Planar Object Recognition using Projective Shape Representation

C.A. Rothwell*, A. Zisserman*, D.A. Forsyth⋆ and J.L. Mundy†

(∗) Robotics Research Group, Department of Engineering Science, University of Oxford, Parks Rd, Oxford OX1 3PJ, UK.

(⋆) Department of Computer Science, University of Iowa, Iowa City, IA 52242.

(†) GE Center for Research and Development, 1 River Rd, Schenectady, NY 12345.

August 1993

Abstract

We describe a model based recognition system, called LEWIS, for the identification of planar objects based on a projectively invariant representation of shape. The advantages of this shape description include simple model acquisition (direct from images), no need for camera calibration or object pose computation, and the use of index functions. We describe the feature construction and recognition algorithms in detail and provide an analysis of the combinatorial advantages of using index functions. Index functions are used to select models from a model base and are constructed from projective invariants based on algebraic curves and a canonical projective coordinate frame. Examples are given of object recognition from images of real scenes, with extensive object libraries. Successful recognition is demonstrated despite partial occlusion by unmodelled objects, and realistic lighting conditions.

1 Introduction

1.1 Overview

In the context of this paper, recognition is defined as the problem of assigning the correct label to an object seen in a perspective view. Recognition is considered successful if the 2D geometric configuration of an object in an image can be explained as a perspective projection of a geometric model of the object. In this paper we restrict ourselves to planar objects, although many man-made 3D objects can be decomposed into recognisable planar patches.

A key aspect of the system is the use of the projective transformation group to represent perspective image projections. Most object recognition systems use approximations to perspective, such as affine or orthographic camera models. Such approximations are often valid, but viewing conditions where depth variation of the object is significant compared to viewing distance, or those that consider a wide viewing angle, require a complete representation of the effects of perspective image formation.

Perspective, or central projection, does not exhibit the full range of geometric transformation possible under the projective model. For example, convexity is preserved under perspective projection (so long as the imaged object does not intersect the image plane), but not under full projective transformation. However, the convenience of homogeneous coordinates, the consequent linearity of projective transformations, and the associated group properties motivate our use of projective geometry throughout. The restrictions associated with perspective are introduced in the recognition process as part of hypothesis confirmation. It should also be noted that the parameters associated with internal camera calibration are implicit in projective projection, so object description and recognition is not dependent on camera geometry.

The recognition system, which is called LEWIS (Library Entry Working through an Indexing Sequence), is designed around the use of invariant indexing functions to represent each object class. An invariant is defined as a function which measures some geometric properties of an object but whose value is independent of projective frame. These indexing functions are computed from the geometric coordinates or coefficients of a small group of image features such as points, lines and conics. The emphasis is on efficiently indexing a large model library, where the index keys are constructed from invariant function values. In practice, the index derived from one view (a model acquisition view) can be used to access the object model in any subsequent view.

Since these indexes can be derived from any viewpoint, it follows that any unoccluded view of the object can serve as a model. We derive the invariant values for the library from a typical view and also include information which is needed for verification, such as the main geometric features of the object and the bounding box of the features. It is beneficial to acquire the model directly from an image since the resulting geometry reflects the actual shape of the object, including rounded corners and other manufacturing artifacts.

Invariant indexing functions are derived according to two approaches. In the first approach, *algebraic invariants* are based on classical results derived from the projective geometry of algebraic curves [Semple52]. The fundamental invariant in projective geometry is the cross ratio, which is defined for four collinear points in terms of ratios of distances between the points. A similar invariant can be defined for four lines concurrent at a single point. More general algebraic invariants can be derived from configurations of conics, points and lines. For example, a cross ratio can be generated from two points and a conic; this arises because the line passing through the two points intersects the conic in two other collinear points. These and other algebraic invariants will be discussed in detail in section 2.2.

The second approach to the construction of invariant indexing functions is the use of projective coordinate frames. In the projective plane, four points, no three of which are collinear, define unique projective coordinates for any other point in the plane. These projective coordinates are invariant to any projective transformation of the plane. We can define a particular frame, usually a fronto-parallel view, which we call the *canonical frame*. Invariant indexes are constructed from a sample of points on the boundary of the object when projected onto the canonical frame. The advantage of the canonical frame construction is that the object boundary does not have to be an algebraic curve.

These ideas have been incorporated into a complete recognition system over the past four years. The system, LEWIS, has been tested on a large set of images and under varying levels of occlusion and clutter. The major issues which have been examined in the evaluation of LEWIS are:

1. The dependence of recognition complexity on the number of models in the database.
2. The discrimination power of projective invariant descriptions, particularly in the presence of clutter and occlusion.
3. The effect of illumination, object surface properties and feature segmentation on invariant values.
4. The practicality of constructing object models directly from an example object view.

We will explore these issues in later sections, but first it will prove useful to establish the framework for object recognition. In particular, we establish the benefits of a model library accessed by invariant keys.

1.2 Recognition Framework

Recognition consists of two process: the first is the identification of which object is potentially present in the scene; and the second is the establishment of a correspondence between the image and the identified model features. Often, these processes are not distinct, though together they can be partitioned into three stages that should be contained within any recognition system (these are similar to those defined in [Grimson90], p.33):

Grouping: what subset of the data belongs to a single object?

Indexing: which object model projects to this data subset?

Verification: how much image support is there for this correspondence?

Naturally, these stages represent an idealised decomposition; robust recognition generally requires numerous interactions between the stages. However, this structure yields a productive framework for defining and measuring the general characteristics of recognition systems.

The aim of grouping (also called *perceptual organisation* [Lowe85], *selection*, or *figure-ground discrimination*) is to provide an association of features that are likely to have come from a single object in a scene. Image features which are exploited in grouping cover all levels of image segmentation, for instance: edgels; corners; algebraic features such as lines and conics; smooth curves represented as splines; and feature descriptions based on regions, such as texture. These features are typically grouped together using cues such as proximity, parallelism [Binford81, Lowe85] collinearity and approximate continuity in curvature [Cox92, Shaashua88]. In the work reported here, we exploit many of these techniques to generate feature groups from which invariant index functions are constructed.

Indexing addresses the problem of model hypothesis generation. For a small number of models, for example two or three, it is reasonable to try simply to find image feature support for each model. This approach is typical of many existing systems [Ayache86, Ayache87, Grimson90, Huttenlocher88, Lowe87, Murray87, Pollard89]. As the size of the model library increases, this approach becomes computationally too expensive. It is then more effective to choose potential models from the library based on the observed image features. That is, image feature measurements are used to *index* into the model base. The work presented in this paper demonstrates that efficient indexing strategies can be constructed, and that through using them, dramatic improvements in hypothesis generation efficiency can be achieved.

The final stage is verification. Grouping and indexing have hypothesised a match between an object and a small number of image features. This match is used to project the model onto the image. The validity of the model hypothesis and model-to-image feature correspondences is determined by searching for image features that have not been used in the construction of indexes. These are features that, for instance, have been missed by the grouping stage. The more features that can be found which are close to the projected model boundaries, the more likely it is that the initial hypothesis is correct. Once all possible correspondences have been accepted or ruled out, a conclusion as to the identity of an object can be made. Generally, a hypothesis is considered successful if the error between projected model features and corresponding image features is below some threshold and a reasonable fraction of the object outline is covered by image features.

There are three distinct algorithms that have been used to compute correspondence. In the first approach, *interpretation trees* [Ayache87, Brooks83, Ettinger88, Fisher89, Grimson87, Murray87, Pollard89, Reid91], the set of correspondences is grown incrementally according to a branch and bound search algorithm. Features are added according to their consistency with a model hypothesis associated with each node in the graph. Consistency is also a function of the specific set of features defined by the path from the root of the search tree to the current node.

The second approach, *hypothesise and test* [Ayache86, Bolles87, Goad83, Huttenlocher87, Lowe87], generates model hypotheses exhaustively from the library, although properties of small feature groups can be used to suggest initial trial feature correspondences. These hypotheses are tested by establishing model-dependent and priority ordered checklist of other features which must be present to satisfy the hypothesis. A set of *focus features* are defined for each object which are easily extracted and also provide maximum discrimination among object classes.

The third approach, *pose clustering* [Cass92, Stockman87, Thompson87], uses the concept of pose consistency to generate hypotheses. An object is projected onto an image under a single transformation acting on all points of the object. The image projection of an object is composed of a 3D Euclidean transformation (called pose), followed by a perspective mapping. The 3D pose can be computed from various model-to-image feature correspondences and should be the same for all correct correspondences from a single object. The search for correct correspondences is then the problem of finding clusters in pose space.

The recognition system reported here shares many characteristics with these approaches, particularly in the stages of feature grouping and hypothesis testing, but differs considerably in how model hypotheses are generated and feature correspondences are established. Our approach to these stages centres on the use of index functions that we now define more formally.

1.3 Indexing Functions

The concept of the indexing function can be developed formally as follows: the index is considered to be a vector, $\mathbf{M}$, which selects a particular model from the library. Each model consists of the set of significant geometric features of the object boundary as well as ancillary information required for hypothesis confirmation such as, the bounding box of the features, pixel chains from which the boundary features are constructed, and perhaps texture or other details of the object surface properties.

The model index is a function only of a set of *projected* model features, $\mathbf{F}$, that is $\mathbf{M}$ can be computed from any image projection of the model features. The practical consequence is that models can be constructed simply by acquiring one or a few image views of the object in isolation. If $\mathbf{F}_{model}$ is the set of features actually on an object, and T is the transformation from the object in an arbitrary pose onto the camera, then:

$$\mathbf{M}(T(\mathbf{F}_{model})) = \mathbf{M}(\mathbf{F}_{model}).$$

This equation states that the index function is *(scalar) invariant* [Forsyth91] to transformations of the object which result from different viewpoints. In the results reported here, the index functions are invariant to projective transformations of the image plane. Each element of the index vector $\mathbf{M}$ is an invariant measure computed from a group of model features such as conics, lines, points and plane curve segments. Ideally, the index function should uniquely retrieve a model from the library, but in practice it is likely that a small number of models are retrieved with the same index. Even so, the search cost is considerably reduced below that of testing every member of the library.

The concept of an indexing function described above assumes that both the indexes for the model and for the object can be measured perfectly in a scene. In practice, the measurements are imprecise due both to modeling and imaging errors[1]. It is therefore necessary to provide a range of invariant values in the construction of the index function. In LEWIS, the range is established by quantising the index space according to the observed variation in invariant values due to the effects just mentioned. The quantised index value is denoted by $\mathbf{Q}$ and a quantisation is selected so that,

$$\mathbf{Q}(\mathbf{M}(\mathbf{F}_{model}) + \mathbf{E}_{model}) = \mathbf{Q}(\mathbf{M}(\mathbf{F}_{image}) + \mathbf{E}_{image}).$$

[1] This is not just due to random image noise which is often considered to be the sole cause of error, but also due to events in the image which are not modeled or expected. Examples of unmodeled image events are: specularities; surface texture; and impinging objects.

Note that the quantisation function $\mathbf{Q}$ is the same for both the model and the image. This is a direct result of being able to acquire models from images. The error characteristics $\mathbf{E}_{model}$ and $\mathbf{E}_{image}$ can also be assumed to be the same.

Other recognition systems have also exploited index functions based on invariants. A system using projective invariants is described by Nielsen for identifying and tracking mobile robots [Nielsen88]. Early versions of the system described here are reported by Forsyth, *et al.* [Forsyth91]. Indexing functions based on affine invariants have formed the basis for a number of planar object recognition systems, for instance the series of papers by Kalvin *et al.* [Kalvin86, Schwartz87, Lamdan88], Wayner [Wayner91], Clemens and Jacobs [Clemens91], Huttenlocher [Huttenlocher91], Taubin and Cooper [Taubin91]. Other avenues of invariant research have been covered by Weiss [Weiss88], Stein and Medioni [Stein92], Califano and Mohan [Califano92], Gueziec and Ayache [Gueziec93], and Rigoutsos and Hummel [Rigoutsos91].

It is also possible to gain some of the advantages of indexing without using index functions that are strictly invariant. For example, Jacobs describes an approach to indexing 3D objects using one parameter families of index values in image transform space. One can then select models based on proximity to these index sets [Jacobs92]. Another approach is the use of quasi-invariants [Binford93], where functions are constructed that are not invariant under general perspective viewing, but are reasonably constant over most practical viewing conditions. The quasi-invariants that have been suggested are *invariants* of more restricted transformation groups such as affine and equiform (scaled Euclidean). Affine transformations apply when the depth change along the object plane is small compared to the distance from the center of perspective. The equiform case occurs when the object plane is parallel to the image plane.

1.4 Outline of the Paper

Section 2 introduces the notation used in the rest of the paper, defines the algebraic and canonical frame invariants, and describes the segmentation and grouping procedures used in LEWIS. Section 3 surveys the recognition architecture with results and statistics given for the systems working on real images. Finally, section 4 highlights weaknesses in the current approach and suggests directions for future research.

2 Invariant Indexing Functions

2.1 Notation

When homogeneous coordinates are used points on the plane are represented by a triple $\mathbf{x} = (x_1, x_2, x_3)^T = (\lambda x, \lambda y, \lambda)^T$ where $(x, y)^T$ are the standard Euclidean plane coordinates of the point and λ is an arbitrary (non-zero) projective scale factor. Points in the projective plane are equivalent for all values of λ. Lines are defined by $\mathbf{l} = (a, b, c)^T = (\mu \sin\theta, -\mu\cos\theta, \mu d)^T$, where θ is the orientation of the line with respect to the x axis and d is the perpendicular distance of the line from the origin. μ is the projective scale factor for lines.

The incidence of a point and a line in the projective plane is given by, $ax_1 + bx_2 + cx_3 = 0$, or in vector notation, $\mathbf{l} \cdot \mathbf{x} = 0$.

A conic is the set of points $(x_i, y_i, 1)^T$ that satisfy:

$$ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i + f = 0. \tag{1}$$

A more convenient representation of a conic uses a planar point $\mathbf{x}$ and a quadratic form $\mathtt{C}$:

$$\mathbf{x}^T \mathtt{C} \mathbf{x} = 0, \tag{2}$$

where:

$$\mathtt{C} = \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix}. \tag{3}$$

From now on, typewriter font denotes matrices, bold letters denote vectors, large letters denote model objects and small letters denote image objects. For example, $\mathtt{C}$ is a model conic, $\mathbf{X}$ a model point, and $\mathtt{c}$ and $\mathbf{x}$ their images in a view where recognition is to be achieved.

2.1.1 Projective Transformations

A projective transformation $\mathtt{T}$ between two planes is represented as a 3×3 matrix acting on homogeneous coordinates of the plane. It is a linear mapping on homogeneous points. A homogeneous representation means that only ratios of matrix elements are significant, and consequently the transformation has 8 degrees of freedom. Under imaging, this transformation models the *composed* effects of 3D rigid rotation and translation of the world plane (camera extrinsic parameters), perspective projection to the image plane, and an affine transformation of the final image which covers the effects of camera intrinsic parameters. The effects of radial distortion due to the camera lens are not modeled.

All of the parameters of these separate transformations cannot be recovered uniquely from a single 3×3 matrix, since there are 6 unknown pose parameters, and 5 unknown internal camera parameters (these are camera centre, focal length, aspect ratio and the angle between the coordinate axes of the image plane). For plane to plane perspective transformations, there are therefore 11 unknowns but only 8 constraints. Fortunately, the invariant description, and model projection used in the recognition system, do not require explicit knowledge of either the pose or the internal camera parameters. We need solve only for the independent parameters of the projective transformation $\mathtt{T}$. Note that projectivities form a group, and so most notably every action has an inverse and the composition of two projectivities is also a projectivity. Consequently, two images from different viewpoints of the same planar object are always related by a projectivity.

