# Extracting Projective Structure from Single Perspective Views of 3D Point Sets

**C.A. Rothwell, D.A. Forsyth, A. Zisserman and J.L. Mundy**
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ
England
email: charlie@uk.ac.ox.robots

## Abstract

*A number of recent papers have argued that invariants do not exist for three dimensional point sets in general position [3, 4, 13]. This has often been misinterpreted to mean that invariants cannot be computed for any three dimensional structure. This paper proves by example that although the general statement is true, invariants do exist for structured three dimensional point sets.*

*Projective invariants are derived for two classes of object: the first is for points that lie on the vertices of polyhedra, and the second for objects that are projectively equivalent to ones possessing a bilateral symmetry.*

*The motivations for computing such invariants are twofold: firstly they can be used for recognition; secondly they can be used to compute projective structure. Examples of invariants computed from real images are given.*

## 1 Introduction

Exploiting structure modulo a projectivity has recently been shown to simplify a number of vision tasks such as model based recognition [1, 7, 10, 11, 17, 18, 19, 20] and relative positioning using epipolar calibration [2, 6, 9, 12]. Projective *invariants* are functions of this structure unaffected by a projective transformation, and so have the same value on the actual object as on any projectively transformed version. One of the major advantages of this approach is that camera calibration is not required at any stage. Work has so far concentrated on planar objects, though recent papers have recovered projective invariants for 3D curved surfaces [8]. Furthermore, Faugeras and Hartley *et al.* [6, 9] have shown that from two views of a point set, structure can be recovered to within a 3D projective transformations of the actual Euclidean positions.

In this paper we describe two methods for obtaining such projective structure from a *single* perspective image of a 3D point set. In each case an assumption must be made on the geometry of the points, but this assumption can be verified by testing other points in the set. The two cases are:

1. **Polyhedral cage:** The points lie at the vertices of a (virtual) polyhedron: the object itself need not be polyhedral. For the polyhedral class explored here trihedral vertices are assumed. A minimum of 7 points are required in order to recover structure modulo a projectivity.
2. **Plane of symmetry:** The object has a bilateral symmetry. A minimum of eight points or four points and two lines are required. Other than the symmetry there is no other restriction on the points.

Projective invariants have already proved very useful in model based recognition by providing *indexing functions* that can be employed to gain rapid access to model libraries [1, 7, 10, 11, 17, 18, 19, 20]. A frequently used plane projective invariant is that of five coplanar lines. However, many polyhedra do not have this many lines lying in a single plane, and so a different type of invariant is required. In section 2 we show how constraints between planes can be used to form projective invariants for entire polyhedra, or in fact, for constrained sets of 3D points.

Burns, *et al.* [3] show that invariants can not be measured for 3D point sets in general position from a single view, that is for sets that contain absolutely no structure. A similar result has been obtained by [4, 13]. The theorem has frequently been misinterpreted to mean that *no* invariants can be formed for three dimensional objects from a single image; the work in

this paper is an existence proof that this is *not* the case.

The rest of the paper is organised as follows: section 2 describes the recovery of the projective structure of polyhedra and the application of this to *caging* point sets. Section 3 describes the recovery of projective structure for objects with a bilateral symmetry. In both cases structure is recovered from single perspective images, and this is demonstrated on images of real objects.

# 2 Polyhedral Cage

We first describe the method for recovering projective structure for polyhedra, and then apply this methodology to non-polyhedral objects.

The work on polyhedral shape recovery is an extension of the work of Sugihara [16], and the algebraic formulation given is related very closely to the one given in [16]. A more complete review of shape recovery for polyhedra is given in [15]

## 2.1 Solid Polyhedra

Following the usual practice we concentrate on *trihedral* vertices: these are points formed by the intersection of three world planes (that is planes in 3D Euclidean space). Trihedral junctions are *stable* because three planes generically meet in a single point; that is one can perturb any one of the planes and still form a single vertex close to the original vertex. This is not true for higher order junctions.

This assumption has little affect on the mathematics presented in section 2.2, the difference being that if a point lies on $n$ (rather than 3) planes we can form $n-1$ constraints rather than the 2 for the trihedral case. However, it is impossible to infer structure from the self-occluded part of the viewed polyhedron unless the vertices are trihedral.

A further assumption we make is that the polyhedra *represent* real solid objects, that is, a polyhedron is not just a polygon or a group of polygons embedded in 3-space.

## 2.2 Forming the Constraint Equations

In this section we describe the imaging process and show how projective information can be extracted from single views of isolated polyhedral objects. The input is a set of unknown planes that are bounded by an arbitrary number of edges ($\geq 3$), and a set of points whose image locations are known. Using the trihedral assumption, the points always lie on three given planes. Full details of the derivation are given in [15].

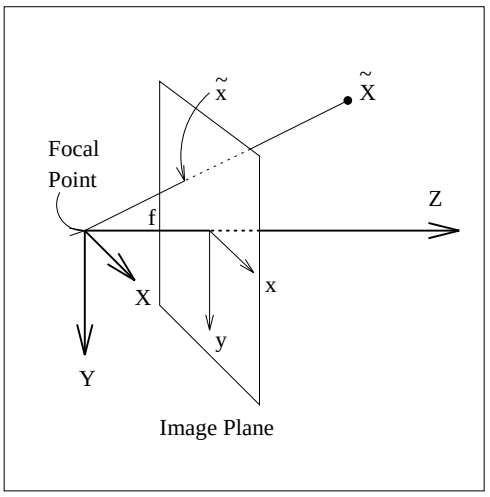



Figure 1: The camera model used is shown with the world and camera coordinate frames marked. We use a perspective imaging model to project the world point $\tilde{\mathrm{X}}$ to the camera plane point $\tilde{\mathrm{x}}$. Note that we can measure only image coordinates, rather than the camera coordinates of an image point (these two frames are linked by an affine map).

