Model Based Perspective Inversion

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

The problem of finding the spatial correspondence between an object and the image of the object under perspective projection is investigated and a new technique is demonstrated. The technique used is based upon a geometrical description, or model, of the object and a non-linear least squares solution of the resulting equations. An analysis of performance and a comparison with the previous work of Lowe is given. Three further areas of application in model-based vision are discussed.

The key problem with object recognition from a single image is that the perspective projection used in the formation of an image is singular. This means that it is impossible to reconstruct the position and orientation of an object using only the information contained in the image of the object. Other information must be used, such as the spatial characteristics of the object itself in terms of a geometrical description or model. The knowledge about the fixed interrelation between lines on the object, and their correspondence to image features, allows the construction of a set of simultaneous non-linear equations in terms of parameters of the object position.

Once a labelling has been established between image features and model features the solution of the resulting simultaneous equations can proceed by use of an iterative technique. At each iteration the equations are linearised by an expansion about a base position. The solution of these linear equations derived by matrix inversion is then used as the base position for the next iteration.

The use of an iterative technique requires some initial estimate of the position of the object. Fortunately the method is not overly sensitive to the initial position, which can usually be obtained merely by consideration of the visibility of the features being matched. The method also allows the inclusion of linear, or algebraic non-linear, constraints. It is possible to constrain the position of the object, for example, to lie in a given plane or allow rotation only about a fixed axis.

The iterative approach to perspective inversion has previously been studied by Lowe[1,2]. We propose here an alternative approach which avoids a non-linearity in depth and leads to simpler constraints; we call this the interpretation plane[3] method. The behaviour of the method is tested using a Monte Carlo simulation on a simple model. Results for both the interpretation plane method and the method used by Lowe are given and compared.

METHODOLOGY

The problem in model based vision is how to take image features provided by "low level operators" and use them to determine the position (and orientation) of a three dimensional model. To do this it is necessary to determine a correspondence between the features found in the image and features on the model. For example "Image line 27 is the line between the off side and the roof in the model". The problem of how to determine this correspondence is discussed else where.[4,5] Here we will assume that such feature correspondence hypotheses have been made.

The data that is available from the image consists of edges, which may be mapped onto lines in the model. Straight edges come from a sequence of edgelets which have been grouped. This means that the end points of the edges are not very stable. We therefore use only the analytic form of the line.

The association of image lines with model labels establishes constraints on the model position. We wish to find a position for the model which satisfies these constraints.

If we consider a rigid model then we have six parameters, three translation and three rotation. This requires at least six constraining equations. The constraints contain a number of separate non-linearities which must be solved in order to determine the spatial position of the model.

As the model is rotated some features become occluded by the rest of the model. This presents the problem of what should occur when model features that are being used in the perspective inversion become occluded. Fortunately we need only consider small changes in orientation which will not cross the boundary between topologically distinct views of the model. For this reason it possible to use a wire frame model and exclude visibility considerations.

The perspective projection of the model into the image is non-linear in depth. For this reason we project the image line out into three dimensions to form interpretation planes. Therefore constraints on the model position can be expressed as lines in the model lying in their interpretation planes.

AVC 1988 doi:10.5244/C.2.3

The rotational component of the model transformation introduces non-linearities in the constraints. It is not possible to avoid these non-linearities. Thus it is necessary to use an iterative approach.

The method then breaks down into a sequence of steps

1. Make a wire frame from the labeled lines.
2. Project the labeled image features out into three dimensions to form interpretation planes.
3. Set up a system of linear constraints based on local transformation parameters and an initial model position.
4. Find the values of the parameters that satisfy these constraints.
5. Generate a new model position.
6. If the new model position explains the image features with "sufficient accuracy" then stop, otherwise, go to step 3

The Interpretation Plane

The camera frame is defined as a right handed system with the origin at the nodal point of the camera. The y-axis is used as the depth. The image is formed on a screen parallel to the x-z plane at distance f, the focal length, from the nodal point, see figure 1.

We can now set up a frame in the image plane (u,v) with the origin at the intercept of the camera axis and the plane, and the u,v axis in the plane and parallel to the x and z axes respectively.

