manipulate in a computer.

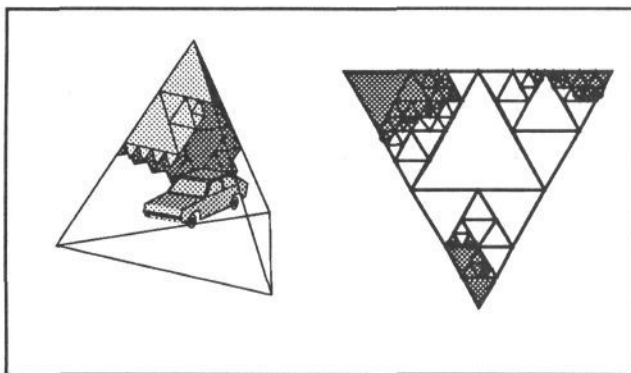
## Viewpatch Representation

This representation of a viewpatch can be used to encode constraints which depend upon the particular view of an object. The object is placed in the centre of the view-tetrahedron which is effectively at infinity. For each point on the view-tetrahedron the view-dependent property of the model can be computed and stored in a quadtree. For example, the visibility of a given facet on the model can be stored as a binary value in the quadtree. If all the subordinate triangles of a given triangle have the same value then that value need only be stored in the superordinate triangle. This reduces the storage size and the cost of computation on the viewpatch.

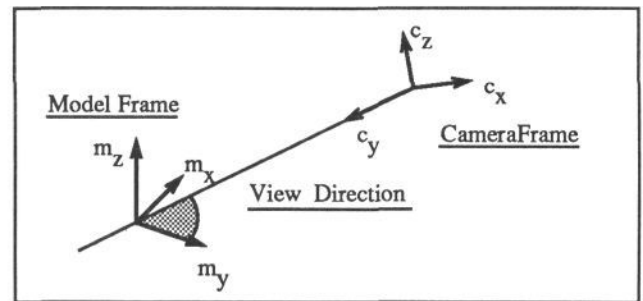
Quadtrees can be easily and quickly combined due to their recursive structure. This is very useful for model-based vision using a facet model, where the visibility of model lines is given by a logical combination of the visibility of the model facets. The viewpatch from which the two facets containing the offside windows and the rear window are both visible is shown in figure 2. The viewpatch is shown on the view-tetrahedron, with the car model inside. The viewpatch was expanded to a depth of five.

## The Roll Angle

If we consider the object fixed in the centre of the tetrahedron then a point on the view-tetrahedron represents the camera placed at an arbitrary distance along the view direction pointing towards the centre of the tetrahedron. Before the roll angle consistency constraint can be applied, it is necessary to determine a zero point for the roll angle. This is done by establishing a transformation from the model frame to the camera frame. The choice of the transformation is



**Figure 2.** A car model inside the view-tetrahedron with the viewpatch from which the offside windows and the rear window area visible. The car and the view-tetrahedron are not drawn to scale.



**Figure 3.** The transformation from the model coordinates to the camera coordinates by a rotation in the plane defined by  $m_y$  and the view direction.

arbitrary and need only be reproducible. In fact in our implementation the transformation for each view direction is independent of the transformations for the adjacent view-directions. The prescription we have used to define the zero roll transformation is as follows.

The model y axis ( $m_y$ ) is rotated in the plane defined by the  $m_y$  and the view direction such that it points along the view direction, see figure 3. The resulting frame is then translated along the view direction by a fixed distance. This transformation can be pre-computed for each view-direction and stored in a quadtree.

Consider a line in the image that has been labelled as corresponding to a particular line on the model. The orientation of the line in the image is known. For a given view-direction the local coordinate system allows the projection of the model line onto the image. The difference between the orientations of these two lines is the angle of rotation of the camera about the view direction. This angle is the roll angle. If only a single line is considered it is impossible to distinguish between a rotation by an angle and the angle plus  $180^\circ$ . Thus for a single line we have two possible roll angles.

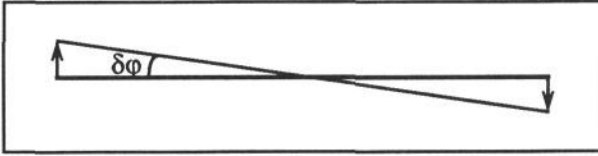
The orientation of the projected model line is independent of the image data and so it can be pre-computed. This allows a strategy similar to Goad's to be adopted, where for each view-direction the visible model lines and their orientations in the local coordinate frame are pre-computed and stored [5]. Thus the roll angle is calculated simply from the difference of the measured image line angle and the pre-computed projected model line angle.