The coordinate frame used to develop the theory has the world $X$ and $Y$ axes in a plane parallel to the image plane with directions shown in figure 1 and centred at the focal point. The $Z$ axis lies along the principal axis of the camera. A polyhedron is made up of a set of planes[1]:

$$a_j X_i + b_j Y_i + c_j Z_i + 1 = 0, \qquad (1)$$

where $(X_i, Y_i, Z_i)$ is a point known on the plane, $\mathbf{v}_j = (a_j, b_j, c_j, 1)^T$ is the homogeneous representation of the plane in projective 3-space, and $j \in \{1, ..., n\}$ where $n$ is the number of planes on the object. Note that $n$ is the sum of the planes that we observe in an image and the inferred occluded planes. In the sequel we assume that the world points $(X_i, Y_i, Z_i)$ can all be observed uniquely in an image, and so must be formed by polyhedral vertices.

Under a pinhole projection model, projection onto the plane $Z = 1$ maps the point $(X_i, Y_i, Z_i)$ to $(x_i, y_i)$ where:

$$x_i = \frac{X_i}{Z_i} \quad \text{and} \quad y_i = \frac{Y_i}{Z_i}.$$

Substituting these into equation 1, dividing through

[1] Planes that pass through the focal point are not represented as these project to a single line in the image.

by $Z_i$ and setting $t_i = 1/Z_i$ gives:

$$a_j x_i + b_j y_i + c_j + t_i = 0. \quad (2)$$

Note that this equation is linear in the unknowns $\{a_j, b_j, c_j, t_j\}$, and is precisely of the form derived by Sugihara in [16]. In practice we are unable to measure $(x_i, y_i)$ because of the image plane translation and scaling of the image points (if we have an uncalibrated camera). We will however, be able to measure image points $(x'_i, y'_i) = (fx_i/\alpha + c_x, fy + c_y)$ with the camera focal length and aspect ratio being $f$ and $\alpha$ respectively, and the camera centre at $(c_x, c_y)$. Rewriting equation 2 gives:

$$\begin{aligned} \frac{a_j\alpha}{f}x'_i + \frac{b_j}{f}y'_i + c_j - \frac{\alpha c_x + c_y}{f} + t_i &= 0 \\ \Longrightarrow a'_j x'_i + b'_j y'_i + c'_j + t_i &= 0, \quad (3) \end{aligned}$$

with $\mathbf{v}'_j = (a'_j, b'_j, c'_j, 1)^T$ the assumed plane equation; $\mathbf{v}'_j$ is a projectivity of the actual plane equation, $\mathbf{v}'_j = \mathtt{C}\mathbf{v}_j$ ($\mathtt{C}$ is a $4 \times 4$ matrix). Equation 3 is used from now on. Note that we have assumed a very simple distortion on the image plane, ie. a scaling and a translation. As $\mathtt{C}$ represents a projectivity we may in reality apply any linear map to the image.

Using the trihedral assumption we know that each image point lies on three planes, say $j \in \{p, q, r\}$. We eliminate the $t_i$ term in equation 3 between pairs of equations to give two linear equations per observed image point. For the $m$ observed image vertices form a set of $2m$ equations (which may or may not be independent) in the $3n$ unknowns:

$$\mathtt{A}(\mathbf{x}', \mathbf{y}')\ \mathbf{w}' = 0, \quad (4)$$

where $\mathbf{x}' = (x'_1, \ldots, x'_m)^T$, $\mathbf{y}' = (y'_1, \ldots, y'_m)^T$, $\mathtt{A}(\mathbf{x}', \mathbf{y}')$ is a $2m \times 3n$ array of constraints, and $\mathbf{w}' = (a'_1, b'_1, c'_1, a'_2, \ldots, c'_n)^T$. The solution space of $\mathbf{w}'$ is the null space or kernel of the matrix $\mathtt{A}$, and represents the set of polyhedra that can be reconstructed from the image. The dimension of the kernel is $d \geq 3n - 2m$. The '$\geq$' is due to possible linear dependence and results from *position freeness* (as defined by Sugihara [16] and discussed in [15]).

The dimension of the kernel immediately tells us whether the line drawing can be constructed as a real polyhedron, ie. this process acts as a further filter in the interpretation of line drawings. We may consider the following two cases:

1. $\mathbf{d = 3}$ : The drawing can be interpreted only as a plane polygon (not a polyhedron).
2. $\mathbf{d \geq 4}$ : It represents a $d$ dimensional class of solid polyhedra.

It should be noted that the presence of image noise often increases the dimension of the kernel for non position free objects. The correcting of this effect is hard.

### 2.2.1 Forming the Invariants

In this section the invariants are defined for the three dimensional point sets of interest. Using the results of the previous section, the invariants are formed by reconstructing the caging polyhedra of the point sets in 3-space, and then measuring the projective invariants for the planes of the polyhedra.

In [15] it is shown that all of the non-degenerate solutions represented by the kernel of $\mathtt{A}$ lie in the same projective equivalence class. Therefore, they all have the same projective invariants, and as the actual realisation lies in the kernel they have the same invariants as the physical object. We may therefore choose any arbitrary solution and measure its projective invariants. The only restriction is that the 3D realisation of the chosen polyhedron is not planar.