The mapping of four points between two planes, of which no three points are collinear, is sufficient to determine the transformation matrix $\mathtt{T}$. Each point provides two linear constraints on the transformation parameters, therefore four independent points provide the required $4 \times 2 = 8$ constraints. Corresponding points (x_i, y_i) and (X_i, Y_i) are represented by homogeneous 3 vectors $(x_i, y_i, 1)^T$ and $(X_i, Y_i, 1)^T$. The projective transformation $\mathbf{x} = \mathtt{T}\mathbf{X}$, $(|\mathtt{T}| \neq 0)$ is:

$$k_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix},$$

where k_i is an arbitrary non-zero scalar. Note that in using this formulation we are unable deal with plane points lying on the ideal line; this is, however, unimportant as in practice all of the points to be transformed lie within the finite and bounded image plane. For $N \geq 4$ points, singular value decomposition can be used to compute $\mathtt{T}$. The computation can be formulated as minimising $||\mathtt{A}\mathbf{t}||$ subject to $||\mathbf{t}|| = 1$, where $\mathbf{t}$ is the nine-element vector of transform parameters and $\mathtt{A}$ is a $N \times 9$ element matrix of elements formed from the coordinates of the matched image and model points.

Using similar algorithms, projectivities can be computed between sets of lines, or as shown in [Rothwell94] for different combinations of points, lines and conics. The projective transformation of lines is closely related to that for points. Given the transformation matrix for points, $\mathtt{T}$, lines transform according to

$$\mathbf{l} = \mathtt{T}^{-T}\mathbf{L},$$

where $\mathtt{T}^{-T}$ is the inverse transpose of $\mathtt{T}$. The transformation of conics is as follows: given $\mathtt{C}$ and its respective image conic $\mathtt{c}$, and point transformation matrix $\mathtt{T}$, is constrained by:

$$\mathtt{c} = \kappa \ . \ \mathtt{T}^{-T}\mathtt{C} \ \mathtt{T}^{-1}. \tag{4}$$

2.2 Algebraic Invariants

There are three different algebraic invariants used within the recognition system for coplanar algebraic features: five lines; a pair of conics; and a conic and two lines. Their derivation is given, for example, in [Mundy92]. There are many other possible configurations (with points, cubics, etc.) that could also be used to generate invariants. The particular configurations used in LEWIS have been chosen because the constituent geometric features can be produced directly and accurately from segmentation. In contrast, points are extracted most accurately *indirectly* by intersecting lines.

2.2.1 Five Coplanar Lines

Given five coplanar homogeneous lines $\mathbf{l}_i$, where $i \in \{1, .., 5\}$, two functionally independent projective invariants are defined using determinants

$$I_1 = \frac{|\mathtt{M}_{431}||\mathtt{M}_{521}|}{|\mathtt{M}_{421}||\mathtt{M}_{531}|} \qquad \text{and} \qquad I_2 = \frac{|\mathtt{M}_{421}||\mathtt{M}_{532}|}{|\mathtt{M}_{432}||\mathtt{M}_{521}|}, \tag{5}$$

where $\mathtt{M}_{ijk} = (\mathbf{l}_i, \mathbf{l}_j, \mathbf{l}_k)$.

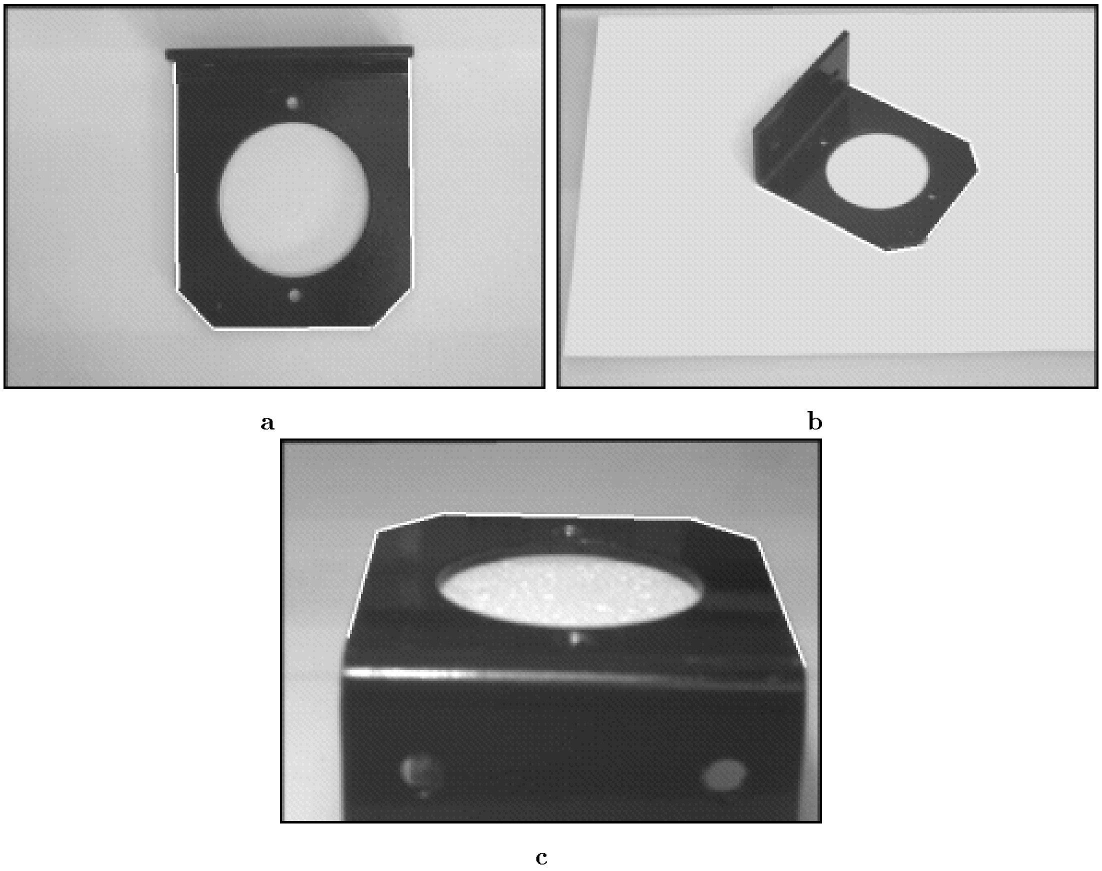


Figure 1: *Examples of similarity, affine and perspective images of a bracket. For each view the lines used to compute the invariants are marked in white. The pair of five-line invariants are computed using the determinant formulae given in this section. The invariant values for the images, and those actually measured on the object are shown in table 1. The fact that they remain essentially invariant demonstrates the stability of the invariants under real imaging conditions. For reference, the values of affine invariants computed from area ratios are also given in the table.*

Table 1: I_1 *and* I_2 *are five-line invariants computed for the similarity, affine and perspective views of the bracket shown in figure 1.* I_{a1} *and* I_{a2} *are affine invariants defined by the ratio of areas of triangles constructed from the points of intersection of the lines. The values of* I_1 *and* I_2 *are consistent with those measured on the object and vary only slightly with viewpoint which demonstrates the practicality of deriving invariant measures from image features. Note that particularly for the image with substantial perspective distortion, the affine invariants* I_{a1} *and* I_{a2}*, vary considerably more than* I_1 *and* I_2*.*

	I_1	I_2	I_{a1}	I_{a2}
object	0.840	1.236	0.739	1.083
similarity	0.842	1.234	0.706	1.051
affine	0.840	1.232	0.743	1.066
perspective	0.843	1.234	0.623	0.949

Table 2: *The conic-pair invariants computed for the similarity, affine and perspective views of the computer tape shown in figure 2. Note the stability of the measured values with respect to change in viewpoint.*

	I_1	I_2
object	3.073	3.082
similarity	3.074	3.082
affine	3.072	3.080
perspective	3.070	3.078

A major problem with the determinant formulae given in equation 5 is that the invariants can become undefined for certain geometric configurations. The determinant, $|\mathtt{M}_{ijk}|$ vanishes when the lines $\mathbf{l}_i$, $\mathbf{l}_j$ and $\mathbf{l}_k$ are concurrent. In LEWIS, grouping is used to eliminate configurations where both invariants are undefined so that one of the values of I_1 and I_2 can always be used. The grouping algorithm, described later, ensures that only the lines $\mathbf{x}_i, i \in \{1,3,5\}$ are allowed to be concurrent. Since there is no determinant of $\mathtt{M}_{135}$ in I_2, it will always be well formed, though I_1 will sometimes fail.

Examples of the invariants computed for real image distortions are demonstrated in figure 1, and the invariant values given in table 1. The fact that the values remain constant over a change in viewpoint demonstrates the stability of the invariants under image noise.

2.2.2 Two Coplanar Conics

A pair of conics $\mathtt{C}_i, i \in \{1,2\}$ has two independent projective invariants. These can be expressed in terms of ratios of eigenvalues [Quan91], or equivalently

$$I_1 = \frac{Trace[\mathtt{C}_1^{-1}\mathtt{C}_2].|\mathtt{C}_1|^{1/3}}{|\mathtt{C}_2|^{1/3}} \qquad \text{and} \qquad I_2 = \frac{Trace[\mathtt{C}_2^{-1}\mathtt{C}_1].|\mathtt{C}_2|^{1/3}}{|\mathtt{C}_1|^{1/3}}.$$

If the conics are *normalised* so that $|\mathtt{C}_i| = 1$ the invariants take on the simpler form of:

$$I_1 = Trace[\mathtt{C}_1^{-1}\mathtt{C}_2] \qquad \text{and} \qquad I_2 = Trace[\mathtt{C}_2^{-1}\mathtt{C}_1].$$

These invariants have been tested extensively during the development of the system reported in this paper, and have been found to have good noise characteristics. A simple example showing the measured invariants for similarity, affine and perspective views of the computer tape shown in figure 2 are given in table 2. The small deviation of the invariants demonstrates their stability, more complete results are given in [Forsyth91].

2.2.3 A Conic and Two Lines

For a conic $\mathtt{C}$ and two lines $\mathbf{l}_i, i \in \{1,2\}$, a single invariant can be computed:

$$I \quad = \quad \frac{(\mathbf{l}_1^T \mathtt{C}^{-1}\mathbf{l}_2)^2}{(\mathbf{l}_1^T \mathtt{C}^{-1}\mathbf{l}_1)(\mathbf{l}_2^T \mathtt{C}^{-1}\mathbf{l}_2)}.$$

The invariant computed for the similarity, affine and perspective image sequence of the bracket is shown in figure 3. The corresponding invariant values in table 3. Again the stability of the invariant form is demonstrated over a large range of viewpoints.

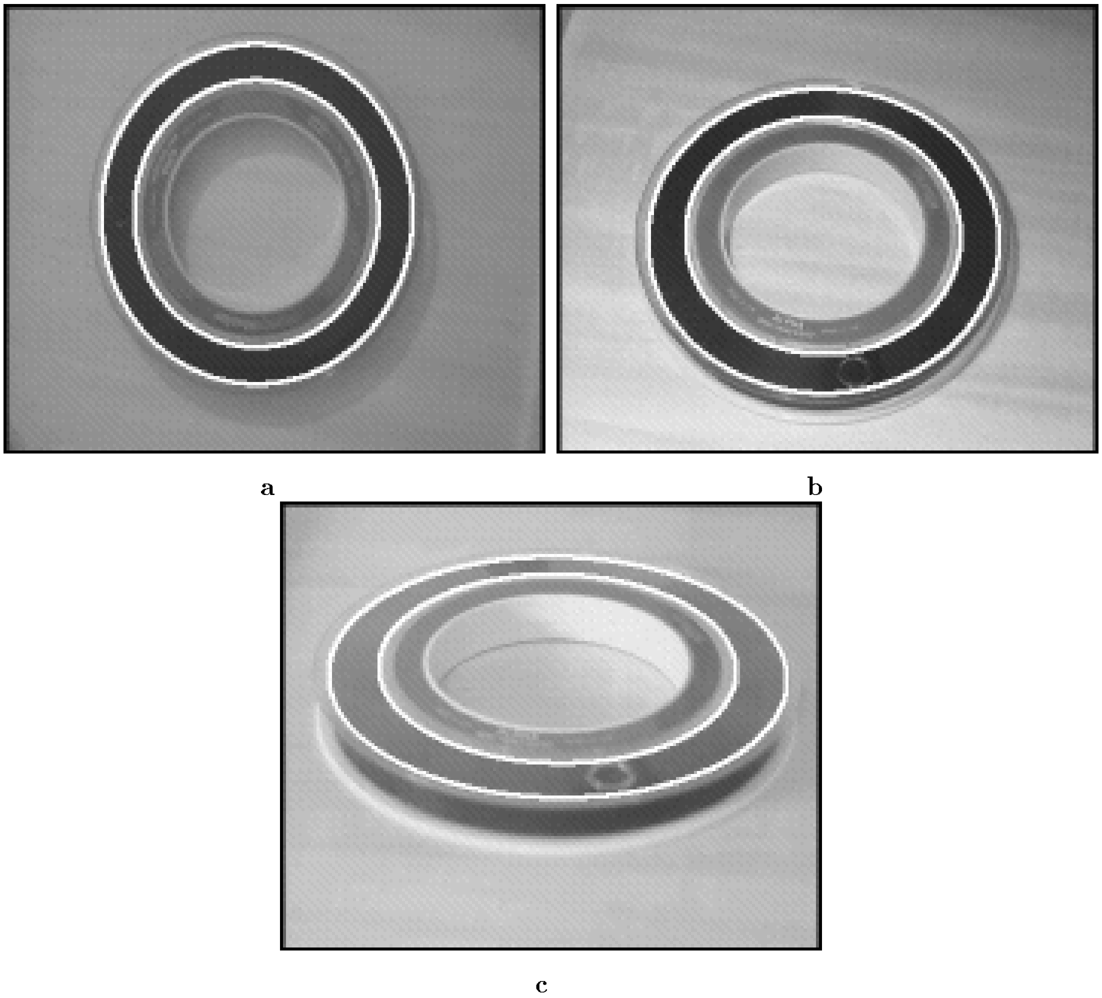


Figure 2: *Similarity, affine and perspective images of a computer tape with the conics used to compute the invariants marked in white. The invariant values are given in table 2.*

Table 3: *The conic and line-pair invariants computed for the similarity, affine and perspective views of the bracket shown in figure 3. Note the stability of the measured values with respect to change in viewpoint. For comparison, an affine invariant is also tabulated. In this case, I_a is defined by the ratio of the areas of a triangle and of the conic itself. One vertex of the triangle is located at the intersection point of the two lines; the other two vertices are defined by the points of tangency to the conic of the pair of lines through the first point that touch the conic. Note that I_a is significantly less stable than I.*

	I	I_a
object	1.33	0.398
similarity	1.33	0.389
affine	1.31	0.403
perspective	1.28	0.437