## Coping with Roll Angle Errors

There are three main sources of error in the determination of the roll angle. The first is the uncertainty in the orientation of an image line. The second is in the assumption that the object of interest is centred on the camera axis at a fixed distance from the camera. The third arises from the fact that the density of samples of view-direction places a limit on the accuracy to which the roll angle can be determined.

Because of these sources of error the roll angles are not represented as a single value but as upper and lower bounds on the roll angle. Thus a labelled image line gives rise to two intervals of possible roll angles.

This representation allows the enforcement of the roll



**Figure 4.** The error in the roll angle  $\delta\phi$  produced by displacing the endpoints of an image line by one pixel perpendicular to its orientation.

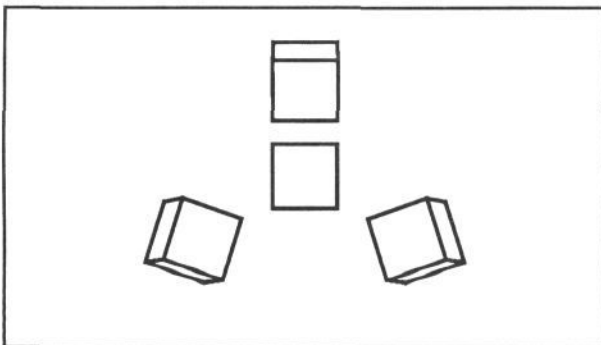
angle consistency constraint between multiple lines by taking the intersection of the intervals of the roll angles for each individual line. If there is no intersection then this view-direction is inconsistent and need not be considered any further.

### Estimating the Errors in the Roll Angle

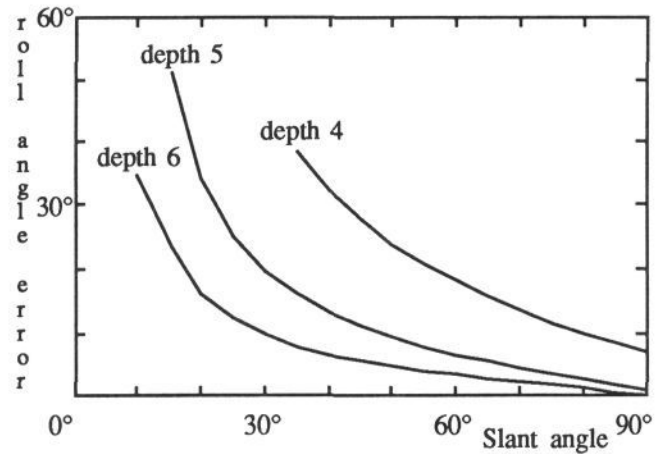
The errors in the roll angle from each of the three major sources needs to be estimated in order to calculate the range of roll angles to be allowed. In the case of the image line there are two obvious ways of estimating the error. If the edge detector used to find the line estimates the orientation of each edgel then the standard deviation of all the edgel orientations can be used to determine the error. This is adequate for long lines but if the line is short then there may not be enough data to determine the standard deviation accurately. In this case the change in orientation produced by displacing the ends of the lines by one pixel in a direction perpendicular to the line can be used to estimate the error in the error, see figure 4.

The angle of the projected model line is calculated in the centre of the triangle for each viewpoint. The allowed error in the roll angle must be large enough to cope with the fact the real viewpoint might not be at the center of the triangle. The consideration of how the roll angle changes as view direction changes is complicated by the fact that change in orientation of the projected line depends upon the angle between the line and the view-direction, the slant angle, as well as the direction and magnitude of the change in the view direction. Figure 5 shows how the view of a cube changes between adjacent viewpoints in a viewpatch of depth 4.

The magnitude of the change in the view-direction is greatest for the viewpoints at the center of the tetrahedron faces. This is because they are the closest to



**Figure 5.** Four view of a cube from adjacent view-point in a viewpatch of depth 4.

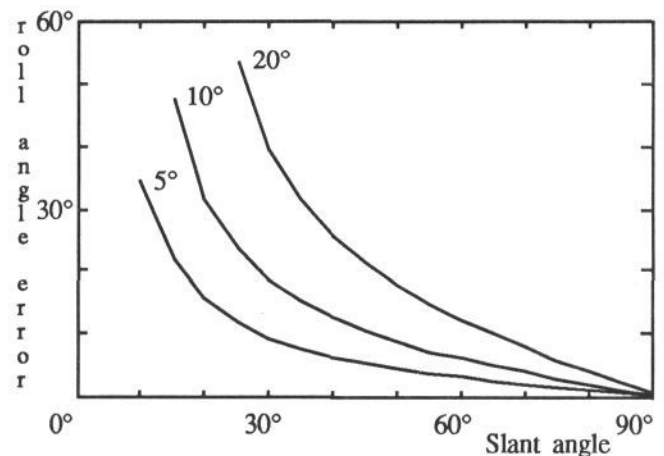


**Figure 6.** The maximum error in the roll angle as a function of the slant angle for three viewpatches with deferent depths.

the centre and so the angle between adjacent view-direction is greatest.