Considering a simple 4 *dof.* figure such as a cube which has six planes, we can form three independent projective invariants. A sample three are:

$$I_1 = \frac{|\mathtt{I}_{3561}|.|\mathtt{I}_{3542}|}{|\mathtt{I}_{3564}|.|\mathtt{I}_{3512}|}, I_2 = \frac{|\mathtt{I}_{3562}|.|\mathtt{I}_{3142}|}{|\mathtt{I}_{3512}|.|\mathtt{I}_{3642}|}, I_3 = \frac{|\mathtt{I}_{3564}|.|\mathtt{I}_{5612}|}{|\mathtt{I}_{3561}|.|\mathtt{I}_{5642}|}, \quad (5)$$

where $\mathtt{I}_{abcd} = [\mathbf{v}'_a, \mathbf{v}'_b, \mathbf{v}'_c, \mathbf{v}'_d]$. The proof that these expressions are invariant under projective transformations is given in [15].

Euclidean actions on the world space result in projective transformations in the plane dual space [15]. As the above expressions are invariant to projective transformations, they will also be unchanged under the dual transformation, and hence also to Euclidean motions of the world object.

An interesting case to note is when the four planes $\{a, b, c, d\}$ forming $I_{abcd}$ become concurrent, such as a cycle of faces on a cuboid: in this case $|I_{abcd}| = 0$ and so the invariants may become zero or undefined. When noise is added to the invariant measures they become non-zero. It is impossible to quantify a 'near zero' value, as under a projectivity any determinant can be made arbitrarily large. For example, see the value of the first invariant in table 1 for the images in figure 2.

### 2.2.2 Examples of Caging

In the following examples *no use* is made of the polyhedral structure of the test objects. This avoids the intractable task of extracting labeled polyhedral line

| | $I_1$ | $I_2$ | $I_3$ |
|---|---|---|---|
| view 1 | -0.0141 | 0.9862 | -1.0586 |
| view 2 | -0.0356 | 0.9657 | -0.9770 |
| view 3 | -0.0189 | 0.9796 | -1.0446 |
| view 4 | -0.0400 | 0.9619 | -0.9247 |
| view 5 | -0.0012 | 0.9988 | -1.0043 |
| mean | -0.0219 | 0.9784 | -1.0018 |
| $\sigma$ | 0.0143 | 0.0135 | 0.0482 |
| $\sigma$ (%) | - | 1.4 | 4.8 |

Table 1: The invariants computed for the views in figure 2. The mean and standard deviations are shown in rows 7 and 8. In the last row the standard deviation is shown as a percentage of the mean for $I_2$ and $I_3$. The value is not computed for $I_1$ as it is ideally zero. From these values it is evident that the invariants remain constant over a change in viewpoint.

drawings from the image. Instead we just use observed vertex positions and group these points on virtual planes (which may or may not be physical object planes). This process is termed *polyhedral caging*. Caging 3D points mean that we can recognise a much larger class of objects than just polyhedra, in fact any object for which we can observe points that lie at the vertices of a polyhedral structure. Note that the constraint that there are a number of sets of four coplanar points is required, otherwise there is the situation of [3, 4, 13] where no invariants can be formed.

In figure 2 we show a series of five images of a punch with the same seven points marked on each image. These points are the images of vertices of the virtual polyhedron for which the invariants are computed.

The images are processed by fitting straight lines to edge data. Then vertex positions are found by intersecting pairs of lines (by hand) and the points are then grouped into planes to form a six sided polygon. The measured invariants for the five views are given in table 1. They are nearly constant over the change in viewpoint. In the table the standard deviations are given, as well as the value of the standard deviation as a percentage of the mean invariant value (except for the first invariant which is meant to be zero), and these are below 5%.

In figure 3 a second example is shown for two views of a calibration table. The invariants for the two views are given in table 2; they remain constant over the change in viewpoint, but are different from the invariants computed for the punch. This means that the invariants are both stable and give discrimination between different objects. Note that different invariants provide differing amounts of discrimination between objects; for example between the punch and the calibration table $I_3$ offers the best discrimination.

| | $I_1$ | $I_2$ | $I_3$ |
|---|---|---|---|
| view 1 | -0.0014609 | 0.990833 | -6.29954 |
| view 2 | 0.00117287 | 1.0074 | -6.4445 |

Table 2: The invariants measured for the two views of the calibration table in figure 3. Note that the invariants stay fairly constant even under image noise.

## 2.3 Verifying Polyhedral Assumption

So far we have computed only the projective structure for the point sets represented by the seven visible points in the images. However, the polyhedral assumption predicts the position of other points in the image (in this case the eighth point forming the six sided figure) which may or may not be visible. If it is visible, then the predicted and actual image position provides an independent check on the validity of the polyhedral assumption. The position of the remaining points can be determined either algebraically as here or geometrically as in [15].

### 2.3.1 Algebraic Approach

Predicting of the location of the eighth point is carried out by intersecting the three virtual planes that bound it. Note that we are using the equations of planes that are totally obscured.

Considering all of the solutions represented by the basis $\mathbf{w}'$, we find that the locus of the eighth point is a line in space passing through the centre of projection. If this line is projected into the image we recover a single image point which is the position of the hidden point. This point is therefore the same for all of the solutions of the caging polyhedra. The position of the point is shown in figure 5 for the examples given in figure 2. Instead of just showing the extra point we show the complete polyhedra that was used to cage the data points. The good agreement between where we would expect to find the eighth point (which is visible in some of the images) and the predicted position highlights the accuracy of the method. The position of the eighth point and the caging polyhedra for figure 3 are shown in figure 6.