The perspective projection that takes a point $\underline{c}$ in the camera frame into the image plane is given by

$$u\,c_y = f\,c_x \quad \text{and} \quad v\,c_y = f\,c_z \tag{1}$$

From a line in the image we can construct a plane that passes though the origin of the camera frame and contains the image line. This defines the interpretation plane. This plane contains all possible lines in the three dimensional space which could gives rise to the image line through a perspective transformation. If the equation of the line in the image is

$$\alpha u + \beta + \gamma v = 0 \tag{2}$$

then the interpretation plane is given by

$$A\,c_x + B\,c_y + C\,c_z = 0 \tag{3}$$

where

$$A = \alpha f,\ B = \beta,\ C = \gamma f \tag{4}$$

The Linear Approximation

The transformation between the model and camera frames can split into two components a rotation represented by the matrix **R** and a translation given by the vector $\underline{t}$. The rotation aligns the two frames so that the axes are parallel and the translation moves the origins so that they coincide.

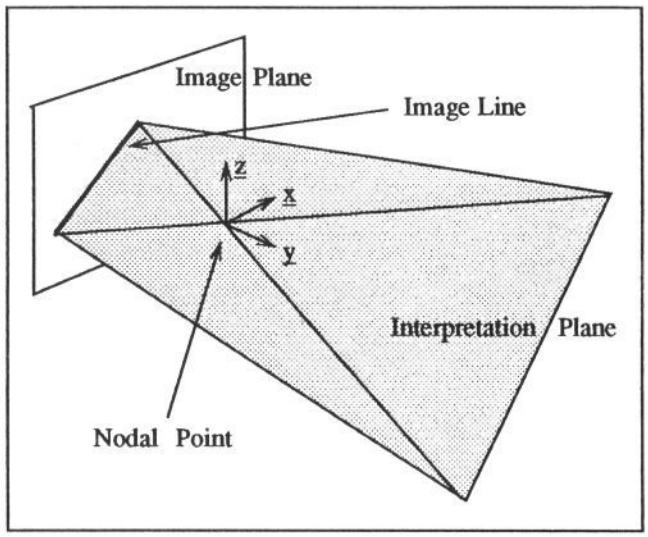

Figure 1. The Camera frame with an interpretation plane projected from an image line

Thus a point in model frame $\underline{m}$ is transformed into the equivalent point $\underline{c}$ in the camera frame by the equation

$$\underline{c} = \mathbf{R}\,\underline{m} + \underline{t} \tag{5}$$

The rotation and translation each depend upon three parameters giving a total of six independent parameters needed to specify the transformation.

The solution of the problem depends upon the choice of these parameters. The translation vector $\underline{t}$ is just the position of the origin of the model frame in the camera frame and so is linearly dependent on the cartesian coordinates (x,y,z) and does not present any serious problem.

The parameterisation of the rotation matrix **R** is fundamentally non-linear. In order to get round this problem we will parameterise not the rotation matrix itself but the change in the rotation matrix. This is sufficient since in the non-linear least squares fit approach we have to expand about an initial, or base, position.

Let us consider a base transformation defined by

$$\underline{c}_0 = \mathbf{R}_0\,\underline{m} + \underline{t}_0 \tag{6}$$

then we can parameterise the space of transformations by

$$\underline{c} = \mathbf{R}_z(\varphi_z)\,\mathbf{R}_y(\varphi_y)\,\mathbf{R}_x(\varphi_x)\,\mathbf{R}_0\,\underline{m} + \underline{t} + \underline{t}_0 \tag{7}$$

where the $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$ are the rotation matrices about the appropriate axes and $\underline{t}$ = (x,y,z) is the translation vector in the camera frame.

This parameterisation spans the space of transformations but is not orthogonal except in the limit where the φ's are infinitesimal. Since the transformation we require is assumed to be close to the base transformation we can make a linear approximation for the transformation space in the neighborhood of the base transformation

$$\underline{c} = \tilde{\mathbf{R}}(\varphi_x, \varphi_y, \varphi_z)\,\mathbf{R}_0\,\underline{m} + \underline{t}(x, y, z) + \underline{t}_0 \tag{8}$$

where

$$\tilde{\mathbf{R}} = \begin{pmatrix} 1 & -\varphi_z & \varphi_y \\ \varphi_z & 1 & -\varphi_x \\ -\varphi_y & \varphi_x & 1 \end{pmatrix} \quad (9)$$

Equation (8) can be written in component form as the set of equations

$$c_x = m'_x - \varphi_z m'_y + \varphi_y m'_z + x + x_0 \quad (10a)$$

$$c_y = \varphi_z m'_x + m'_y - \varphi_x m'_z + y + y_0 \quad (10b)$$

$$c_z = -\varphi_y m'_x + \varphi_z m'_y + m'_z + z + z_0 \quad (10c)$$

where we have introduced the rotated model vector

$$\underline{m}' = \mathbf{R}_0 \underline{m} \quad (11)$$

Linear Constraints

A point given by the vector $\underline{m}$ in the model frame is constrained to lie on the interpretation plane given by equation (3).