If we consider the viewpoint at the centre of one of the faces then it is possible to determine the maximum the error in the roll angle for a line as a function of the slant angle. The maximum roll angle error is shown in figure 6. As would be expected the error goes to  $\pm 90^\circ$  as the line becomes parallel to the view-direction where the roll angle is ill-defined. For any slant angle smaller then the data shown in figure 6 the roll angle is not constrained and so the error is  $\pm 90^\circ$

Displacing the model from the camera axis also causes a roll angle error for a given view-direction. The maximum roll angle error for displacement which subtend angles of  $5^\circ$ ,  $10^\circ$  and  $20^\circ$  at the camera are shown in figure 7. However, these errors can be compensated for by moving to a view-direction which is parallel to the line from the camera to model. This means that the roll angle viewpatch for a model that is not on the camera axis will be slightly displaced. The method of perspective inversion which uses the roll angle viewpatch is capable of correcting these errors.



**Figure 7.** The maximum error in the roll angle as a function of the slant angle when the displacement of the model from camera axis subtend angles of  $5^\circ$ ,  $10^\circ$  and  $20^\circ$  at the camera.



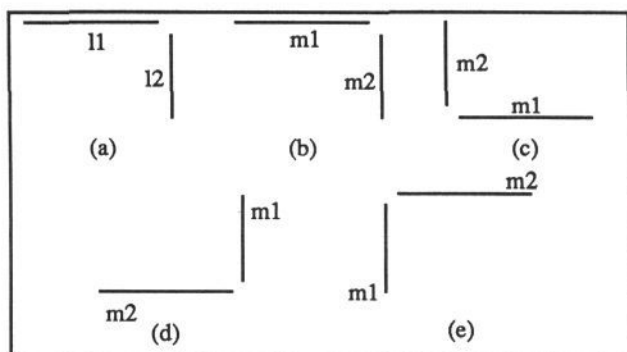


Figure 8. (b) to (e) show four consistent roll angles for the image line l1 and l2 shown in (a) when labelled as model lines m1 and m2.

For a given view-direction a labelled line has a roll error associated with it that is the combination of the uncertainty in the image line orientation and error dependent of the angle between the model line and the view direction. This error dependent on the slant of the model line can simply be pre-computed in tables which can be looked up for the appropriate depth of viewpatch being used.

### Topographical Constraint

The constraint on the roll angle only requires that the projected model lines are parallel to the labelled image lines. This does not always discriminate between two distinct ranges of roll angle. An example of this is shown in figure 8. In all four cases the projection of the model lines m1 and m2 are parallel to the image lines l1 and l2 respectively. These cases are different because of the spatial relationship between the projected model lines. This provides a method of distinguishing between the four cases. On the image lines we determine the angle of the line from the centre of l1 to l2 and similarly for the line from the centre of m1 the centre of m2 in cases (b) to (e). For the roll angle to be consistent the angle of the line between m1 and m2 must be broadly the same as for the line between l1 and l2, see figure 9. The allowed error in the line angles is  $\pm 45^\circ$ . This is large enough to cope with errors due to estimating the centre of image lines and small enough to reject roll angle that are out by almost  $180^\circ$ .

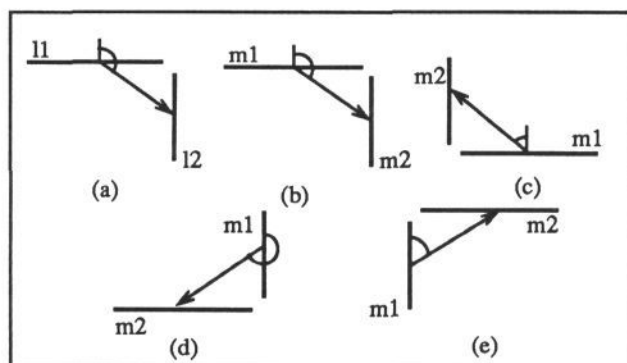


Figure 9. The angle of the line between (a) the image lines l1 and l2 and the model lines m1 and m2 in cases (b) - (e).