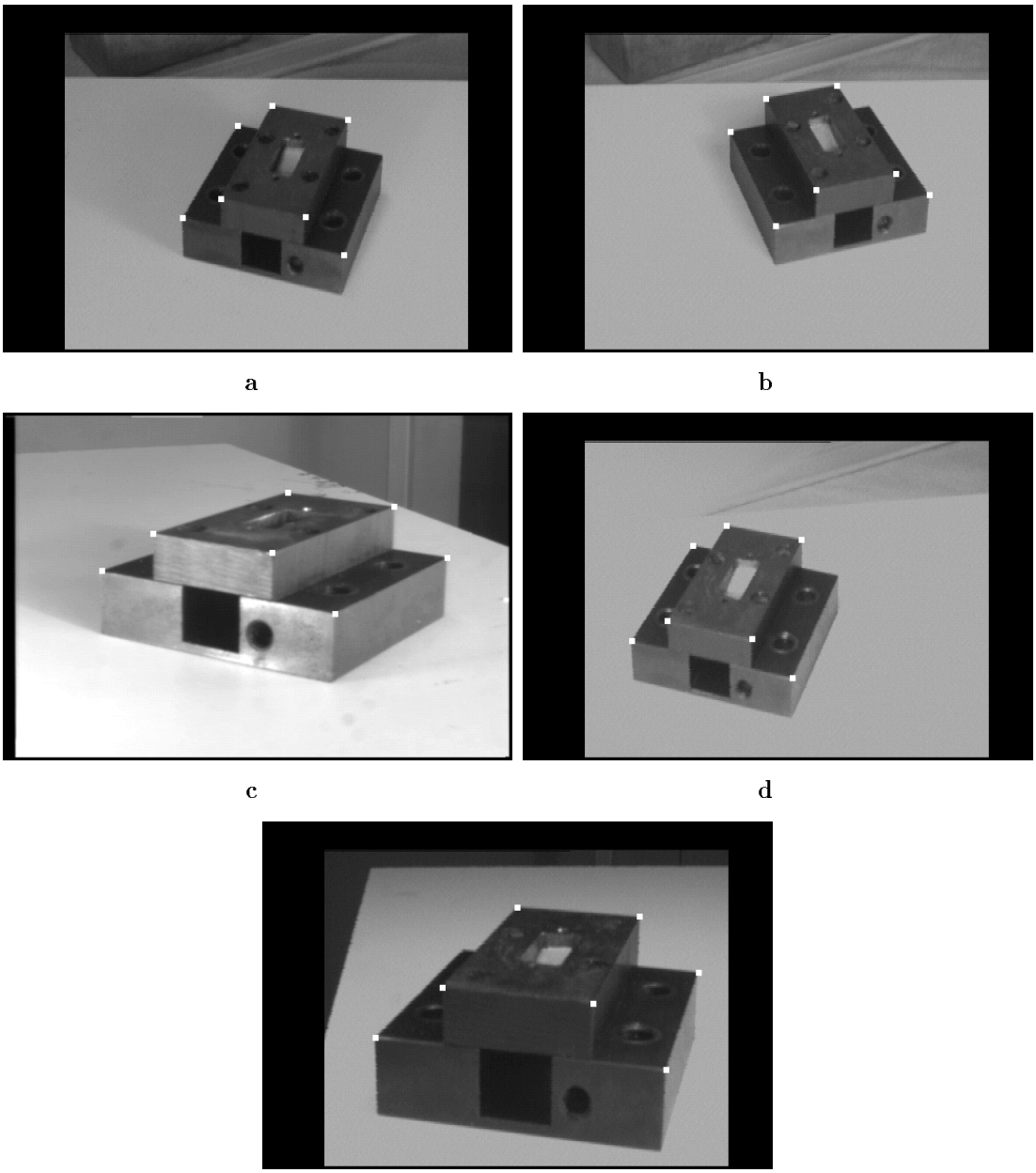
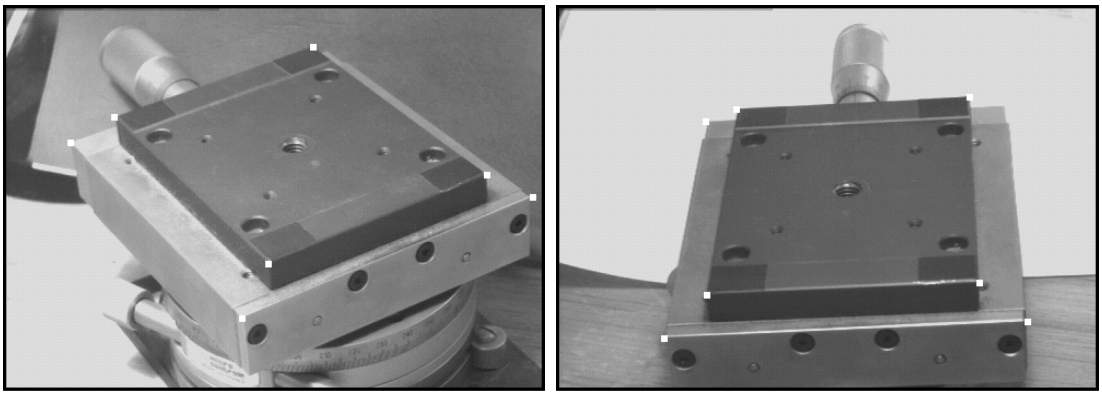



Figure 2: This sequence of images shows five views of a punch and the points used to compute the invariants. These points are caged by a polyhedron whose bounding planes are not actual object planes. We do not use the physical object planes to compute the invariants, for example by using one of the cuboids making up the object, because the invariants for a cuboid are always zero or infinite (as the planes form stars they are not independent). Making use of the points to construct a cage ensures that we can still measure invariants for the punch. In (c) and (e) the seventh point is occluded. Its position is located by symmetry. The computed invariants are given in table 1.



a b

Figure 3: Two views of the calibration table used to test the invariants are shown. The seven points used to compute the invariants are marked in white. In the right image the eighth point is also visible; this could be used to overconstrain the solution, but in this case we ignore it.