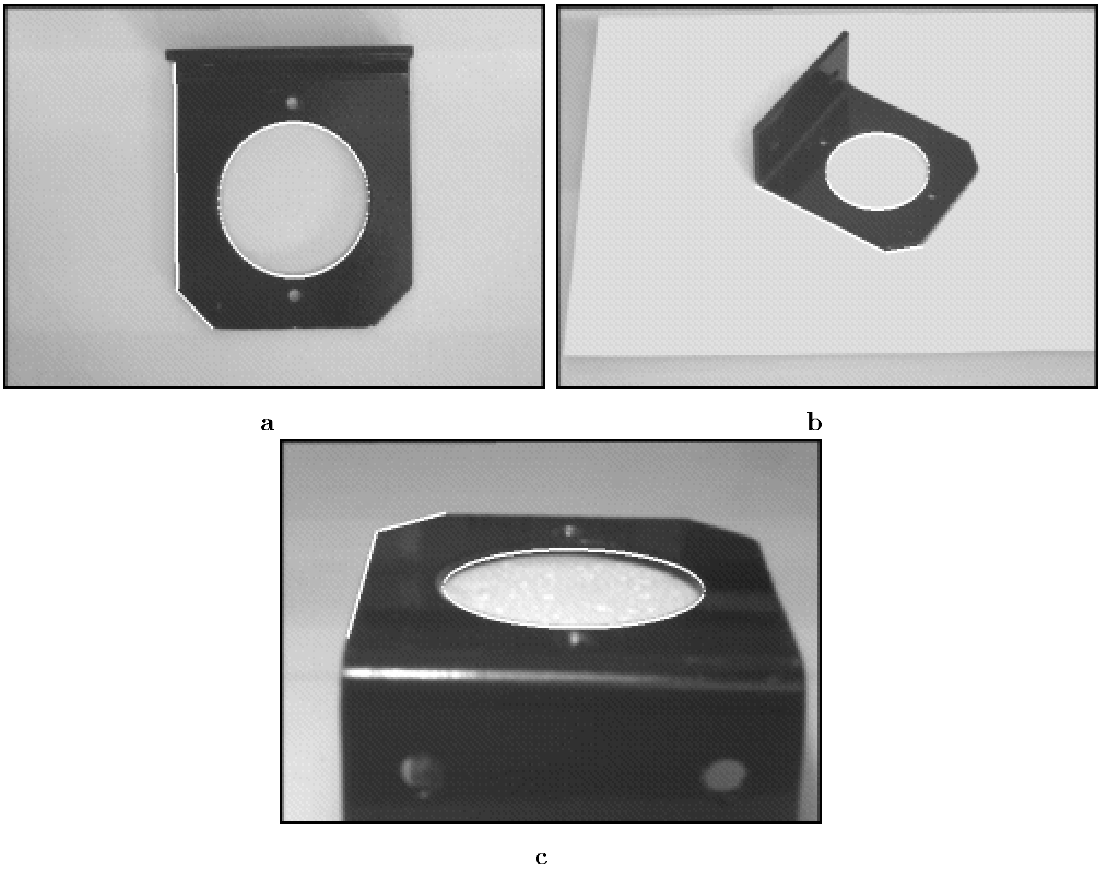


Figure 3: *Similarity, affine and perspective images of a bracket with the conic and two lines used to compute the invariants marked in white. The invariant values are given in table 3.*

We have found in practice that the conic and line pair invariant is not stable enough alone to provide sufficient discrimination for the class of objects used in our experiments. Three independent invariants can be formed from three lines and a conic, using the lines two at a time. The combined index provides better discrimination as explained in section 3.5.1.

2.3 Canonical Frame Invariants

A canonical frame construction can be used to form an invariant *signature* for smooth planar curves. The rest of this section describes the construction of the signature for a non-convex class of plane curves; the work is a projective extension of that of Lamdan, *et al.* [Lamdan88].

First, we illustrate the concept of a canonical frame with a set of of five coplanar points, four used as a projective basis and the fifth to generate invariants. We then show how four *distinguished points* can be defined on a concavity in a plane curve. The rest of the curve can then considered as a set of individual points whose coordinates with respect to the projective basis define the signature.

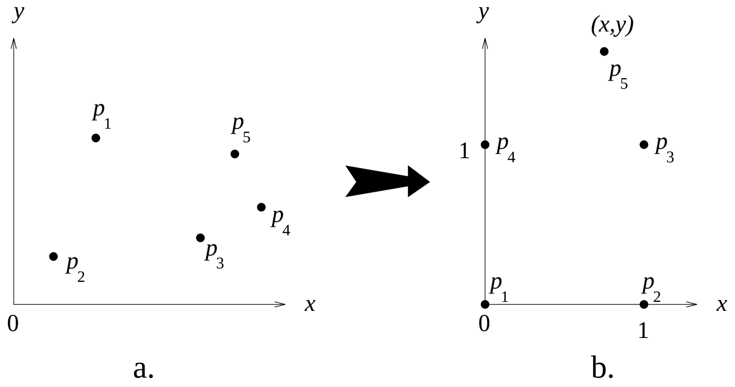


Figure 4: *One way of measuring the invariants of five coplanar points in a image (a) is to compute the projective transformation of four of the points* $\mathbf{p}_i, i \in \{1, \ldots, 4\}$ *to reference points in the canonical frame (b). In this case the projection is to the corners of the unit square. Once this map is known* $\mathbf{p}_5$ *can also be transformed to the new frame and its coordinates* (x, y) *used as invariants.*

2.3.1 Mapping Five Points to the Canonical Frame

As four points define a projective mapping between two frames, the first four points of a set of five can be used to define the map between the image frame and a standard measurement or *canonical frame.* The fifth point can then be mapped to the new frame in which its coordinates are projectively invariant. To ensure that the coordinates really are invariant, the first four points must always be mapped to a standard set of four reference points in the canonical frame. The choice of these points is arbitrary: the corners of the unit square may be used (as in figure 4), or some other frame chosen according to noise performance.

2.3.2 Mapping a Plane Curve to the Canonical Frame

The aim is to find four distinguished points (or lines) on a curve, and use these to define the projectivity $\mathtt{T}$ that can be used to take the whole curve to the canonical frame. The method is shown in figure 5: for the given concavity, the location of the points of bitangency is determined as described in section 2.4.3. These are (A) and (D), and they segment the curve of interest from the rest of the edge chain. This curve segment is known as an $\mathcal{M}$ *curve.* The *cast tangents* are then determined, these are lines tangent to the $\mathcal{M}$ curve that pass through the bitangency points. The points of cast tangency are (B) and (C). The projection of the $\mathcal{M}$ curve into the frame using $\mathtt{T}$ is the *curve signature*; it is a projective representation of the original object curve.

2.3.3 Discrimination

Examples of the canonical frame construction for single views of three different objects are given in figure 6. A single $\mathcal{M}$ curve for each spanner and the pair of scissors is marked in (a), (b) and (c), and these are projected into the same canonical frame in (d). All three canonical curves are different and so the

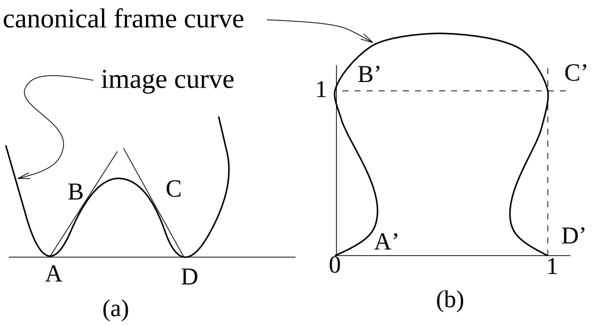


Figure 5: *(a) Construction of the four points necessary to define the canonical frame for a concavity. The first two points, (A) and (D), are points of bitangency that mark the entrance to the concavity. Two further distinguished points, (B) and (C), are obtained from rays cast from the bitangent contact points and tangent to the curve segment within the concavity. These four points are used to map the curve to the canonical frame. (b) the curve in the canonical frame. A projection is constructed that transforms the four points in (a) to the corner of the unit square. The same projection transforms the curve into this frame.*

construction provides discrimination (although the spanner curves extracted from (a) and (b) are reasonably similar, they are sufficiently different for recognition purposes).

2.3.4 Semi-Local Description

Non-global descriptions must be used if objects are to be recognised under occlusion; the canonical frame construction provides a *semi-local* object description. Furthermore, for genuine tolerance to occlusion, there must be a number of different descriptors on each object so that there is not an excessive requirement for any single object region to be visible. This is called *redundancy*. Single objects frequently possess large numbers of bitangents (see figure 13); this provides a high degree of redundancy as each bitangent can be used to derive a canonical frame. However, such a high degree of redundancy is not required for recognition, and only a few bitangents are actually used for shape description. For the spanner in figure 6a, four suitable bitangents exist and bound $\mathcal{M}$ curves. The four $\mathcal{M}$ curves are shown in figure 7.

2.3.5 Stability

The final criteria discussed in this section is stability: if the construction is to be useful, similar frame curves must be obtained from different views of the same object curve. Even if the curves are not identical, they should be sufficiently similar so that discrimination between objects is possible. This is the case for the canonical frame construction. Three very different views of a spanner are given in figure 8 (they vary by a full perspective distortion, and not just an affine one). The same $\mathcal{M}$ curve is marked in each image, and these are mapped to the canonical frame in (d). As can be seen, the construction is stable even over a wide change in viewpoint.

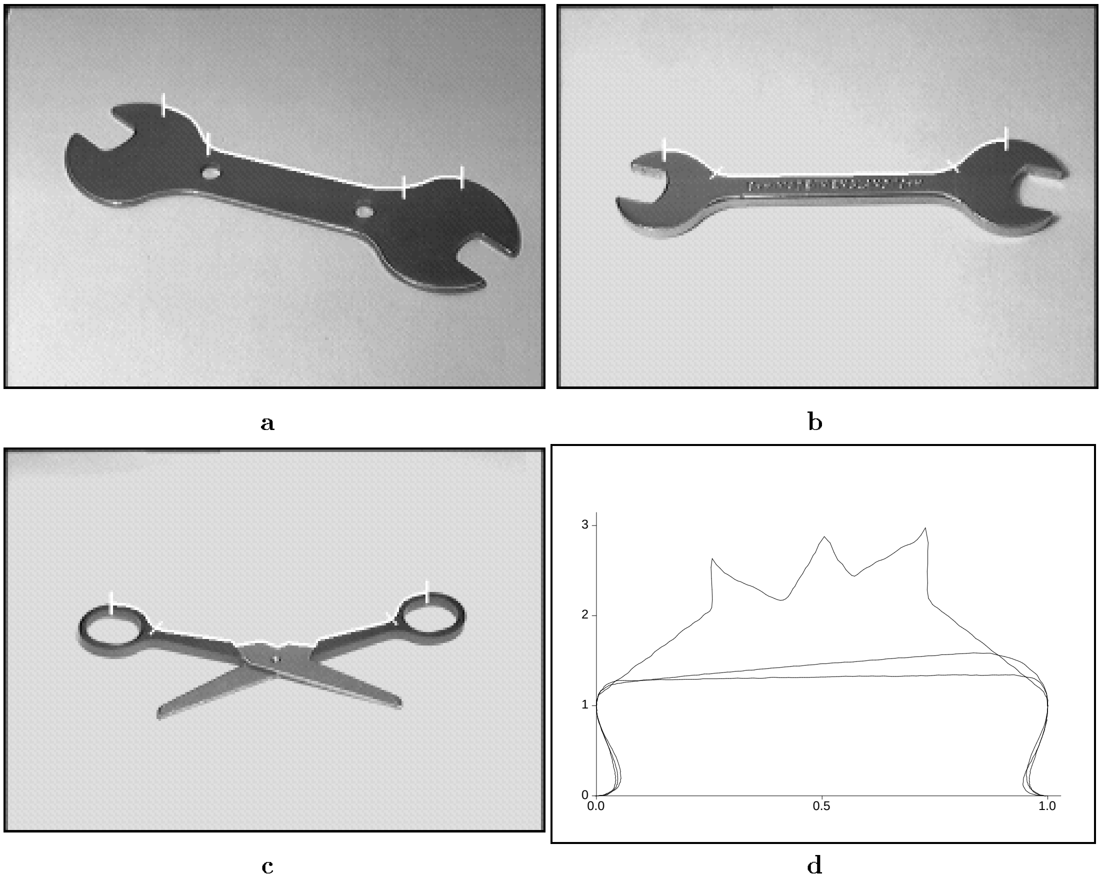


Figure 6: *In (a)–(c) a single $\mathcal{M}$ curve and the four distinguished points are marked on each object. The three curves are projected to the canonical frame and superimposed in (d). The scissor $\mathcal{M}$ curve is obviously very different from each spanner, but in fact the two spanner curves are sufficiently different for recognition purposes.*

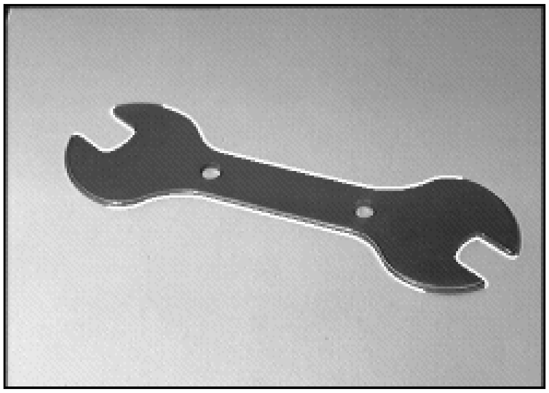


Figure 7: *Even for a simple object such as a spanner there is a sufficient degree of redundancy when the canonical frame construction is used. Here, four useful $\mathcal{M}$ curves are shown that essentially cover the entire perimeter of the object, and yet each one is potentially a sufficient recognition cue on its own.*

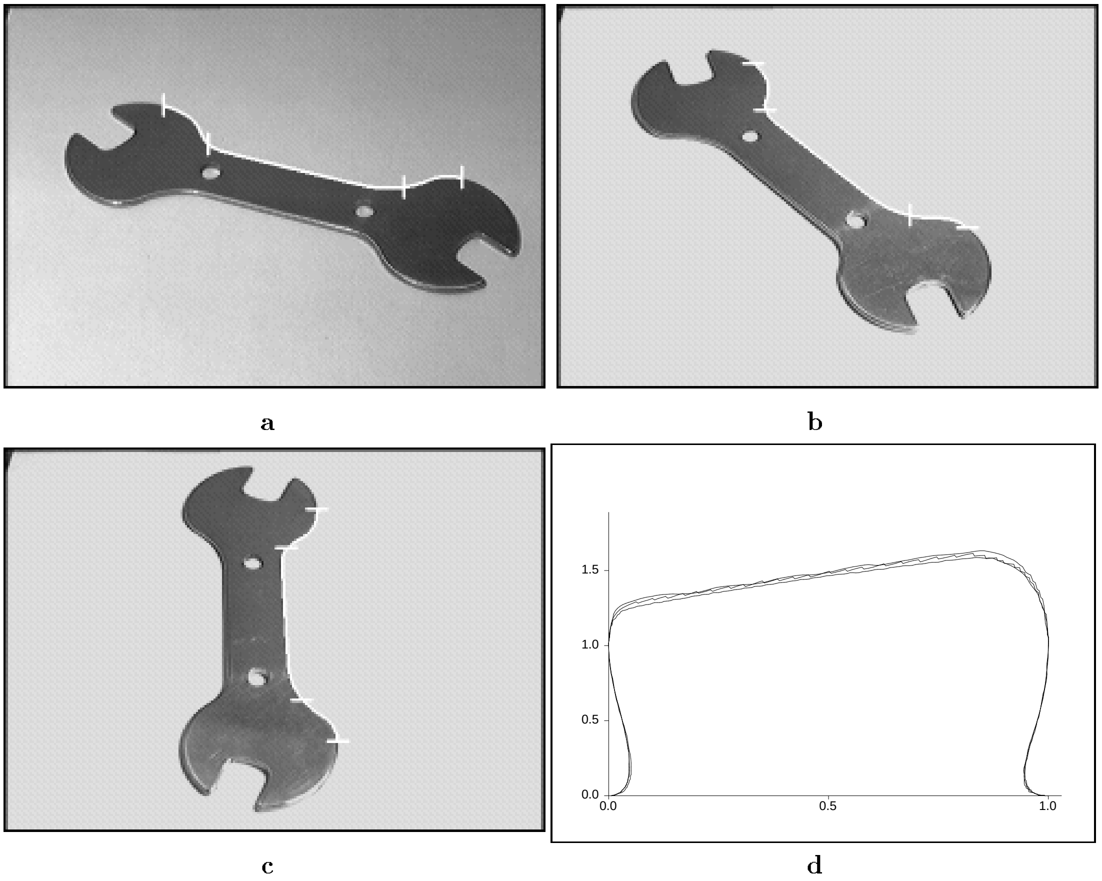
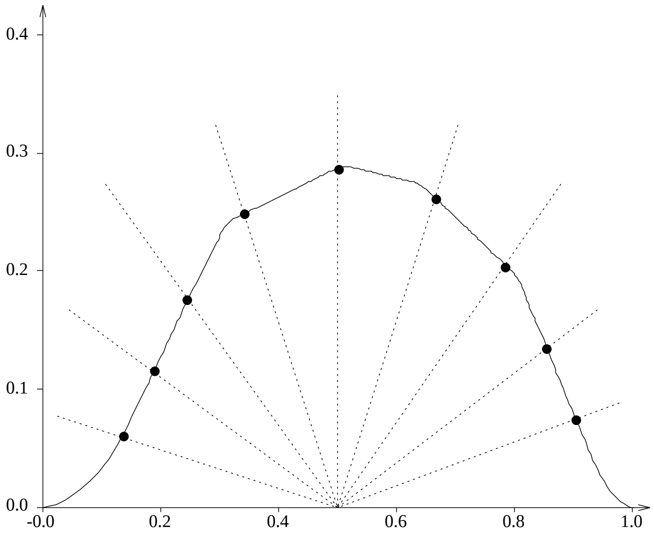