Therefore, by substituting the values of coordinates in the camera frame into the equation for the interpretation plane we obtain the equation

$$A(m'_x - \varphi_z m'_y + \varphi_y m'_z + x + x_0) +$$

$$B(\varphi_z m'_x + m'_y - \varphi_x m'_z + y + y_0) +$$

$$C(-\varphi_y m'_x + \varphi_z m'_y + m'_z + z + z_0) = 0 \quad (12)$$

If the coefficients of the interpretation plane A, B and C are normalised, then, for an arbitrary set of parameters, the left hand side of equation (12) is the perpendicular distance from the model point to the interpretation plane. Obviously when the point lies in the plane the distance is zero and the equation is satisfied.

We can rewrite equation (12) to isolate the parameters of the transformation to form the constraint equation

$$A x + B y + C z +$$

$$(C m'_y - B m'_z)\varphi_y +$$

$$(A m'_z - C m'_x)\varphi_y +$$

$$(B m'_x - A m'_y)\varphi_z =$$

$$-(A c_{0x} + B c_{0y} + C c_{0z}) \quad (13)$$

or in the more compact vector form

$$\underline{n} \cdot \underline{t} + (\mathbf{R}_0 \underline{m}) \wedge \underline{n} \cdot \underline{\varphi} = -\underline{n} \cdot \underline{c}_0 \quad (14)$$

where $\underline{n}$ is the vector perpendicular to the interpretation plane.

The right hand side of equatuion (13) is minus the perpendicular distance of the model point to the plane with the base transformation.

There are six parameters to be determined so at least six equations are needed. From each straight line in the model we can obtain two linearly independent equations by using two arbitrary points on the line. A convenient choice for these points are the end points of the model line, see figure 2. This means that at least three independent straight lines are required to solve for the six parameters.

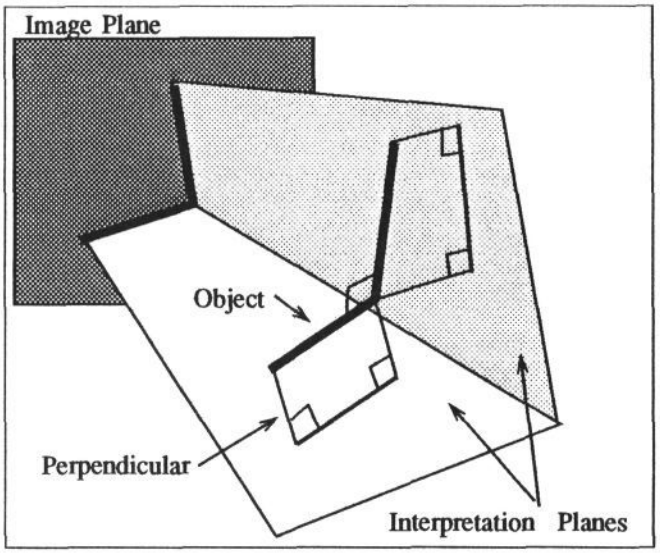

Figure 2. Interpreting two line in the image as coming from a right angles in tree dimensions.

If two lines being matched are parallel then they only contribute three independent equations and not the expected four equations. Thus a parallelogram found in the image only just constrains the model position. A line parallel to two other lines contributes no further information in the analytic case. However, it is of use in real images because of errors and for rejecting an incorrect labelling.

Model features other than straight lines can be used provided they are planar. For example, points identified on the model such as the center of a wheel. An image point gives rise to a interpretation line instead of a plane. It is more convenient to consider this line as the intersection of two orthogonal planes, in particular a vertical plane and a horizontal plane, see figure 3.