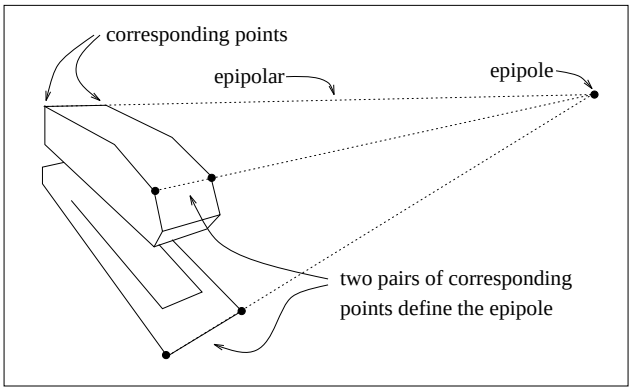



Figure 4: The *epipole* can be located using the intersection of lines between two corresponding points on an object possessing a mirror symmetry (the points are marked by filled in circles). Epipolars can then be constructed through the epipole to aid correspondence.

# 3 Objects with Bilateral symmetry

If an object has a plane of symmetry, invariants can be measured without restricting the points to lie on polyhedral vertices. Furthermore, any object projectively equivalent to one with a plane of symmetry will have the same properties.

A single perspective image of a symmetrical object is equivalent to two perspective views of half of the object: the two halves are projectively equivalent (a reflection is a specialised projection). Consequently, the single image is equivalent to two perspective images, with different optical centers, of the same (half) object. We can therefore measure projective invariants, using the two view methods developed by Faugeras [6] and Hartley, *et al.* [9] (using stereo reconstruction). This procedure requires eight points to be visible on each side. However, for point sets the mirror symmetry guarantees that a plane of points exist, thus we can exploit the two view method of [2, 12] which capitalises on this planar configuration and requires fewer points (only four corresponding points need be visible on each side, that is eight in total rather than 8 in each half-image).

Projective structure can be obtained by first solving for epipolar calibration between the "views". All that is required to do this is a pair of matching points on each side of the object. Drawing lines through the corresponding points give epipolars, the intersection of two such lines gives the location of the epipole. This is demonstrated in figure 4. Any line through the epipole is an epipolar that can be used to test for correspondence.

From the four points used to define the epipolar geometry projective invariants can be computed using either a single line lying out of the plane of the points and its mirror correspondence, or using two more pairs of points that similarly define two lines. The construction works by determining the intersection of the line with the plane, and computing the invariants for the four points and the intersection point. Details are given in [2, 12]. In figure 7 we shows two views of a stapler with the image features of interest superimposed in white. For each image we can compute the 3D invariants. As shown in table 3, the invariants remain fairly constant under change in viewpoint.