Figure 8: *(a)–(c) three views of a spanner with the extracted $\mathcal{M}$ curves and distinguished points marked. Note the very different appearance due to perspective effects. (d) shows the canonical frame curves for the three different views. The curves are almost identical demonstrating the stability of the method. Of course, the same curve would result from a projective transformation between the object and canonical frame.*

2.3.6 Index Functions and Discrimination

The canonical frame curves are essentially projectively invariant *templates* for the shapes, and so one may attempt $\mathcal{M}$ curve recognition using traditional curve correlation matching techniques with model curves. Such techniques would lead to a linear search of the model library, so instead, an index is constructed from the signature. The goal is to compute a function of the signature that uniquely identifies the $\mathcal{M}$ curve. The current solution is to use a few points along the signature to construct the index. This data is adequate to distinguish the spanners and brackets used in our experiments. The complete signature is retained as part of the model description and used during verification as a more complete representation of the object shape.

The invariant indexes used are constructed using the geometry of figure 9. This construction is similar to the technique of *footprints* [Lamdan88], though points are used rather than areas. The drawback of this method for measuring invariants is the ambiguity occurring when a ray crosses the curve more than once. However, such multiple crossing did not occur for the model base used in our experiments. The vector of invariant line lengths $\mathbf{l}$ is not used directly as an index. Instead, an index vector $\mathbf{M}$ is constructed from $\mathbf{l}$



l = (0.339, 0.308, 0.301, 0.280, 0.288, 0.294, 0.313, 0.350, 0.385)

M = (-0.089, -0.181, -1.089, -0.169, 0.602, -0.183, -0.173)

Figure 9: *A set of n equally spaced rays are drawn from the point $(\frac{1}{2}, 0)$ so that they intersect the curve signature. The aim is to construct an n-dimensional length vector $\mathbf{l} = (l_1, \ldots, l_n)^T$, where l_i is the distance from the intersection point of the i^{th} ray to the point $(\frac{1}{2}, 0)$. This distance is projectively invariant. Here $n = 9$. The invariant index $\mathbf{M}$ is related to $\mathbf{l}$ by $\mathbf{M} = \mathtt{E}\ \mathbf{l}$, where $\mathtt{E}$ is provided by a linear classifier. See text for details.*

using a statistical classifier over all curves in the model base. There are two advantages of this: first, the index is more discriminating than the "raw" lengths; second, the dimension of the index can be reduced and so the computation of an efficient hashing function is simplified. The Fisher linear discriminant [Duda73], which is an optimal linear classifier, is used for the computation of the index. The discriminant encodes information by minimising the intra-class variance (that is over several examples of the same curve) and maximising the inter-class separation. It does so by transforming the data to a new (orthogonal) basis, $\mathbf{M} = \mathtt{E}\ \mathbf{l}$, such that feature measurement variance is maximised under projection onto some of the basis directions, and minimised onto others.

Each basis coordinate is ranked by how much discrimination it yields. Then, enough of the highest ranked coordinates are chosen to provide the desired separation between the classes. It was found that taking seven elements of the Fisher discriminant basis are sufficient to define and discriminate a projectively invariant description for each curve class. An example of the Fisher basis is shown in figure 10. A benefit of using the classifier for model learning is that an analytic understanding of the statistical characteristics of the invariant measures is not required. Instead, a number of examples of a single class is built up over a number of images, and the classifier adjusts its action to account for the variation within each class.

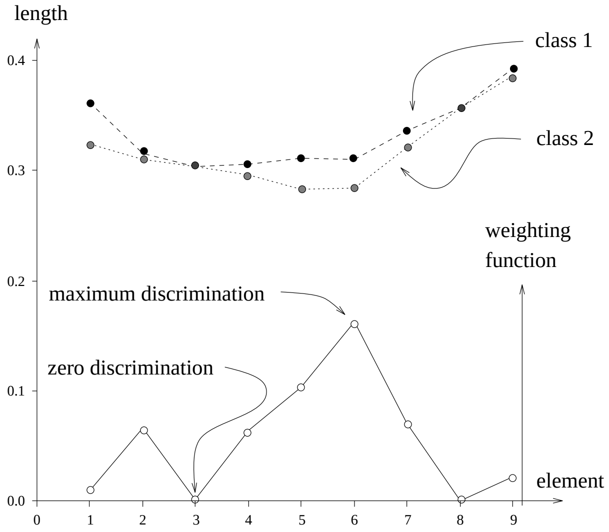


Figure 10: *Example of the Fisher Linear discriminant. The discriminant is trained here only three classes, two of which are shown (black-circles-dashed-line and grey-circles-dotted-line). In each case the curve is represented as a vector of canonical frame ray lengths, each component corresponding to a different angle (see figure 9). For each class a number of vectors, measured for the same curve with varying viewpoint, are included in order to model the intra-class variation. The first eigenvector weighting function produced by the discriminant is shown (white-circles-solid-line). The first invariant is determined as a scalar product of the eigenvector and the (mean) vector for each class. Clearly, this invariant will show good discrimination between the two classes shown.*

2.4 Segmentation and Grouping

LEWIS requires the segmentation of two different categories of features from image data: *lines* and *conics* for the algebraic invariants; and $\mathcal{M}$ *curves* (images curves terminated with common tangents) for the canonical frame construction. The features are grouped depending on the type of invariant that they form.

The first step in the feature detection process is edge detection. We have used an implementation of the Canny edge filter [Canny86]. The next process is segmentation, this can be broken down into three phases:

1. The extraction of discrete edgel chains from the image.
2. The location of breaks between features, and more generally the boundaries between each feature and other data.
3. The accurate representation of image features.

The first step is common for both the algebraic and smooth curve invariants. Single edge curves are extracted from the edge image using a sequential edgel chain linking. The Canny algorithm produces edges with sub-pixel accuracy. This edgel position accuracy yields invariant values with smaller variances (about 10% better) than those computed from integer pixel locations. The reason that using more precise edgel locations does

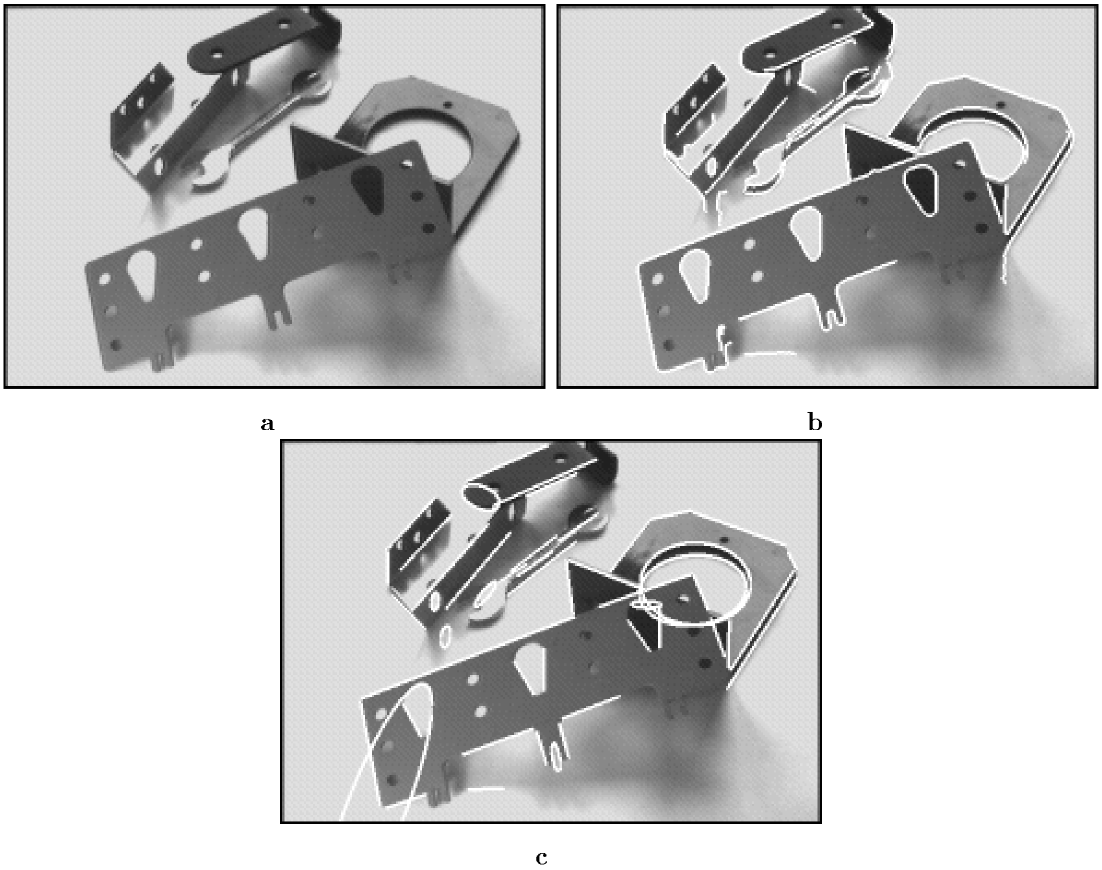


Figure 11: *(b) extracted edge data (Canny) from (a); note that the edge detector fails to locate edges near shadows and that objects (such as the bracket on the right hand side of the image) have a finite thickness and so two edges are reported. The fitted conics and lines are shown in (c) where there is generally accurate location of both tangency and curvature discontinuities.*

not produce such a dramatic improvement in the quality of the measured invariants is that the representation process (principally fitting) is able to smooth out quantisation errors present in the integer edgel locations.

Even with *hysteresis,* single pixel breaks can occur in the edge chains. Such events are accommodated by directional look ahead in a sequential scan of the edge chain. As the quality of the edge data in the images of interest is generally quite good, single pixel look-ahead works well for the objects and illumination conditions used in our experiments. The details of the later stages of the segmentation process depend on the type of invariant that is to be formed, and the different techniques are discussed below.

2.4.1 Algebraic Features

Lines and conics are fitted to extracted edge chains using efficient incremental routines based on orthogonal regression for lines and an improved version of the Bookstein algorithm for conic fitting [Bookstein79]. Full details of the algorithms are given in [Rothwell94]. An example segmentation is shown in figure 11 where it is seen that a reasonably complete description is obtained of the object boundaries.

2.4.2 Grouping

Exploiting structure in the scene for grouping allows invariant indexing to have a low complexity with respect to the number of image features. The approach used makes use of the connectivity provided by the edge chains, this implicitly encodes proximity.

For algebraic invariants, connectivity provides an association and also an ordering on the lines: invariants are formed from sets of consecutive lines within single edge chains at a cost that is linear in the number of lines in the scene, $O(l)$. This type of grouping was also exploited by Huttenlocher who also achieved linear grouping cost [Huttenlocher88].

The use of algebraic curve features rather than isolated points and lines also reduces the combinatorial cost of grouping. In the case of the invariant formed by a conic and three lines the cost of grouping is $O(cl^3)$, where c is the number of conics and l the number of lines. This is for a case in which no image structure is assumed, if connectivity is reliable, the cost reduces to $O(cl)$. The grouping cost for the joint conic invariants is only $O(c^2)$. For the images under consideration l is in the order of a hundred, and c a few tens.

2.4.3 The Canonical Frame

The canonical frame construction requires the accurate location of distinguished points. Stable point constructions are achieved using curve bitangents and points defined by *cast tangents*. In order to form a projective coordinate frame, the canonical frame, four such distinguished points must be found. It is desirable to achieve a canonical projection of the object boundary curve which is minimally distorted and has a roughly uniform variance distribution due to image segmentation effects. In practice, this is achieved by placing the points in the canonical frame in positions that correspond to a fronto-parallel view of the object [Rothwell94] (yielding an equiform distortion of the object).

Bitangent Location

Image bitangents are located using the following four stage algorithm:

- Eliminate points that lie on approximately straight portions of curve. These cannot correspond to actual points of bitangency and so should be ignored.
- Find points on the same edge curve that have approximately common tangents.
- Check that such pairs of points do in fact correspond to bitangents.
- Improve the localisation of the bitangent points using quadratic interpolation.

Straight portions of curve are found by fitting a straight line to short segments of the curve using orthogonal regression and testing the value of the fitting residual. Approximately straight portions will have a low residual. The next step is to map the curve into its tangent dual space and look for self-intersections of the

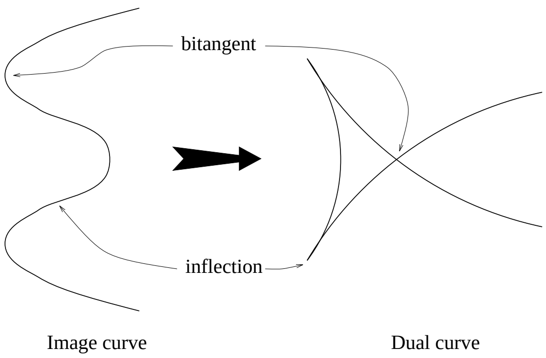


Figure 12: *For continuous curves bitangents in the image correspond to self-intersections in the tangent dual space. Likewise inflections correspond to cusps.*

dual curve. Bitangents, where a line is tangent to the curve at two points, correspond to self intersections in dual space as shown in figure 12. The mapping of a boundary curve into tangent dual space is based on the parameters of a running line fit to the curve. The fitted line is locally tangent at each point along the curve. The representation of the dual space for a curve is essentially the same as a Hough space for lines and is is parameterised by the slope, θ, of the local tangent, and the perpendicular distance of the tangent to the centre of the image.

The dual space is quantised into discrete cells of angle and distance. Since the image curve is discrete, at points of high curvature the difference in tangent direction can vary significantly between adjacent points. This quantisation problem is overcome by linearly interpolating between consecutive points in dual space.