In the case of planar curves the image curve would be used to form an interpretation surface. At each cycle in the iteration it would be necessary to introduce a local linear approximation for the surface. This would seem to make extra effort unprofitable except in special circumstances. In general the number of constraint

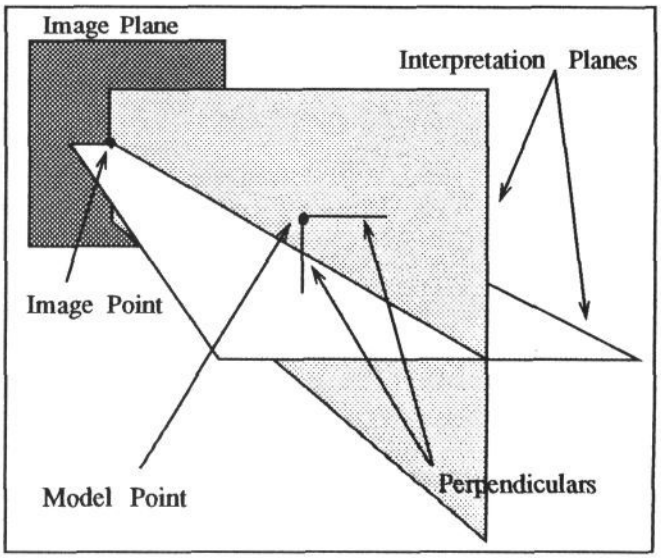

Figure 3. Constraining a model point to lie in two orthogonal interpretation plane.

equations arising from a given line is equal to the number of degrees of freedom of the line.

Least Square Solution

If we obtain a number of equations n, greater than five, then we can proceed by using the least squares method. Then resulting set of n linear constraint equations can be written in a matrix form as

$$\mathbf{A}\,\underline{\Theta} = \underline{e} \tag{15}$$

where **A** is an n by 6 matrix of coefficients with row vector,

$$A, B, C, Cm'_y - Bm'_z, Am'_z - Cm'_x, Bm'_x - Am'_y \tag{16}$$

$\underline{\Theta}$ is the vector of unknown parameters,

$$x, y, z, \varphi_x, \varphi_y, \varphi_z \tag{17}$$

and e is vector of minus the initial perpendicular distances to the interpretation planes.

By pre-multiplying by the transpose of A we can solve for $\underline{\Theta}$ since $\mathbf{A}^T\mathbf{A}$ is an invertible 6 by 6 matrix.

Hence, we find that

$$\underline{\Theta} = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\,\underline{e} \tag{18}$$

This assumes that equal weight is given to all the constraints. If this is not the case then this can allowed for by multiplying the constraint equation by a diagonal weighting matrix **W** so that

$$\mathbf{W}\,\mathbf{A}\,\Theta = \mathbf{W}\,\underline{e} \tag{19}$$

Which can then be solved for Θ to give

$$\underline{\Theta} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\,\underline{e} \tag{20}$$

Since we have made a linear approximation the values for the parameters are not exact and so it is necessary to repeat the process until the parameters are stable.

At each stage of the iteration it is necessary to compute the base transformation $\mathbf{R}_0$, $\underline{t}_0$. The new base rotation matrix is computed from

$$\mathbf{R}'_0 = \mathbf{R}_x(\varphi_x)\,\mathbf{R}_y(\varphi_y)\,\mathbf{R}_z(\varphi_z)\,\mathbf{R}_0 \tag{21}$$

and not the matrix $\tilde{\mathbf{R}}$ since it contains a skew component of the same order as the angles.

At each iteration the model lines are projected onto the image and the mean perpendicular distance of the end points to the image lines is computed. The iteration is assumed to have converged when this distance is less than a threshold of one pixel.

Including Other Constraints

The only dependence upon the camera frame is in the coefficients of the interpretation plane, equation (4). Thus we can easily combine constraint from other views of the object. We can also include other types of constraint. For example we can keep the solution in a fixed plane by using the constraint

$$ax + by + cz = d \tag{22}$$

Or suppress rotation about an arbitrary axis by aligning, say, the x-axis on to that axis. We can then impose the constraint

$$\varphi_x = 0 \tag{23}$$

Of course these constraints are not absolute and will only be satisfied to the same degree as any of the other constraints, but we can stress their importance by using the weight matrix. We can also include other non-linear constraints providing we can supply a local linear approximation.

MONTE-CARLO SIMULATIONS

The method outlined has been tested using a Monte Carlo simulation with a simple model of a cube of size two meters. The simulation consists of randomly selecting a position for the model and projecting the visible lines onto the image. These then form the labeled image features. A starting position for the iteration is selected by randomly transforming the correct position.

The test has considered the behaviour of the method with respect to rotations and translations, by randomly selecting an axis passing through the origin of the model and rotating by a random angle about this axis. The starting point was then offset by a random distance of up to 20 meters in an arbitrary direction.