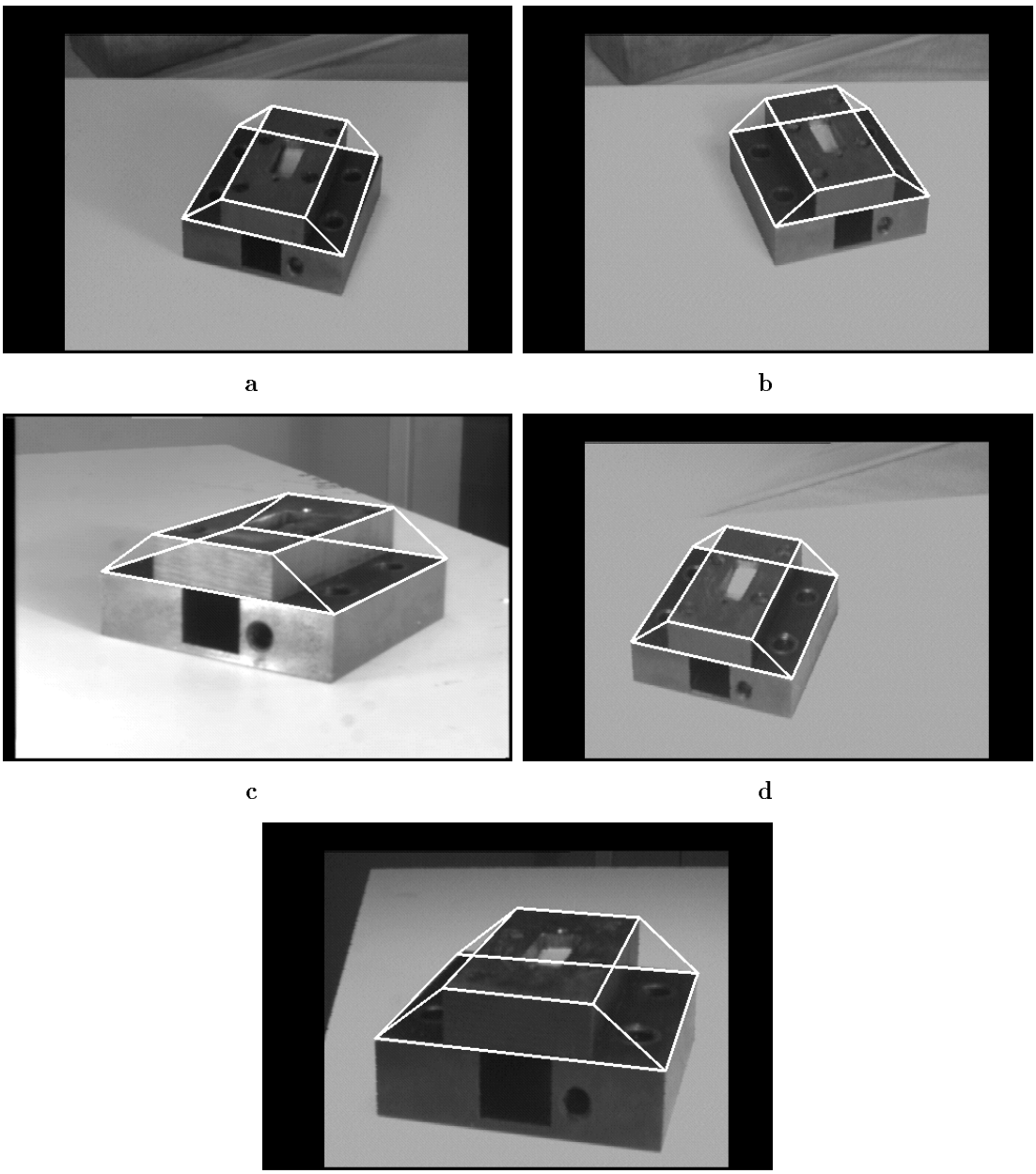
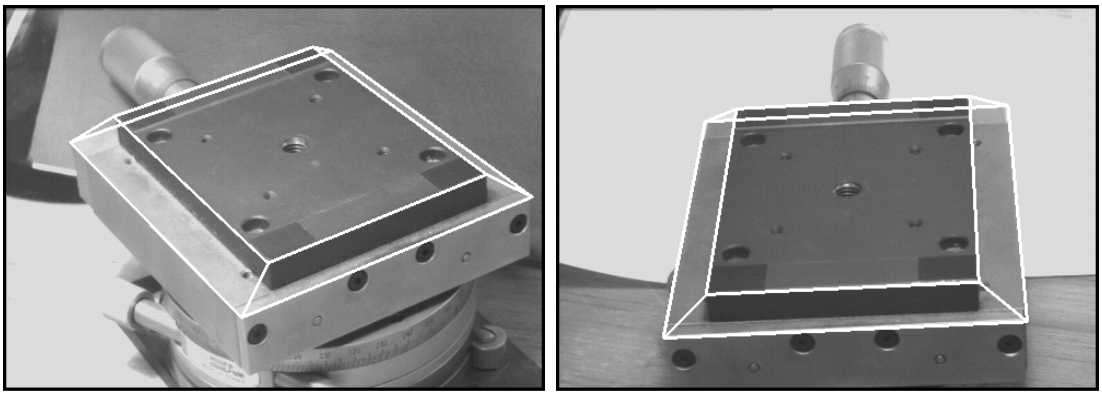
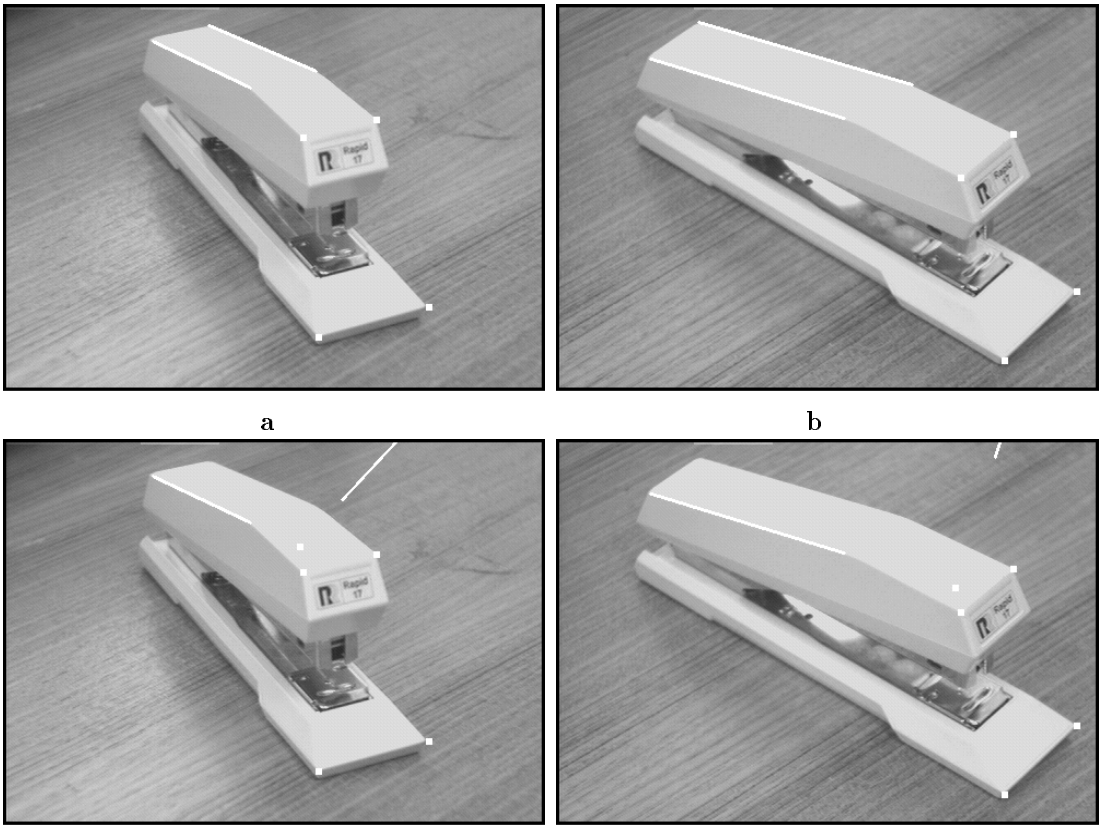