Figure 10. One of the image used in testing CARRS.

This is a topographical constraint which may be applied after the roll angles have been computed. Then the angle of the line connecting each pair of image lines is compared with the similar line for the model projected on to the image using the median value from each of the ranges of roll angles. Only those ranges where the angles agree are accepted. If the centre of two image lines are close together then the constraint is not applied to this pair as the angle of the connecting line cannot be determined accurately.

### Results

The roll angle consistency constraint together with the topographical constraint provide a powerful means to enforce consistency between image lines labelled as model lines. These constraints have been used for this purpose in an ATMS-based labelling system called CARRS developed by Bodington [6]. A Canny edge finder is applied to an image and the resulting edgels grouped into lines. These lines are then analyzed for characteristic shapes, for example "U" and "S" shapes, called cues. A typical image is shown in figure 10. These cues constitute the input for CARRS.

CARRS first searches for the maximal set of cues that can be labelled as features on the three dimensional model which accord with a set of two dimensional



Figure 11. The cues obtained from image with one of the maximal sets found by CARRS highlighted.

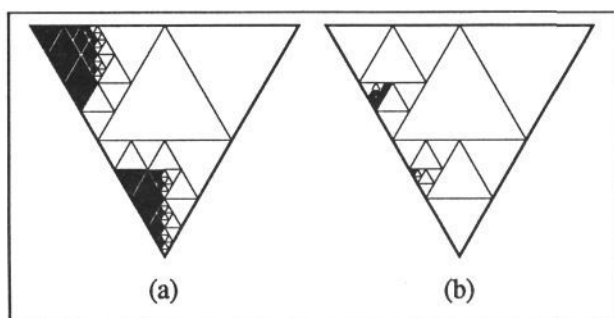


Figure 12. The visibility viewpatch for the correct labelling of RU2.23 (a) and the viewpatch after the roll constraint has been applied (b).

constraints. An example of such a maximal set is shown in figure 11 with the other cues used by CARRS. Each maximal set of labelled cues, then provides an hypothesis for the existence of a car. The hypotheses are then verified by labelling every line in the set of cues with respect to three dimensional model and applying the three dimensional constraints. The first constraint is the visibility constraint. That is, there must exist a viewpatch where all the labelled model lines are visible.

The roll angle constraint is then applied for all the view-direction in the visibility viewpatch. The effect of the roll constraint on the visibility viewpatch for the maximal set highlighted in figure 11 is shown in figure 12. The topographical constraint is then applied to this roll viewpatch.

To illustrate the performance of the roll consistency constraint and the and topographical constraint CARRS has been used on five test images, one of which

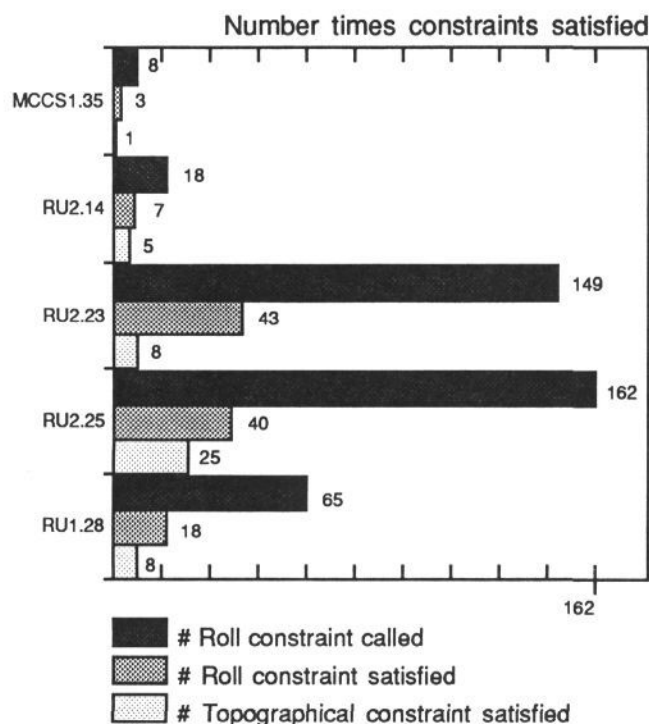


Figure 13. The number of times roll constraint and topographical constraints were used and satisfied for each image. Note the topographical constraint is used for each case which satisfied the roll constraint.