The average number of iterations taken to converge to the solution as a function of the rotation angle and separation is given in figure 4. As might be expected the graph shows a symmetry about zero rotation. Also the number of iterations is independent of the separation.

Since this is an iterative technique convergence to the solution is not guaranteed. The percentage failure rate with respect to the rotation angle and separation is shown in figure 5. Failures come from a number of sources. The principle causes of failure are divergence of

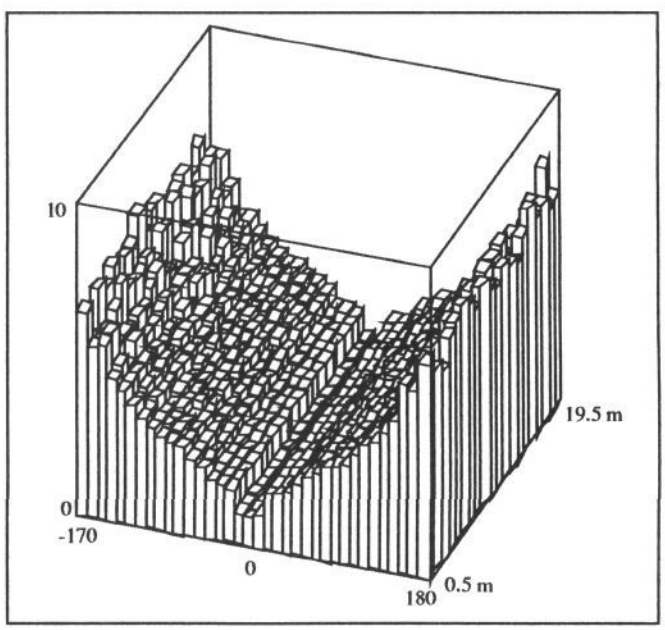

Figure 4. Number of iterations as a function of rotation angle in degrees and separation in meters.

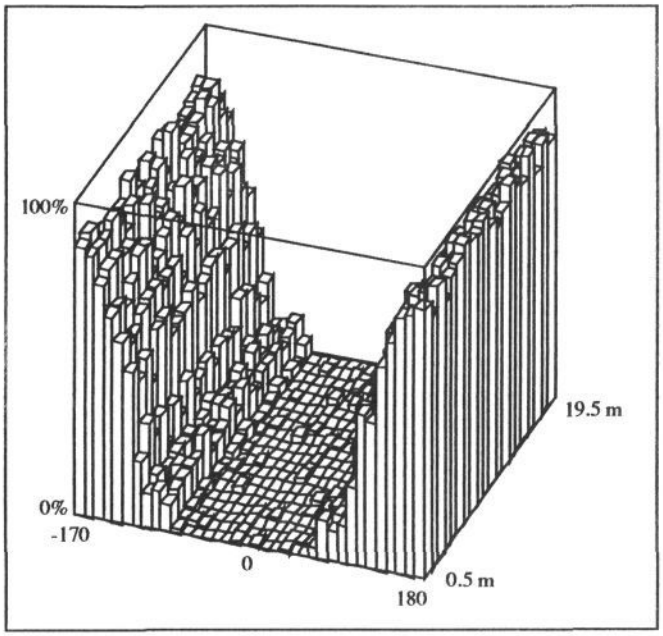

Figure 5. The percentage failure rate as a function of angle and separation.

the iteration and in the final position some of the matched features where not visible. The failures at small angles where all of the later type. These where due to one or more faces of the cube being only just visible in the starting position.

Lowe's method converged more slowly, particularly at large separations due to the non-linearity in depth of his equations. However, the failure rate was lower. These results did not depend upon wither a perspective projection or Lowe's projection was used.

APPLICATIONS

Use on Real Images

Low level edge data can be processed to find significant grouping, or cues. An example of a cue is an "S shape" which can be generated by the combination of the bonnet, windscreen and roof of a car. Such a cue is shown in figure 6.

There are a number possible model labels that could give rise to such an "S shape". By using perspective inversion on all the labellings we can generate possible positions for the car. Only three possible positions where found.