Self intersections, and hence bitangents, are found using a voting scheme in the tangent parameter space. Two image points voting in the same quantised cell represent a self-intersection. Due to small curve fluctuations, joint cell occupancy does not always correspond to actual bitangents. False bitangents are detected by examining regions of the image curve in the proximity of bitangent points. Dual space provides the location of the bitangencies up to discrete pixel coordinates. Significant improvement in accuracy can be obtained by interpolating the bitangent locations between the actual measured edgel locations. A local quadratic fit determines the location of the bitangent points to sub-pixel accuracy. This is done by rotating the image data so that the initial estimate of the bitangent line is horizontal, and fitting (by regression) quadratics of the form $y = ax^2 + bx + c$ to data sets either side of the two bitangent points. The cost used is the error in the y direction. The interpolated bitangent is the line simultaneously tangent to both parabolas. In the implementation the number of points used for each quadratic fit is 13 (6 either side of the hypothesised bitangent point). The data sets are centrally weighted using a Gaussian. The weighting was set empirically by observing how the quality of canonical frame construction changed as the number of points was altered.

The bitangent detection scheme finds many bitangents along single image curves. This is demonstrated in figure 13. Due to excessive redundancy in the shape representation many of the bitangents can be eliminated from consideration, preferably those that are not stable:

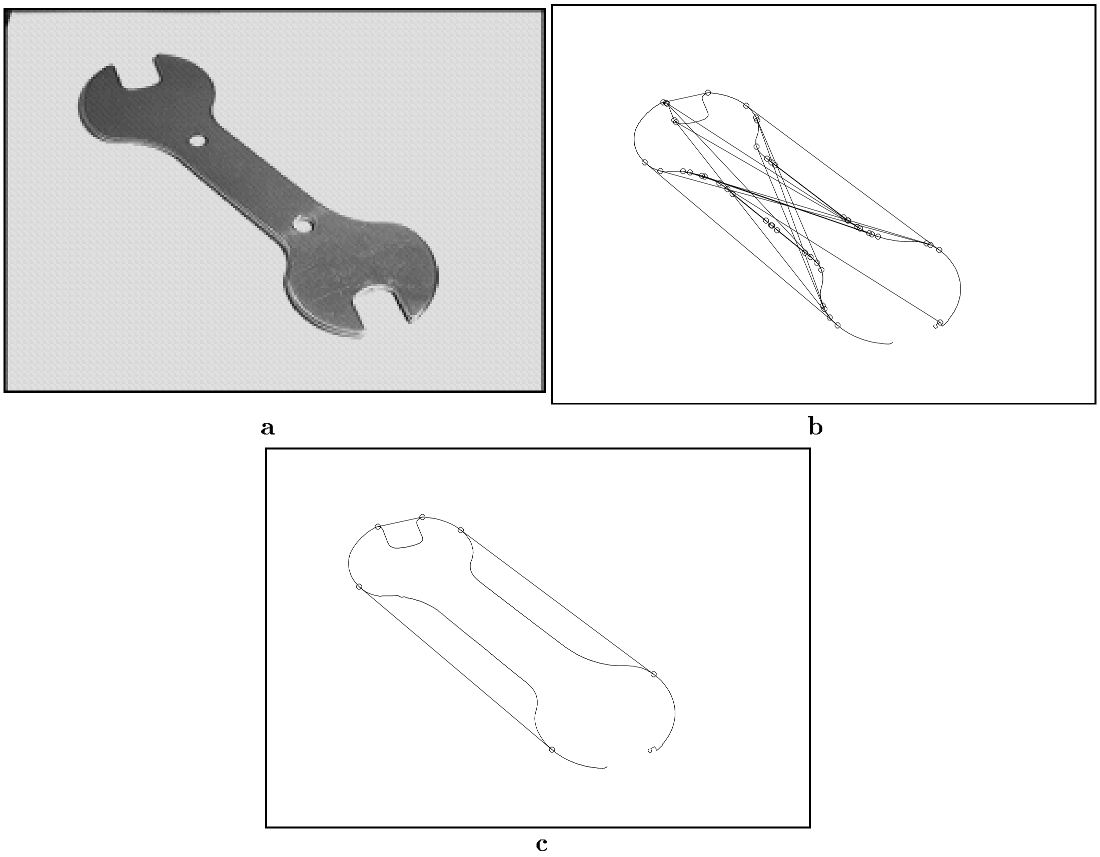


Figure 13: *In (b) it is shown that there are a large number of bitangents that can be found even for a simple object such as the spanner in (a). Each one enables the construction of a canonical frame curve, though only curves that do not cross their own bitangents are used. This reduces the level of redundancy. Bitangents that will not produce a stable construction are also deleted; this leaves the three bitangents shown in (c).*

- Eliminate any bitangents that have their endpoints too close together.
- Remove bitangents whose associated $\mathcal{M}$ curves are not very deep (only a few pixels).
- Do not use tangents that cross the image curves. These tangents will be stable, but eliminating such tangents leads to a simpler canonical frame signature.

Cast Tangents

A cast tangent is a ray from the bitangent point which is tangent to the $\mathcal{M}$ curve. The cast tangent is made unique by selecting the tangent ray making the largest angle with respect to the bitangent line. The construction is projectively invariant and cast tangents are found in a manner similar to that for the bitangent point, again, localisation of the contact point is improved by quadratic fitting.

A sample segmentation for a simple view of a spanner is given in figure 14, in which the bitangent and cast tangent points and lines are superimposed onto the object. The bitangent points bound the $\mathcal{M}$ curves

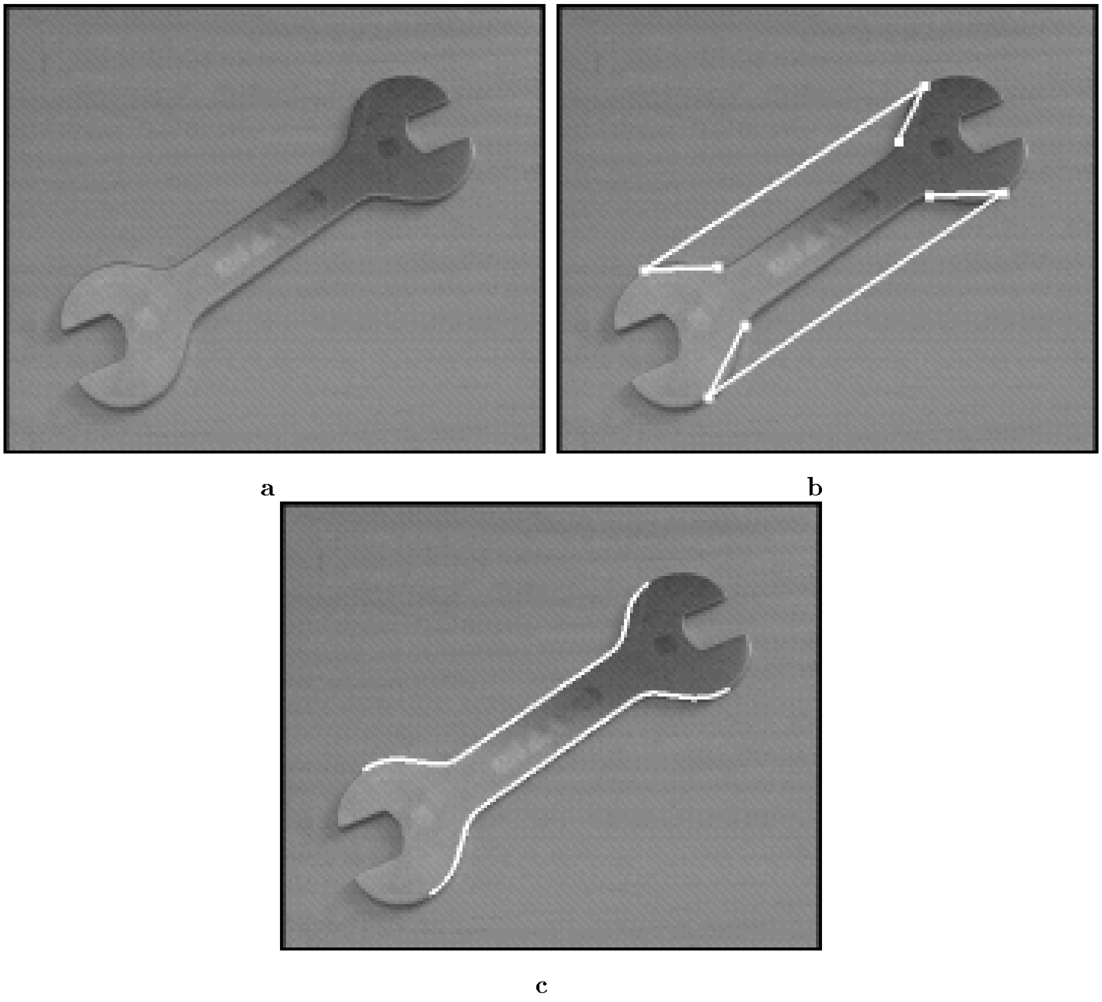


Figure 14: *For a simple object such as the spanner shown in (a) there are two reliable* $\mathcal{M}$ *curves that can be constructed. The bitangent and cast tangent points and lines are shown superimposed in (b). These are located to sub-pixel accuracy using the quadratic fitting method described in the text. The* $\mathcal{M}$ *curves bounded by their bitangent points are shown in (c). The* $\mathcal{M}$ *curves at the ends of the spanner are not used because their canonical frames cannot be determined stably.*

that are shown in (c).

Grouping

The canonical frame construction has a linear grouping cost. This is because all of the features used to form the frame are ordered around single image curves. This result is identical to that of [Huttenlocher88], and means that recognition using the construction is very efficient.

2.5 Errors in the Invariant Measurements

Before an indexing scheme is implemented, the error distribution of the invariant functions must be determined in order to determine whether a measured image index value is within an acceptable experimental

Table 4: *The mean values for the three invariants measured from the image sequence based on the images in figure 15. The standard deviation σ is computed both as an absolute value, and as a percentage of the mean. Note that the 3σ mark is well within 5% of the mean, and so such a bound could be used during recognition. During acquisition, we are more cautious, and use a tighter 3% bound on the allowable errors.*

	I_1	I_2	I_3
mean	(0.707,2.252)	(0.752,1.492)	(0.524,3.043)
σ	(0.0031,0.0170)	(0.0032,0.0086)	(0.0052,0.0433)
σ (%)	(0.44,0.76)	(0.43,0.58)	(0.99,1.42)

error bound of the actual model value. The rest of this section describes a pair of experimental investigations into the expected sizes of the invariant errors (one for algebraic invariants and one for canonical frame invariants). For the algebraic case the empirical investigation is compared to an analytic calculation.

2.5.1 Algebraic Invariant Errors

One can obtain a rough guide to the size of expected errors under *ideal* imaging conditions by differentiating the invariant expressions, and assuming an isotropic noise distribution; such analysis was done in [Forsyth91, Sinclair93], and is also given here. The short-comings of this type of formulation become apparent when real images are observed. The only way to understand the errors that may be encountered within a recognition system is to study real images. All theoretical analysis has to assume some error model in image measurements; frequently this is founded upon a Gaussian error in the locations of individual edge locations due to what is often called *image noise.* The results given below demonstrate that *errors* occur due to behaviour of standard edge detectors used, and cannot be attributed to random *noise.*

Empirical Investigation

The first and twenty-eighth images from the sequence used to do the tests are shown in figure 15. The rest of the sequence of fifty images were constructed by rotating the object at 2^o increments on the calibration table beneath the object. The lines fitted to the edge data, with the seven lines used to compute the invariants, are shown in figure 16. The direction of rotation used to form the sequence is also marked. Three different five-line invariants were computed for each image of the object using these lines. Note that the object is specular, and is on a black background that is also somewhat specular. While the images do not represent ideal imaging conditions, edge detection is expected to be quite reliable, since image step edge contrast is large over the entire boundary.

The mean invariant values for the image set are shown in table 4. These results show that the invariants are in fact very stable, with standard deviations less than 1.5% of the mean values. From these results the error measurements that are used during recognition and acquisition are chosen. For the former the aim is to eliminate as many false negatives as possible and so the error bound is high (that is 5%, which is well above the 3σ mark), but during acquisition one should be more cautious so that only stable invariants are used (and so 3% is used, roughly equal to 2σ).

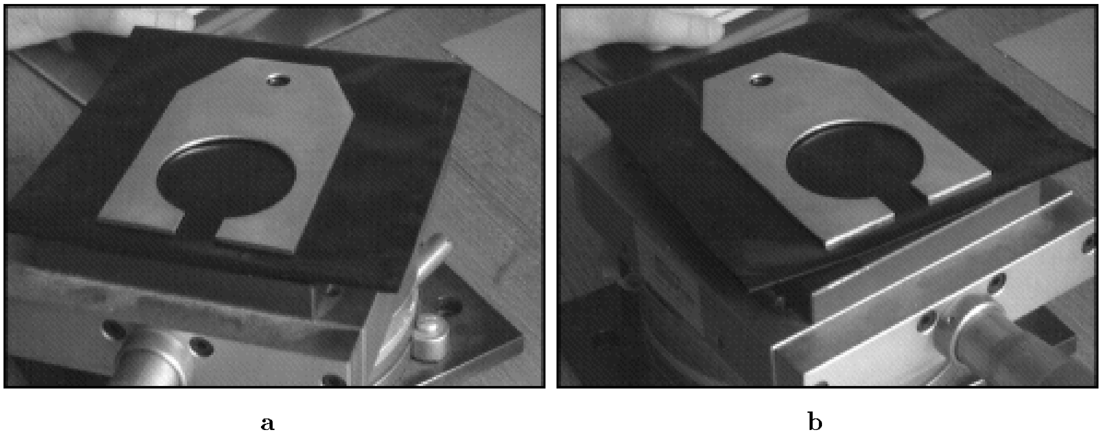


Figure 15: *The first and twenty-eighth image in the fifty image sequence used to test the reliability of the invariants. The rest of the sequence was produced by rotating the calibration table by 2^o between images. Three five-line invariants can be computed for this object using the seven longest lines. The twenty-eighth image is when line 3 (labelled on figure 16) becomes vertical and both lower and upper edges of the object are visible. From this viewpoint, the location of the edge boundary becomes ill defined.*

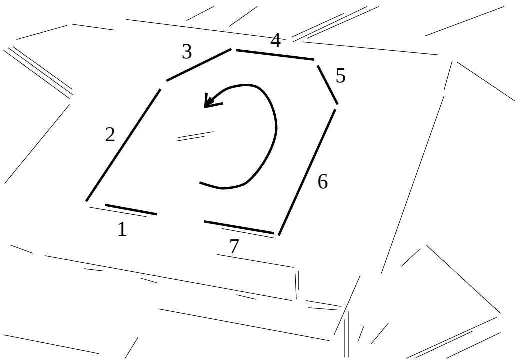


Figure 16: *The lines fitted to Canny edge data from figure 15a. The seven lines used to compute the invariants, and the direction of rotation, are marked.*

The value of I_2, computed on the sequence of lines 2 through 6 is plotted with an enlarged scale in figure 17. The shape of the graph is characteristic of all of the invariant constructions. The graph can be split into three distinct regions:

1. **Region A:** All of the lines are located reasonably well by the Canny edge detector, and so the measured invariants remain constant.