Figure 5: Here the eighth point for the caging polyhedra is shown. Note that we have at no stage measured the position of the eighth point in the image, but have computed it from the other seven points.



**a** **b**

Figure 6: The caging polyhedra for the examples in figure 3.



**a** **b**

**c** **d**

Figure 7: Two different views of a stapler are shown in (a) and (b). From *each* view one can compute a pair of invariants. The process is as follows (just considering (a) and (c)): compute the projectivity that maps each of the four points marked onto its symmetrically corresponding point. Then, map one of the lines, in this case the righthand one, using this projectivity onto the frame of the other half object. This is marked in (c). By construction the intersection of the left line, and the projected line, lies in the plane of the four points. This yields a fifth point in the plane, and hence the pair of projective invariants given in table 3. The process can be repeated for (b) and (d) and the invariants compared. Note that invariants can also be computed by projecting the left line onto the right half image; these invariants will be functionally dependent on the first two values.

| | $I_1$ | $I_2$ |
|---|---|---|
| view 1 | -4.85 | -0.211 |
| view 2 | -5.01 | -0.211 |

Table 3: The measured invariants for each of the two views in figure 7 are given. The values remain reasonably constant under a change in viewpoint.

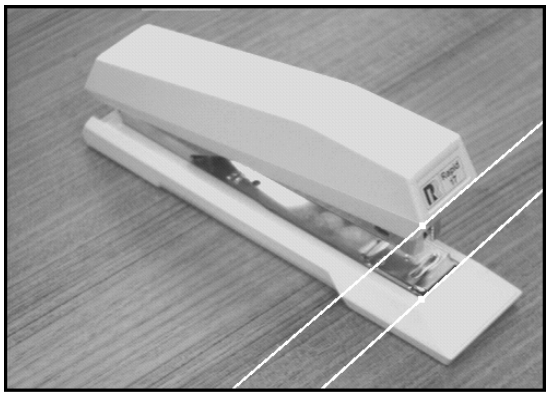


Figure 8: The epipolars for two marked points are shown. The four coplanar points marked in figure 7 were used to determine the epipolar structure. Note that the corresponding points (by symmetry) lie on the epipolars.

### 3.1 Verifying the Mirror Symmetry Assumption

Given the correspondence of the four reference points, we can determine the epipolar calibration between each *side* of the object. This can be used as a test of the mirror symmetry assumption by checking that points on the right side lie on their corresponding epipolar lines generated by points on the left side. In figure 8 the epipoles of two points on the left hand half of the object (as we look at it) are given. Note that the corresponding points do in fact lie on the epipoles.

## 4 Discussion

### 4.1 Applications

One application of the invariants described within this paper is for 3D object recognition. The invariants can be used as model indexes to generate recognition hypotheses. We can compare such a system to that of [14]. The hypotheses are *verified* by projecting the 3D object model into the image, and determining whether there is image support (edges) for the projected outline.

To do this a 3D projective model is needed for every object in the library. There are a number of ways that such a model can be constructed: one way is by using a 3D CAD type model that directly provides Euclidean information about the object. An alternative is to use projective structure gained from two views, such as demonstrated in [5, 6, 9]. As we have access to the actual objects we use the former.

Once the matching invariants have provided a correspondence between a set of eight points on the model and in the world we can transform the model into the frame defined by the camera using a 3D projectivity (such a transform only has 15 degrees of freedom, and is thus determined by five pairs of corresponding points; eight points provide an overconstrained system). The transformed model is then projected into the image to show registration on the measured features. Sample images taken from figures 2 and 3 are shown in figure 9. The good correspondence of the model to the image features shows how well the process performs.