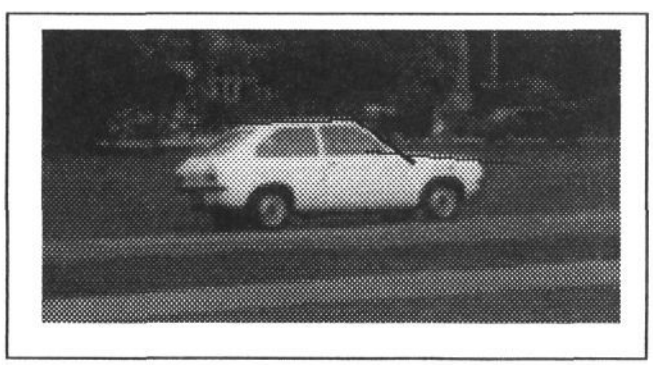

Figure 6. A cue in the form of an "S shape".

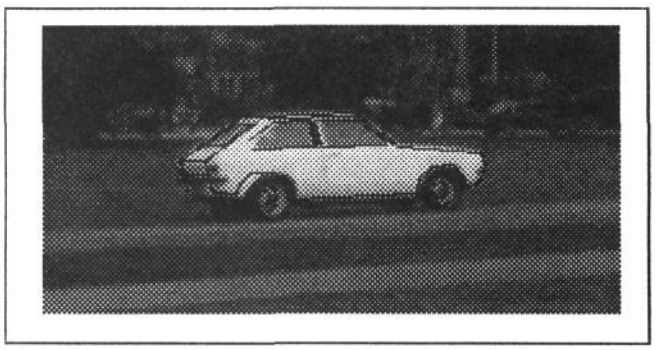

Figure 7. The model position after perspective inversion with one of the possible labeling for an "S shape".

The correct one is shown in figure 7. Only three image line and four model lines where used in the perspective inversion.

Three Dimensional Grouping

We can now use the position we have determined to enable grouping with other cues. This can be achieved by projecting the model onto the image and predicting the possible existence of other cues. For example, in the case shown above we would expect to be able to detect the windows of the car as two closed polygons[5]. This grouping on the basis of three dimensional information, or viewpoint consistency[6], is easier than in the knowledge free case. Using the now labeled window feature we can re-invert the perspective projection to find a position for the model which explains the windows as well as the "S shape". The resulting position is shown in figure 8. This process can be repeated until no further matches are found or it is decided that it is worth checking the hypothesis with an iconic evaluation.

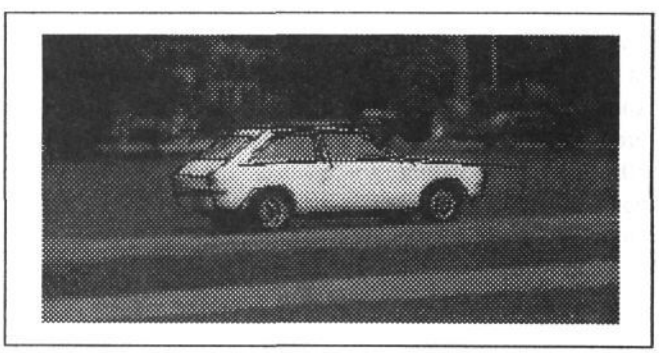

Figure 8. The model position using the Shape and off side windows.

Use In Iconic Search

In Iconic evaluation a position for the model is used to predict lines on the image. These lines are then checked in the raw image using an entirely predictive approach.[7,8] This results in a score and position for each image line. The position of the image line can then be used to re-invert the perspective projection to generate a new position for the model in a single iteration. The iconic evaluation can then be made again using the new position which can then be further refined. In the iconic evaluation the image line can only be detected if it lies

close to the predicted value and is reasonably parallel. Perspective inversion can only be used in iconic search for final refinement of the model position. This approach was used on the position shown in figure 7, and the result is shown in figure 9.

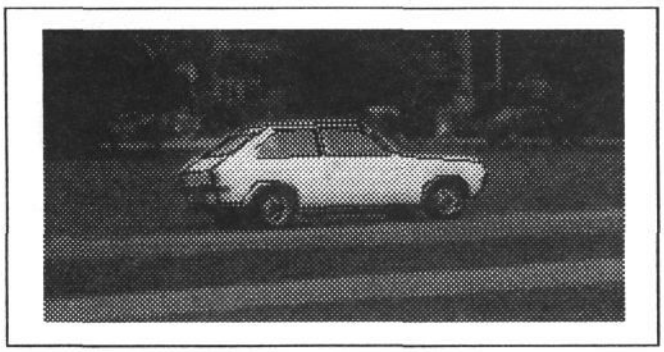

Figure 9. Model position after refinement using perspective inversion and iconic evaluation.