2. **Region B:** When the object has been rotated so that line 2 becomes vertical in the image, the edge on the lower surface parallel to it to becomes visible. The edge detector does not find a pronounced second edge in this orientation, but because of its presence the intensity values no longer form a step edge at the correct feature, but instead a slope. The Canny output locates a position somewhere along the slope and not at the top edge. The fitted line is therefore incorrect and causes the invariant value to be measured erroneously. Notice that as the object is rotated more, the invariant value tends to

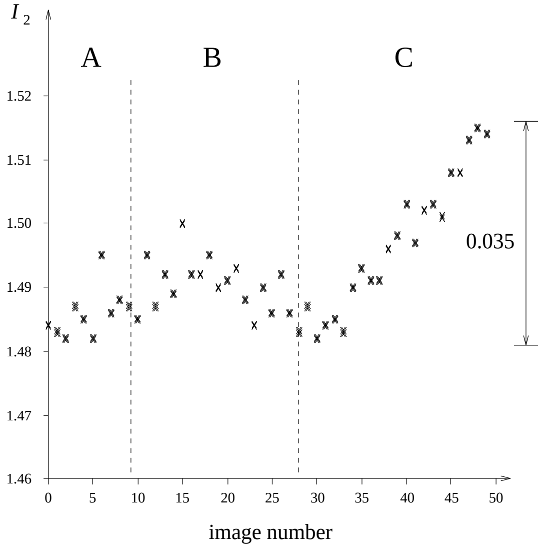


Figure 17: *When the invariants (in this case the second value of the second invariant) are plotted in greater detail, a systematic error becomes apparent in their measurement. This is due to the edge detector becoming distracted towards spurious image features, and is not due to image noise.*

decrease (though noisily), this is because the slope causes the fitted line to move further and further away from the correct edge.

3. **Region C:** In this region the effect is more pronounced as edge 3 moves through the vertical. When the fitted line drifts off the actual geometric edge, there is an obvious systematic error in the invariant measurement.

As can be seen from the graph the systematic errors produced by the edge detector far outweigh any Gaussian or quantisation noise observed in the points. Such noise will still be present, though its effects are small compared to other errors. It could be observed more clearly by removing the effects of the systematic error. Note that other unmodelled image events, such as shadows and close proximity of other objects, will also hinder the extraction of the planar geometric boundary.

Analytic Investigation

The gross effects of the systematic error can be estimated by perturbing the invariant expressions. Given the expression for the second invariant:

$$I_2 = \frac{|\mathtt{M}_{421}||\mathtt{M}_{532}|}{|\mathtt{M}_{432}||\mathtt{M}_{521}|},$$

the aim is to determine the effect of, say, the third line on I_2. If the lines used to evaluate the expression are of the form $\mathbf{l}_i = (a_i, b_i, 1)^T$, $i \in \{1, \ldots, 5\}$, then:

$$\frac{\partial I_2}{\partial a_3} \quad = \quad I_2 \; \frac{|\mathtt{M}_{245}|}{|\mathtt{M}_{532}|.|\mathtt{M}_{432}|} \; (b_2 - b_3),$$

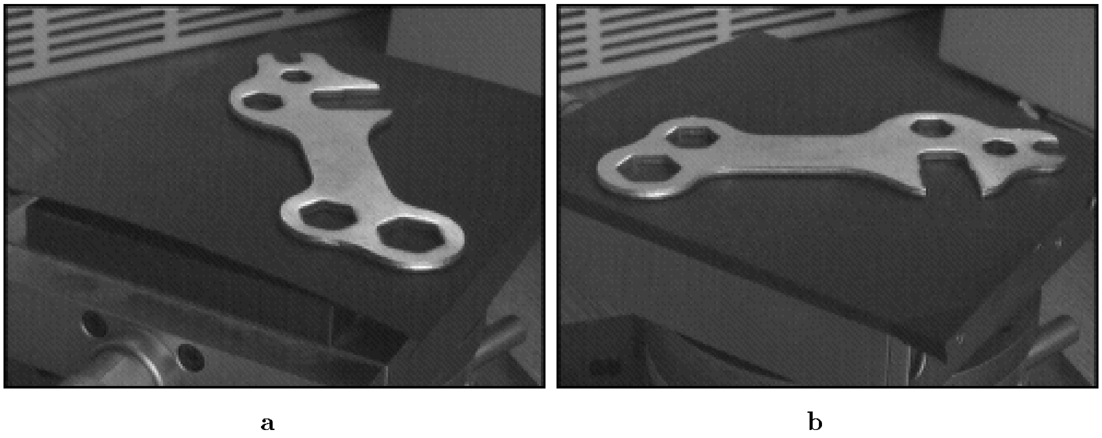


Figure 18: *The first and twentieth views of the spanner in the sequence are shown. The $\mathcal{M}$ curve of interest is the left-most one in (a), and the distant one in (b).*

$$\frac{\partial I_2}{\partial b_3} = I_2 \frac{|\mathtt{M}_{245}|}{|\mathtt{M}_{532}|.|\mathtt{M}_{432}|}(a_3 - a_2). \tag{6}$$

The model for the error observed in the measurement of the fitted lines is a translation parallel to the line normal by δ. If the gradient of the line is $\tan\theta$, by letting $\alpha = \delta / \cos\theta$ the equation of the perturbed line is:

$$\frac{a_3}{1 - \alpha b_3}x + \frac{b_3}{1 - \alpha b_3} + 1 = 0.$$

This directly yields $(\partial a_3/\partial\delta, \partial b_3/\partial\delta)$, from which δ can be estimated given a known ΔI_2.

From region C of figure 17 the value of ΔI_2 can be estimated as 0.035. This is assumed to be due entirely to the movement of $\mathbf{l}_3$ and that all the other lines are measured correctly. Applying the analysis yields a value of $\delta = 2$ pixels for this ΔI_2; this certainly is of the right order of magnitude for the error in fit observed in the image sequence.

2.5.2 Canonical Frame Invariant Errors

An empirical experiment similar to that for the five line invariant has been done for an object for which canonical frame invariants can be computed. Two images from the sequence used to measure the invariants are shown in figure 18. The value of the first invariant measured for each image is plotted in figure 19 against spanner orientation, which is varied through 180^o in 5^o increments. Note that the value of the invariant is stable, but again a systematic error is apparent when the graph is observed in more detail.

3 Recognition

3.1 Overview

An outline of LEWIS is shown in figure 20.

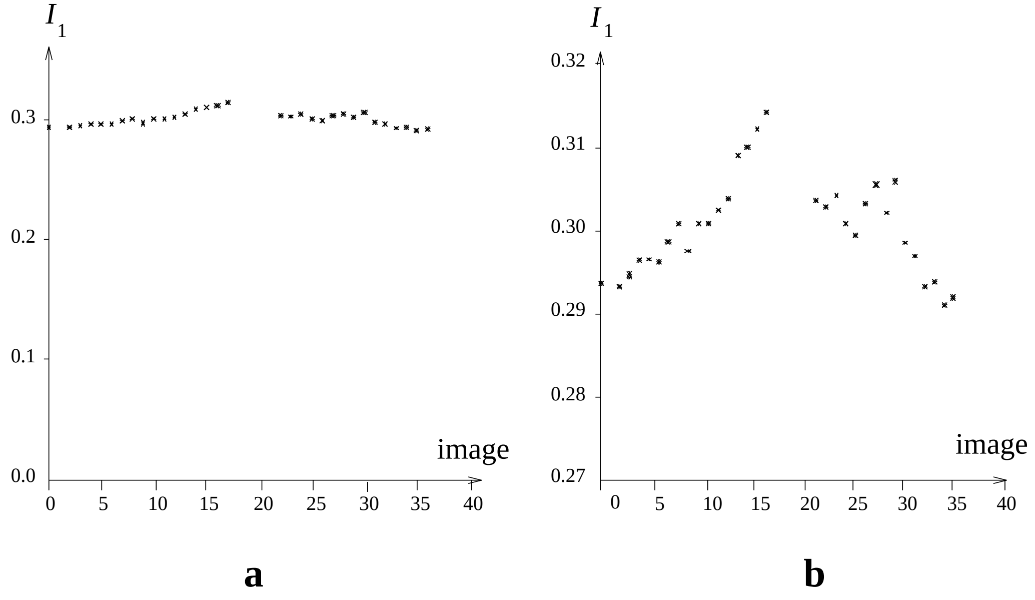


Figure 19: *The first invariant measured for the image sequence in figure 18 is plotted as the spanner is rotated by 5^o between images. Note in (b) that the edge detector produces a systematic error due to the it being distracted by the finite thickness of the object.*

3.1.1 Feature extraction and Invariant Formation

The goal of feature extraction is the formation of geometric primitives suitable for constructing invariants. In the algebraic case this involves straight lines and conics, and for non-algebraic curves, $\mathcal{M}$ curves delineated by bitangents. The fitting and grouping processes were described in section 2.4.

Once sets of grouped features, $\mathbf{F}$, have been produced, the invariants listed in sections 2.2 and 2.3 are computed. Each set of grouped features, or $\mathcal{M}$ curve, produces a number of invariants (one or more) which form a vector[2] $\mathbf{M}(\mathbf{F})$. Of course, if the object is occluded to the extent that the number of features visible is insufficient for an invariant, then no index can be formed.

The invariant vector formed by the above process (when quantised), represents a point in the multidimensional invariant space. Each object feature group is represented by a collection of points that define a region in the invariant space, the size of which depends upon the measured variance in the invariant value (see section 2.5).

3.1.2 Indexing

The invariant values computed from the target image are used to index against invariant values in the library. If the value is in the library, a preliminary recognition hypothesis is generated for the corresponding object.

[2] At this stage in the processing, each feature group is used to form a separate $\mathbf{M}$ vector. Interactions between the groups are only considered later.

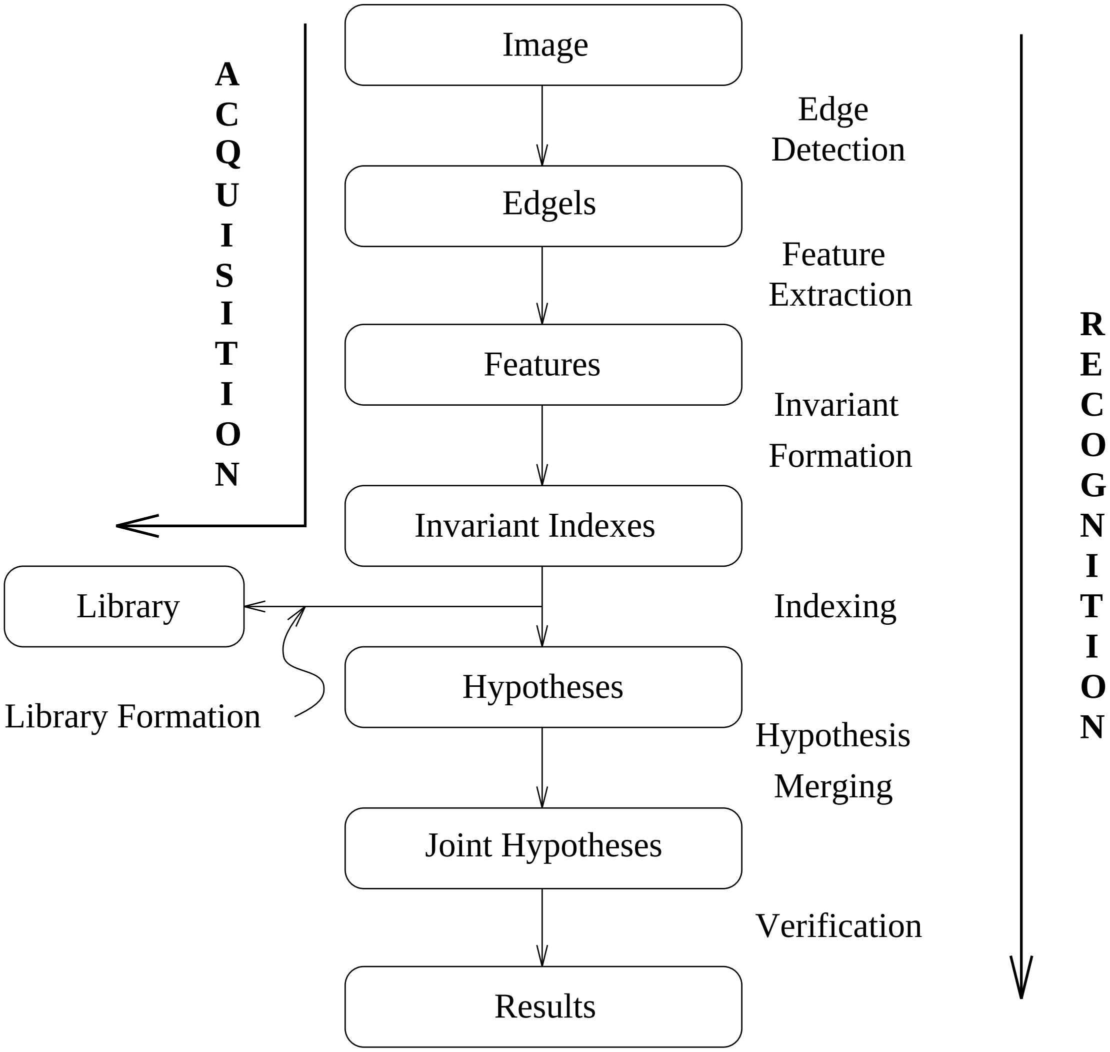


Figure 20: *LEWIS has a single greyscale image as input and the outputs are verified hypotheses with associated confidence values. Many of the processes are shared by the acquisition and the recognition paths. The recognition system is similar to previous systems in all but the indexing and hypothesis formation stages [Grimson90].*

Each type of invariant (for instance that for five lines, or a conic pair) generate separate hypotheses.

This process is made more efficient using a hash table that allows simultaneous indexing on all elements of the measurement vector. In the experiments to date there has not been any significant problem with collisions in the hash table. Hash table collisions[3] should not be confused with the intersection of object invariant measurements in index space. These intersections lead to erroneous hypotheses which cost some effort during the verification stage, but are usually eliminated.

3.1.3 Hypothesis Merging

Because many invariants may actually correspond to the same object, and should therefore be covered by a single recognition hypothesis, *joint hypotheses* are formed prior to recognition by combining 'compatible' hypotheses. There are number of reasons why hypothesis merging is desirable:

1. Backprojection and searching for image support (verification) is computationally expensive and it is more efficient to validate several hypotheses of the same object together.

2. More features facilitate a more accurate least squares calculation of the back projection transformation (there are more matched model and image features), and consequently a reduced error in measuring image support.

3. Many hypotheses indexing the same object in a single part of the scene significantly increase confidence that the match is correct.

During hypotheses merging, an interpretation tree is constructed for each object. The features used in the tree are the groups of invariant features that were successful in indexing. The merging process utilises topology and geometric compatibility. The topological consistency (ordering and connectedness) is illustrated in figure 21. Geometric consistency is implemented efficiently by a second use of invariants; this time joint invariants between the feature groups used to compute each individual hypothesis. This is illustrated in figure 22.

Since topological relations are often unreliable it is possible that two hypotheses could be united into a single joint hypothesis even though they are totally unrelated (for example one may represent a correct match and the other may have been caused by clutter). A list of all the original hypotheses and all possible combinations of compatible hypotheses is therefore maintained. The list is ordered by descending number of simple hypotheses per joint hypothesis. Those with more simple hypotheses are verified first, and if the match is confirmed, other joint hypotheses that represent partial versions of the hypothesis are deleted. If the match is not confirmed only the joint hypothesis under consideration is deleted. The joint hypothesis formation

[3] A hash table collision occurs when a number of models have the same hash index. Such a collision can occur when the number of hash buckets is smaller than the model population or when the hashing function is not uniform and causes many models to hash to the same bucket.