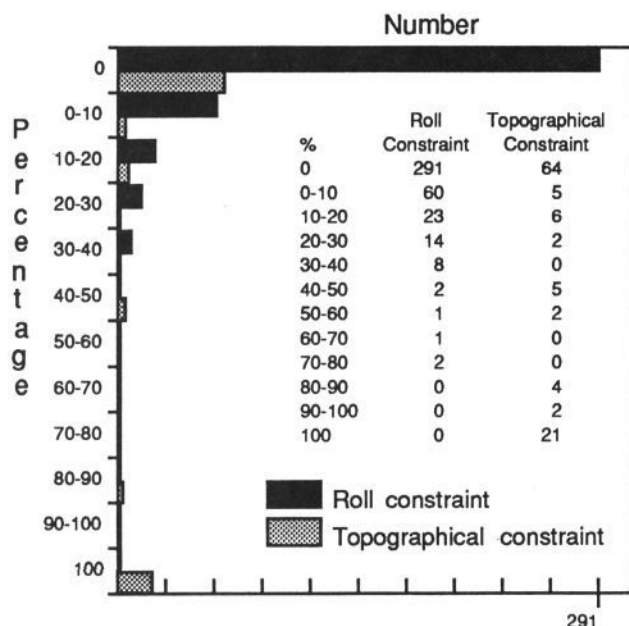


Figure 14. The number of times a viewpatch was reduced to the given percentage for the roll constraint and for the topological constraint.

"RU2.23" is shown in figure 10. The number of times that each of the constraints were called and satisfied are given in figure 13.

In almost three quarters of the cases that the roll consistency constraint was used it was not satisfied and so the labelling was rejected.

The topographical constraint was applied to the remaining quarter of the labellings and in over half of those cases the labelling was rejected.

The size of the resulting viewpatch as a percentage of the viewpatch before the constraint was used is shown in figure 14 for all the test images.

In most of the cases that were not rejected by the roll consistency constraint (i.e. those not reduced to a size of 0%) the size of the viewpatch was less than 30% of the size of the original viewpatch

In about half of the cases where the topographical constraint was satisfied, the size of the view patch did not change. However, each point in the resulting viewpatches was reduced to a single consistent roll angle.

Only 64 of original 402 hypotheses satisfied roll angle consistency constraint and the topographical constraint. These hypotheses were then used to invert the view perspective using an iterative technique [2]. The 13 hypotheses where the perspective inversion converged where then used for further evaluation. These 13 hypotheses contained all six of the foreground cars in the test images.

## Conclusion

A method has been presented for representing the view-dependent properties of an image of a know object under rotations. Given a set of image lines labelled with respect to the model we have shown how the representation can be used to impose the roll angle consistency constraint.

It has been shown how the use of upper and lower bounds on the roll angle can allow for the uncertainty in the determination of the orientation of image lines, imprecise assumptions in the use of a view-directions and the errors due to finite sampling of view-directions.

The use of the roll angle and the topological constraints has been used to prune the search for a consistent set of labelled image lines. When a such set of labelled image lines is found then the result of the constraint on the view-direction and roll angle is used in initial conditions for an iterative technique to invert the view perspective.

## Acknowledgements

The authors wish to thank Rob Bodington for providing data on the use of the roll angle constraint and the topographical constraint in the line labelling of image features and his help in the presentation of the results.

## REFERENCES

1. Lowe, D. G. "Solving For the Parameters of Object Models" *Proc. ARPA Image Understanding Workshop* (1980) pp 121-127.
2. Worrall, A.D., Baker, K. D. & Sullivan, G. D. "Model based perspective inversion" *AVC-88 Manchester* (1988) pp 13-18 and *Image and Vision Computing* Vol. 7 No. 1 (1989) pp 17-23.
3. Lowe, D. G. "The Viewpoint Consistency Constraint" *International Journal of Computer Vision* Vol. 1 (1987) pp 57-72.
4. Rydz, A. E., Sullivan, G. D. & Baker, K. D. "Model-Based Vision using a Planar Representation of the Viewsphere" *AVC-88 Manchester* (1988) pp 1-6.
5. Goad, C. "Special Purpose Automatic programming for 3D Model Based Vision" *International Journal of Computer Vision* (1983) pp 94-104
6. Bodington, R., Sullivan, G. D. & Baker, K. D. "Consistent labelling of image features using an assumption-based truth maintenance system" *AVC-88 Manchester* (1988) pp 7-12 and *Image and Vision Computing* Vol. 7 No. 1 (1989) pp 43-49.