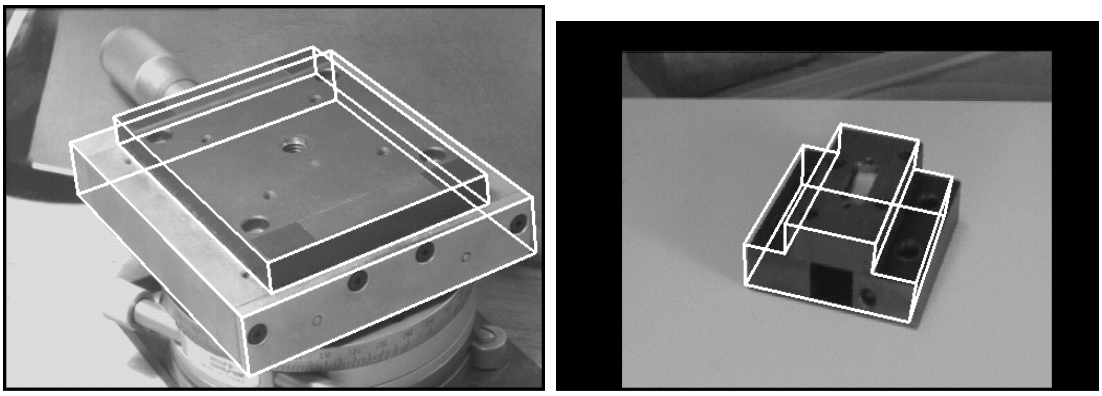
### 4.2 Conclusions

In this paper we have shown how projectively invariant indexing functions can be constructed for 3D point sets given some assumption on structure. In itself this makes a significant contribution to the applications of invariant theory and disspells the belief that no invariant exist for (non general) 3D point sets. However, there is still much work to be done if these descriptors are to be used within a reliable object recognition system. In particular:

- The ability of the indexes to *discriminate* objects in a large model library is currently being examined.
- In this paper we have considered only a construction for 4 *dof.* figures. Current work is investigating to what degree of polyhedra the construction can be applied. Furthermore, attention needs to be paid to the cases where the objects are not position free and are thus over constrained. Spotting this situation may not be too hard, but dealing with it in a principled manner does appear to be difficult.

**Acknowledgements**

Financial support was provided by General Electric, The University of Oxford, the University of Iowa and Esprit Project VIVA. We are grateful to Sabine Demey for providing one of the image sequences and its segmentation, and to Tom Binford for helpful discussions.



a b

Figure 9: In these two images we show the registration of 3D models onto the test images. To perform this task one has to compute the position of the objects in 3D projective space. This type of information is useful not only for recognition, but also for navigation.

## References


[1] Barrett, E.B., Payton, P.M. and Brill, M.H. "Contributions to the Theory of Projective Invariants for Curves in Two and Three Dimensions," Proceedings $1^{st}$ DARPA-ESPRIT Workshop on Invariance, p.387-425, March 1991.

[2] Beardsley, P.A., Sinclair, D.A. and Zisserman, A. "Ego-Motion from Six Points", Insight meeting, Catholic University Leuven, February, 1992

[3] Burns, J.B., Weiss, R.S. and Riseman, E.M. "The Non-existence of General-case View-Invariants," *Geometrical Invariance in Computer Vision,* Mundy, J.L. and Zisserman, A. editors, MIT Press, 1992.

[4] Clemens, D.T. and Jacobs, D.W. "Model Group Indexing for Recognition," Proceedings CVPR91, p.4-9, 1991, and *IEEE Trans. PAMI,* Vol. 13, No. 10, p.1007-1017, October 1991.

[5] Demey, S., Zisserman, A. and Beardsley, P. "Affine and Projective Structure from Motion," Proceedings BMVC92, p.49-58, 1992.

[6] Faugeras, O. "What can be Seen in Three Dimensions with an Uncalibrated Stereo Rig?" Proceedings ECCV2, p.563-578, 1992.

[7] Forsyth, D.A., Mundy, J.L., Zisserman, A.P., Coelho, C., Heller, A. and Rothwell, C.A. "Invariant Descriptors for 3-D Object Recognition and Pose," *PAMI*-13, No. 10, p.971-991, October 1991.

[8] Forsyth, D.A., Mundy, J.L., Zisserman, A.P. and Rothwell, C.A. "Recognising Curved Surfaces from their Outlines," Proceedings ECCV2, p.639-648, 1992.

[9] Hartley, R.I., Gupta, R. and Chang, T. "Stereo from Uncalibrated Cameras," Proceedings CVPR92, p.761-764, 1992.

[10] Lamdan, Y., Schwartz, J.T. and Wolfson, H.J. "Object Recognition by Affine Invariant Matching," Proceedings CVPR88, p.335-344, 1988.

[11] Mohr, R. and Morin, L. "Relative Positioning from Geometric Invariants," Proceedings CVPR91, p.139-144, 1991.

[12] Mohr, R. "Projective Geometry and Computer Vision," in *Handbook of Pattern Recognition and Computer Vision,* Chen, Pau and Wang editors, 1992.

[13] Moses, Y. and Ullman, S. "Limitations of Non Model-Based Recognition Systems," Proceedings ECCV2, p.820-828, 1992.

[14] Rothwell, C.A., Zisserman, A., Mundy, J.L. and Forsyth, D.A. "Efficient Model Library Access by Projectively Invariant Indexing Functions", Proceedings CVPR92, p.109-114, 1992.

[15] Rothwell, C.A., Forsyth, D.A., Zisserman, A. and Mundy, J.L. "Extracting Projective Information from Single Views of 3D Point Sets," *TR OUEL 1973/93*, Department of Engineering Science, Oxford University, Oxford, 1993.

[16] Sugihara, K. *Machine interpretation of Line Drawings,* MIT Press, 1986.

[17] Taubin, G. and Cooper, D.B. "Recognition and Positioning of 3D Piecewise Algebraic," Proceeding DARPA Image Understanding Workshop, p.508-514, September 1990.

[18] Van Gool, L. Kempenaers, P. and Oosterlinck, A. "Recognition and Semi-Differential Invariants," Proceedings CVPR91, p.454-460, 1991.

[19] Wayner, P.C. "Efficiently Using Invariant Theory for Model-based Matching," Proceedings CVPR91, p.473-478, 1991.

[20] Weiss, I. "Projective Invariants of Shapes," Proceedings DARPA Image Understanding Workshop, p.1125-1134, April 1988.