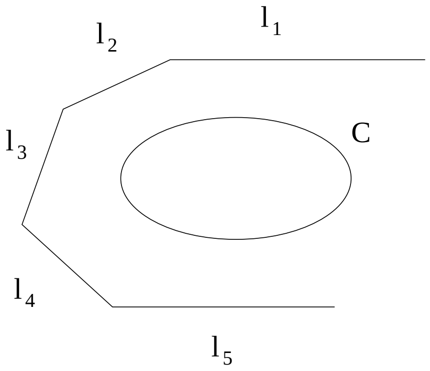


Figure 21: *If the same model is indexed by a five-line invariant (due to lines* $\mathbf{l}_i$*,* $i \in \{1, \ldots, 5\}$*), and a conic three-line invariant that is* compatible *with it (due to* C *and* $\mathbf{l}_i$*,* $i \in \{2, \ldots, 4\}$*), then it is wise to verify both hypotheses together. The invariants are compatible if the ordering of the image lines are consistent with those on the model; see text for details.*

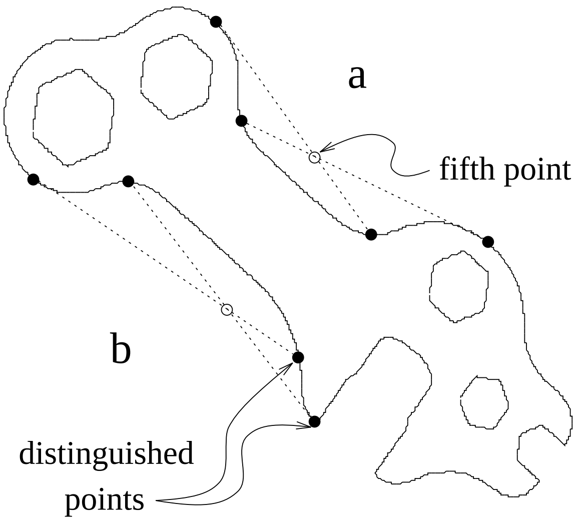


Figure 22: *For a pair of* $\mathcal{M}$ *curves there are 8 distinguished points which could be used to form* $2 \times 8 - 8 = 8$ *different five point invariants. Rather than computing so many, which is unnecessary, invariants are computed between the four distinguished points of each* $\mathcal{M}$ *curve, and the 'central' point of the other. This yields four invariants, and does so using a symmetric construction. These invariants are sufficient to hypothesise compatibility.*

stage can potentially cause an exponential number of hypotheses to be formed. However, in practice, deleting verified hypotheses keeps the verification process under control (as is shown later in table 6).

3.1.4 Verification

There are two steps involved in verification, both of which can reject a (joint) recognition hypothesis. The first is an attempt to compute a common projective transformation between the model features and the putative corresponding features in the target image. The second is to use this transformation to project the entire model onto the target image, and then measure image support.

Incorrect hypotheses arise because grouped image features happen to have an invariant value that coincides (within the error bounds) with one in the library. Also, because the invariants are not complete (completeness is defined in detail in [Rothwell94]), structures with the same invariant may not be projectively equivalent. The features used to produce the matching model and image invariants provide sufficient constraints to compute the projective transformation between the model and image[4]. In general, the projective transformation is over determined as the feature groups tend to provide more than the required eight constraints. Consequently, if a common transformation cannot be computed, the features are not projectively equivalent and the hypothesis is rejected [Rothwell94].

Backprojection and subsequent searching involves the entire model boundary, not just the features used to form the invariant. Projected model edgels must lies close to image edgels with similar orientation (within 5 pixels and 15^0). In the case of algebraic features, two preliminary hypothesis filtering steps can be invoked:

1. The model lines must project to within 15^o of the image lines.

2. The projected model conics must project to ellipses, and they must have similar circumferences and areas to the image conics.

Orientation in the target image is determined from the Canny edgel orientation. The orientation of the projected model feature is determined as follows:

1. For model edgels on straight lines the projected orientation of the line is used for each edgel.

2. For model edgels on conics, the orientation is obtained from the projected conic via their polars[5] which are close approximations to the tangents for edgels close to the conic.

3. For other edgels, the orientation provided by the edge operator is used. This orientation is determined in the target image by projecting the tangent line to the model. Model edgel orientations are less accurate (than a fitted line or conic), so a 30^o threshold is used instead of 15^o.

[4] Unless the invariant exploits an isotropy. In this case, certain parameters of the transformation are unrecoverable because they do not affect the projected geometry, e.g. a circle under rotation about its center.

[5] For a point $\mathtt{x}$ and a conic $\mathtt{C}$, the polar $\mathbf{l}$ of the point with respect to the conic is $\mathbf{l} = \mathtt{C}\mathbf{x}$.

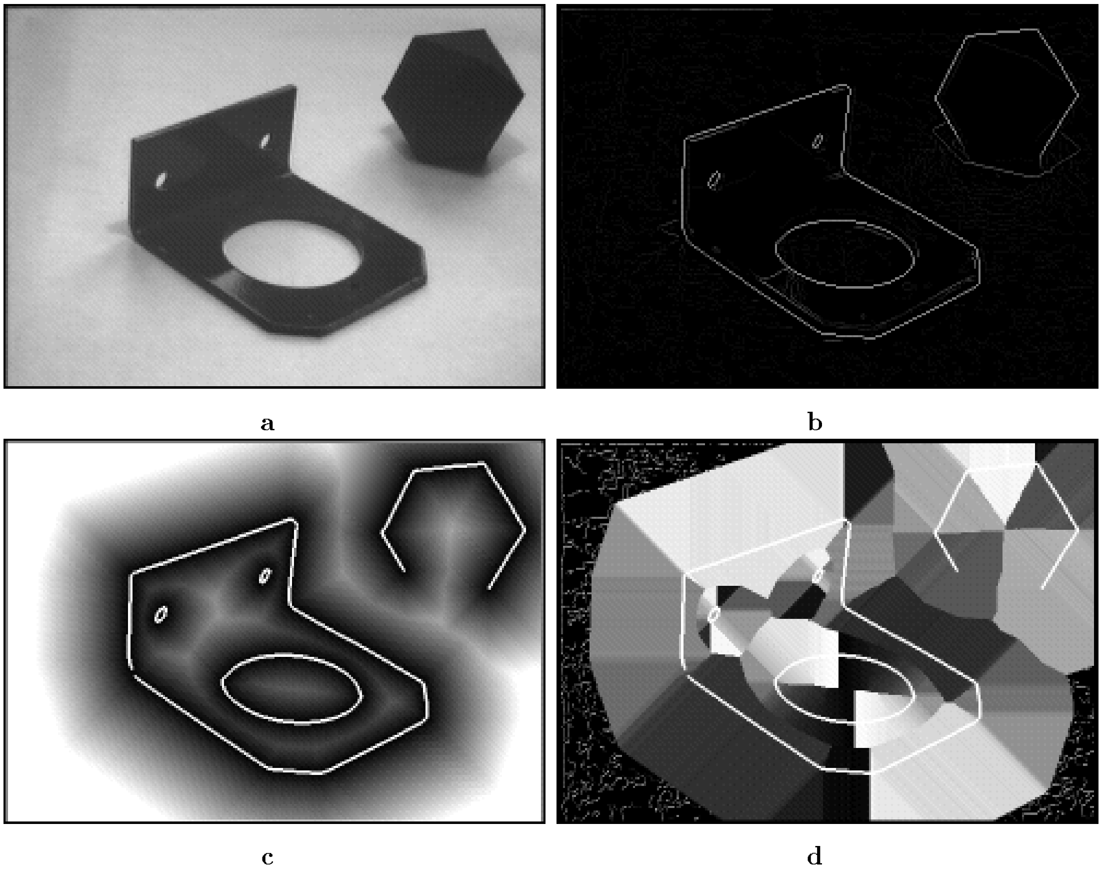


Figure 23: *(a) shows a simple scene with two objects in it. The output of the Canny edge detector is shown in (b). The 3-4 distance transform is computed for all edges over a certain strength and is displayed in (c); distance from the edge (white) is coded by intensity, with zero being black. (d) shows, coded by intensity, the orientation of the the edge nearest to a given image point.*

If more than a certain proportion of the projected model data is supported (the threshold used is 50%), there is sufficient support for the model, and the recognition hypothesis is confirmed. The final part of the process is expensive as $O(10^3)$ edgels need to be mapped onto the image. Efficiency in the distance computation is achieved by approximating the distance using the 3-4 distance transform of Borgefors [Borgefors88]. The distance transform is found by passing chamfer masks over the image, which is carried out within image preprocessing. An example of the 3-4 distance transform output for a simple image is shown in figure 23.

If the projected model is too small in the scene it must have arisen from an object so far away that it would not be observed reliably. An upper bound on the size of the projected model can be computed by bounding the model by a box and projecting that to the image first; if it is too small, then the hypothesised object must be too small and so can be rejected. In practice the bounding box used is the perimeter of the acquisition image.

There is a trade off involved in setting the support threshold. A heavily occluded correct match may have as much support as an incorrect match. Particularly if there is dense edge data (such as wood texture), then it is quite likely that a large number of edges may be close to, and have the same orientation as, the projected model edges. In a structured scene, a few erroneous straight lines of the right orientation will be sufficient to give over 50% support for a model, and so render a false positive. Obviously, any object which is over 50% occluded will not be found by the recogniser. As the threshold is lowered, an occluded object is more likely to be found, but there will also be more false positives. On the other hand, if more than one invariant forms a hypothesis that passes verification, the level of confidence in the result is high. This is discussed further in section 4.1.

3.2 Model Acquisition and Library Formation

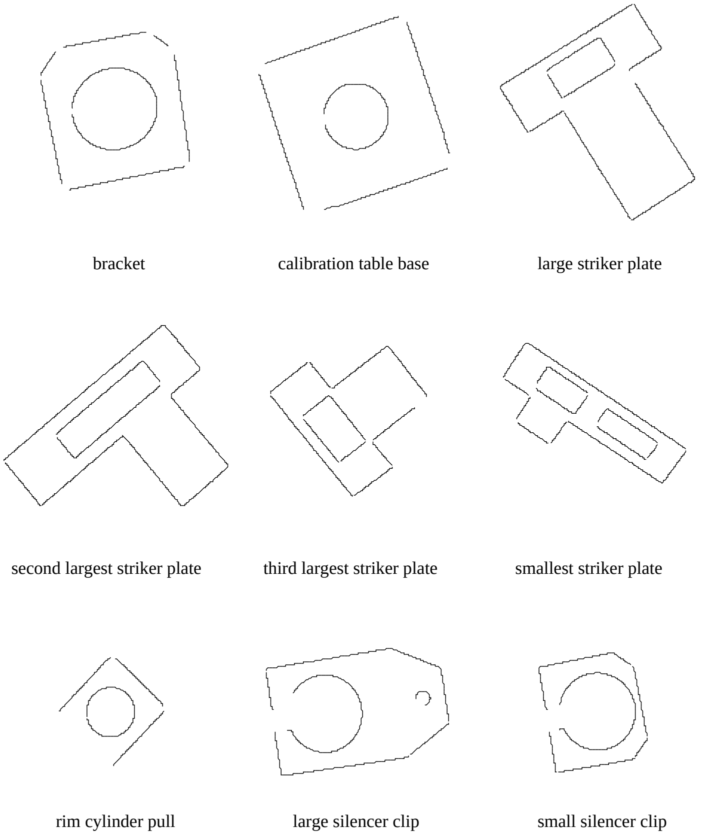
A model consists of the following:

1. A name.
2. A set of edge data from an acquisition view of the object for use in the backprojection stage of verification.
3. The lines, conics and $\mathcal{M}$ curves that represent the edge data.
4. The expected invariant values and which algebraic features or $\mathcal{M}$ curves they correspond to.
5. The bounding box of the model features.
6. Topological connectivity relations between feature groups that will be used in the construction of joint invariants.

The library is segmented into different sub-libraries, one for each type of invariant. Each sub-library has a list of each of the invariant values tagged with an object name, and is structured as a hash table.

One benefit of using only projective representations, rather than Euclidean, is that model acquisition can be done directly from images. No special orientations or calibrations are required. Acquisition is simple and semi-automatic (for instance, curves do not have to be matched by hand). It proceeds as follows:

1. A number of images are taken of the isolated object from a variety of 'standard' viewpoints (for algebraic invariants two images are used, generally for the canonical frame system more are required to compute the Fisher discriminant).
2. The invariants are computed for each view. This involves the same segmentation and invariant computation as used during recognition. For non-algebraic curves *significant* $\mathcal{M}$ curves are extracted.
3. The invariant values are compared between views. The useful invariant shape descriptors will remain reasonably constant under a change in viewpoint. These are recorded in the modelbase. Any measures that are not constant are due to features that do not form correct invariant configurations (for instance lines that are not coplanar), or are caused by unstable features. For matching values (within 3%, see section 2.5), the mean value is entered into the model library.



bracket

calibration table base

large striker plate

second largest striker plate

third largest striker plate

smallest striker plate

rim cylinder pull

large silencer clip

small silencer clip

Figure 24: Nine of the models in LEWIS's model base (the edge data of the models is shown).

4. Connectivity between algebraic features or $\mathcal{M}$ curves is utilised to form joint invariants to be used during hypothesis combination.

3.3 Algebraic Invariants Examples

The results reported here have been carried out with a model library containing over thirty objects. Typical algebraic objects in the library are shown in figure 24. Recognition accuracy is excellent if the object boundaries are not severely disrupted by shadows and specularities. On a SPARC IPX, edge detection takes 15 seconds; feature extraction 5 seconds; matching less than a second; and verification normally about 2 seconds.

The first recognition example is a bracket in a scene with occlusion and clutter caused by other objects (figure 25). All possible algebraic invariants are formed from configurations of lines and conics. The measured and the matching values are given in table 5. From a scene such as this, a large number of possible invariants can be derived. It was found that two image five-line invariants matched model invariants of the bracket, with the second (incorrect) one ruled out during verification. Three conic-and-three-line invariants were measured in the scene that matched the invariants of the bracket, and all these constituted correct matches.

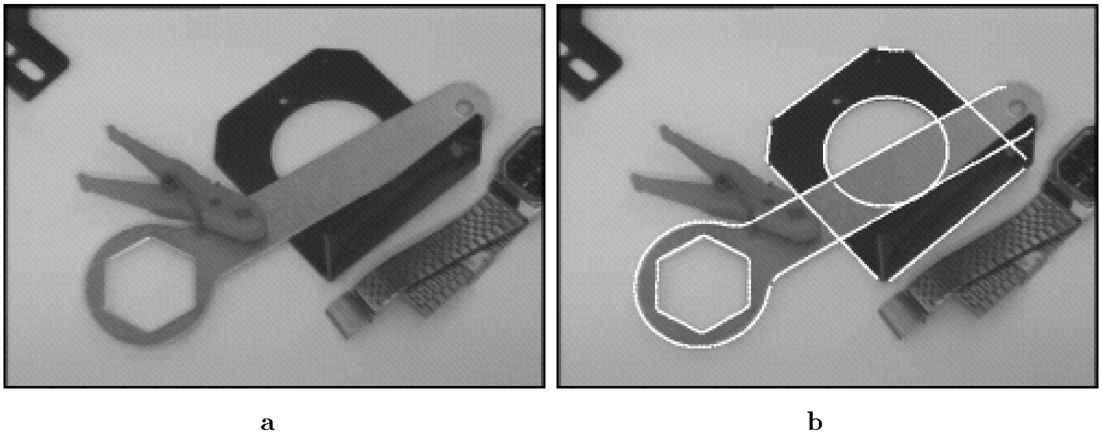


Figure 25: *(a) shows the bracket occluded in a scene. Some of the occlusion is due to an object not in the model library. In (b) the edge data from the first calibration scene are shown projected onto the test scene using the model to image transformation hypothesised by the match. The close match between the projected data (shown in white), and the scene edges shows that the recognition hypothesis is valid. Projected edge data from the model of a spanner are also shown projected into the scene as this was also recognised.*

Table 5: *The invariants measured from figure 25 which are formed by features actually corresponding to bracket features. The second column shows the library values and the third column scene values. In the fourth column the deviations from the mean invariant values are given; this shows that the five-line invariant is very stable under real image conditions, and the conic-and-three-line invariant is reasonably stable.*

invariant	library	scene	error %
five-line	(0.8415,1.2340)	(0.842,1.235)	(0.1,0.1)
conic-line	(1.3410,1.3080,2.6285)	(1.372,1.291,2.676)	(2.3,1.3,1.8)
conic-line	(1.3080,1.3025,1.8850)	(1.291,1.287,1.852)	(1.3,1.2,1.8)
conic-line	(1.3025,1.3395,2.5915)	(1.287,1.365,2.604)	(1.2,1.9,0.5)

Incorrectly indexed hypotheses can be ruled out during verification when the hypothesised model is projected into the scene (all such hypotheses were ruled out in this case). For the bracket, 74.5% of the projected edges match to within 5 pixels and 15^o of the image data. There is a second object from the model base, a spanner, also in the scene. This is correctly identified using three different invariants. In this case an 84.5% projected edge match is achieved with the model data also shown in white in figure 25.