We can use the scores returned by the iconic evaluator to weight the constraints in the perspective inversion so as to maintain good matches and ignore poor, possibly spurious, matches. This technique is currently under further development.

Model Acquisition

One of the problems with using a geometric model approach to image understanding is that one needs to construct a geometrical model of the object. This means taking many measurements of the object and laboriously typing them into the computer. It is desirable for the geometry of the object to be obtained from images stored on the computer.

If we have multiple views of a static object, then by equating points from two or more images, we can reconstruct the three dimensional positions of model points. This require that we know the relative position of the cameras involved. If we include some already know reference model then we can invert the perspective projection using the method outlined above, see figure 10. This can be done for each image to determine the position of the camera with respect to the reference model. Armed with this information we can now proceed with the acquisition of model points.

CONCLUSION

This method allows the inversion of perspective independently of the starting translation and over a wide range of rotations. The main advantages over the method used by Lowe is that the constraints are expressed in three dimensional space. This means that it is easy to change the camera model and include constraints from other images or knowledge sources.

Further work needs to be done on use of other convergence conditions. For example convergence could be tested on the size of the parameters relative to an error matrix. Use of other parametrisations should also be explored. For example the rotations could be parametrised as the sines of angles.

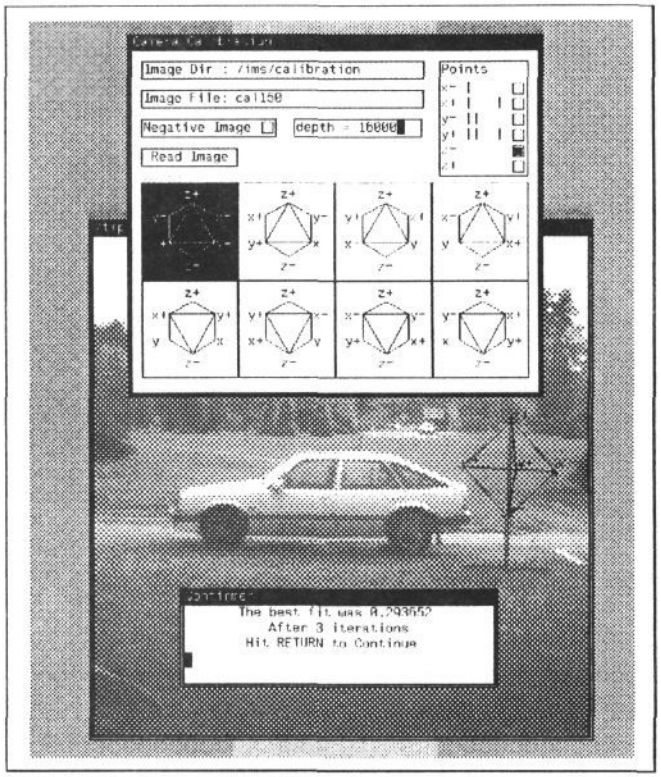

Figure 10. Example of perspective inversion being used to determine a camera position for model acquisition

REFERENCES

1. **Lowe, D. G.** "Solving For the Parameters of Object Models" *Proc. ARPA Image Understanding Workshop* (1980) pp 121-127.

2. **Lowe, D. G.** "Three-Dimensional Object Recognition from Single Two-Dimensional Images" *Artificial Intelligence* Vol. 31 (1987) pp 355-395.

3. **Horaud, R.** "New Methods for Matching 3-D Objects with Single Perspective Views" *PAMI* Vol. 9 No. 3 (1987) pp 401-412.

4. **Rydz, A. E., Sullivan, G. D. & Baker, K. D.** "Model-Based Vision using a Planar Representation of the Viewsphere" *AVC-88* Manchester 1988.

5. **Bodington, R. B., Sullivan, G. D. & Baker, K. D.** "The consistent Labelling of Image Features using an ATMS" *AVC-88* Manchester 1988.

6. **Lowe, D. G.** "The Viewpoint Constancy Constraint" *International Journal of Computer Vision* Vol. 1 (1987) pp 57-72.

7. **Brisdon, K.,** "Evaluation and Verification of Model Instances", *AVC-87* Cambridge 1987. pp 33-37

8. **Brisdon, K., Sullivan, G. D. & Baker, K, D.** "Feature Aggregation in Iconic Model Evaluation" *AVC-88* Manchester 1988.