In figure 26 the bracket is recognised despite a significant amount of occlusion (in this case there is only a 59.3% edge match during verification). Figures 27 to 45 show the system operating on a few test scenes with some of the match statistics shown. For figure 27, 1049 invariants were computed which indexed 41 hypotheses. These were converted into 131 joint hypotheses that had to be verified, of which 13 were rejected by first stage verification, based on valid projective transformations, and 78 required the second stage, based on image support. For figure 28, 806 invariants indexed 36 hypotheses, forming 44 joint hypotheses of which 23 needed the second verification stage after 13 were rejected by the first stage.

In table 6 various match statistics are shown that have been taken from a number of scenes (around 100) similar to that of figure 26. In each case, a single object from the model library was in the scene, and it was recognised correctly in all except one instance which was when verification broke down due to

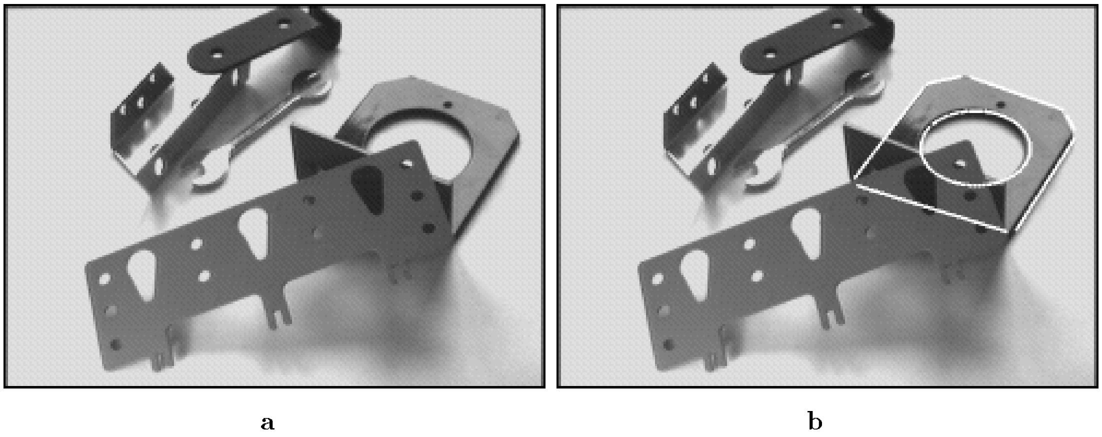


Figure 26: *(a) shows the bracket occluded in a scene by objects not in the library. In (b) the edge data from an acquisition scene are shown projected onto the test scene using the model to image transformation hypothesised by the match. The close match between the projected data (shown in white), and the scene edges shows that the recognition hypothesis is valid.*

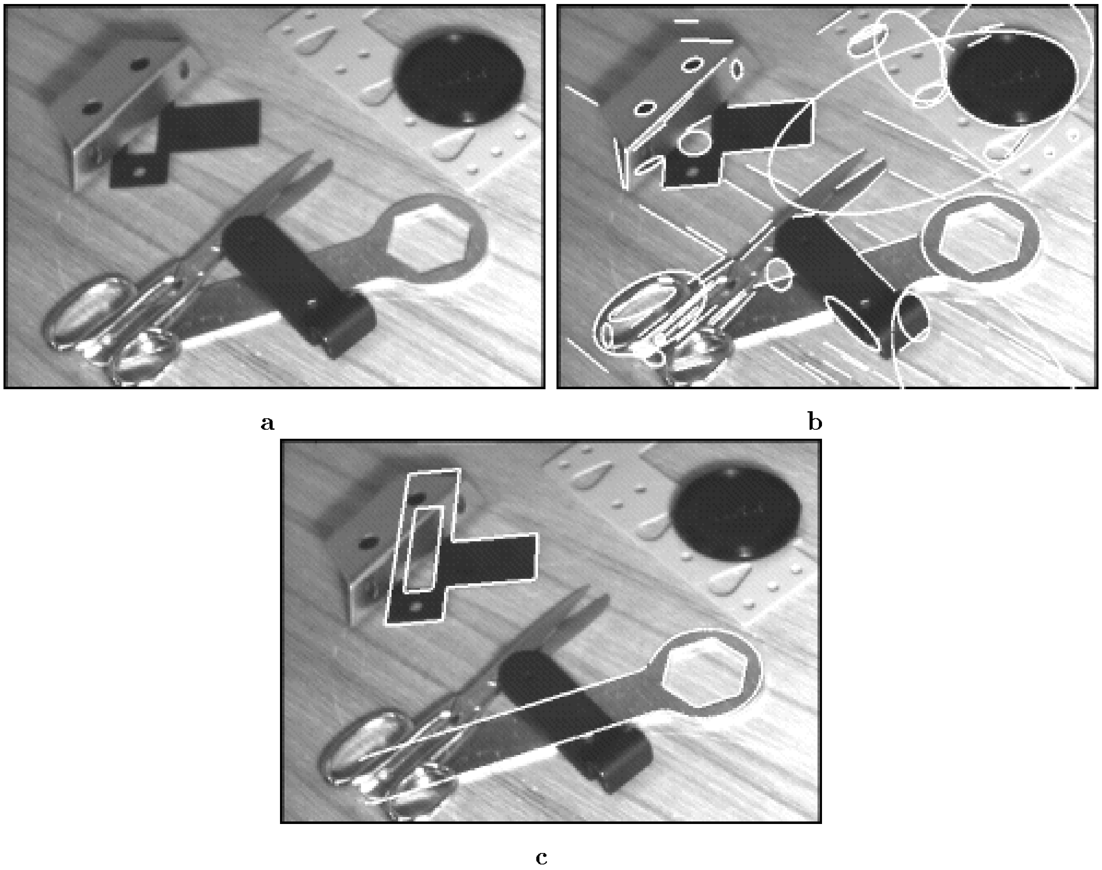


Figure 27: *(a) shows a scene containing two objects from the model base, with fitted lines (100 of them) and conics (27) superimposed in (b). Note that many lines are caused by texture, and that some of the conics correspond to edge data over only a small section. The lines form 70 different line groups. (c) shows the two objects correctly recognised, the lock striker plate matched with a single invariant and* 50.9% *edge match, and the spanner with three invariants and* 70.7% *edge match.*

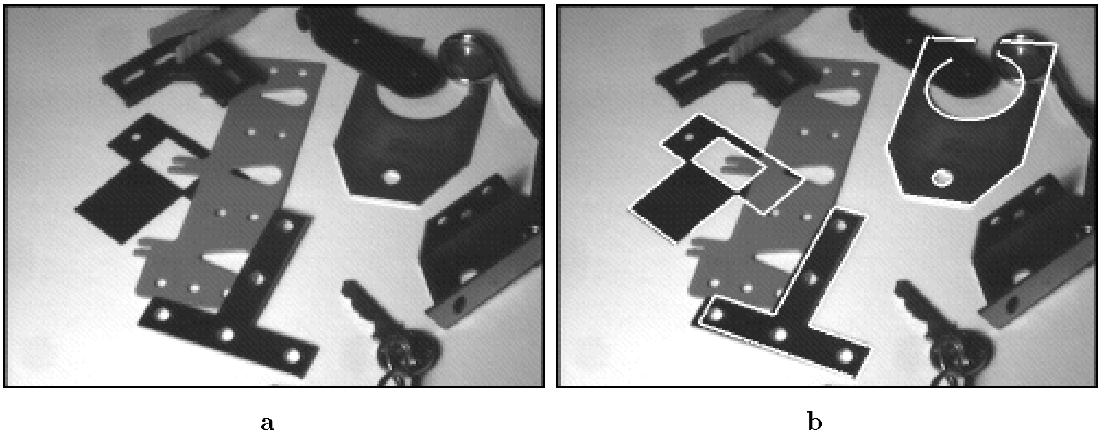


Figure 28: *Another typical scene containing three objects from the model base. The recognised objects are outlined with* 74.7% *(2 invariants),* 84.6% *(1 invariant) and* 69.9% *(3 invariants) edge matches for the objects from left to right. 58 lines and 14 conics were found.*

a poor segmentation preventing a sufficient amount of edge support. The total number of indexes formed (an average of 1755.3), depends solely on the number of features in the scene, and the way in which they are grouped. This number roughly equates to the number of hypotheses that would have to be verified per model for a hypothesise and test technique. After indexing, these form only an average of 60.4 hypotheses, which constitutes a nearly thirty-fold reduction. Because of redundancy in the shape representation, multiple hypotheses can occur for a single model instance. Joint hypothesis formation processing yields an average of 72.7 joint hypotheses.

Verification is performed once the joint hypotheses have been constructed. On average, 5.9 hypotheses do not have to be verified as their structures are subsumed by larger joint hypotheses. This means that only 66.8 joint hypotheses actually require verification, which is similar to the original 60.4 individual hypotheses. It is clear that the joint hypothesis formation stage does not lead to an exponential number of hypothesis being formed, and yet it provides improved recognition. Once the projectivity between the hypothesised models and the image features has been computed, a check is made that the projected and image algebraic features are consistent (the preliminary filter). On average this filter removes 41.1 hypotheses, 6.4 due to line correspondences, 4.9 for conic and line configurations, and 29.8 for conic configurations. In the end, only 23.7 hypotheses have to be verified through full back projection, compared with the 1755.3 original indexes formed.

In each case a single object should have been recognised. Essentially, a negligible number of false negatives are observed. One false positive, on the average, is successfully verified in a given image in addition to the correct model hypothesis. The false positive is partly due to the symmetry of an object, where the projected boundary can achieve good support from the set of image features, even though the correspondences and object pose are incorrect (such as in figure 42). False positives also occur due to confusion between projectively similar objects, that is, the projective transformation generates large shape equivalence classes.

Table 6: *The average match statistics for using algebraic invariants within LEWIS taken over a large number of images. See the text for an explanation.*

number of actual model instances	1.0
total number of indexes formed	1755.3
total number of individual hypotheses	60.4
total number of joint hypotheses	72.7
number not requiring any verification	5.9
number rejected by algebraic test	41.1
number rejected through full back projection	23.7
number of correct hypotheses	1.0
number of false positives	1.0
number of false negatives	0.0

3.4 Canonical Frame Invariants Examples

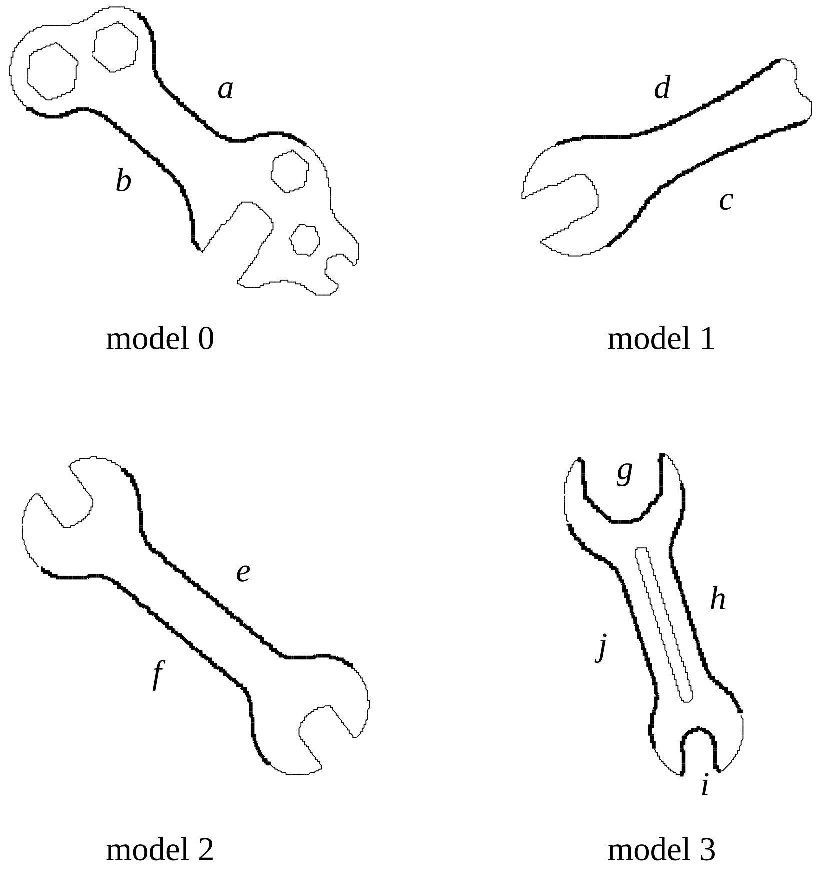
3.4.1 Classes

Typical (non-algebraic) objects in the library are shown in figure 29. The object $\mathcal{M}$ curves are sufficiently similar (in all cases there are only two inflections) to allow a grouping of the library into a number of *classes*, see figure 30. Indexing is then *hierarchical*: first, *sub-parts* (classes) are indexed and verified. For the class verification, rather than backproject the whole model curve, the $\mathcal{M}$ curve alone is projected into the canonical frame. It is verified by measuring the difference in areas between the image class and the model class curves (computed using rectangular quantisation in the canonical frame). If the difference is sufficiently small, the hypothesis is accepted. This covers the non-completeness of the canonical frame invariants. Second, if the class is accepted, hypotheses are generated for each of the models in that class. Joint hypotheses are then formed and verified by back projection to the target image using the entire boundary model curve.

The efficiency of the indexing process can be demonstrated empirically: from a series of typical images an average of 56 $\mathcal{M}$ curves were observed; 27.8% of these produced class hypotheses on indexing; and 23.9% of these were verified as classes (only 6.6% of the original number of $\mathcal{M}$ curves). Note that although a large number of classes were hypothesised in the scene, only 14.0% of the indexed hypotheses were later found to be incorrect. Based on these preliminary results (56 $\mathcal{M}$ curves found in image, 10 model curves, no false positives) it would seem that there is not an excessive tendency towards false positives.

3.4.2 Recognition Examples

The first example shown in figure 31 shows a simple unoccluded view of model 0. This object can be recognised using up to two classes. First, the classification algorithm locates classes 0 and 1 as marked in figure 31b, and uses these to form a single joint hypothesis by the procedure of section 3.1.3. The joint hypothesis is verified through backprojection in which 92.8% of the model outline is matched to image data. A 100% confidence is not found (as would be expected for an unoccluded object) because the Canny edge detector fails to extract and localise all of the object edges correctly. This results mainly from specularity on the object. The same effect can be observed in all of the images in this section because the objects are



class	0	1	2	3	4	5	6	7
M-curve	*a*	*b*	*c*	*d*	*e,f,h*	*g*	*i*	*j*

Figure 29: *For the model base consisting of the four spanners there are ten useful $\mathcal{M}$ curves. These are shown by thick lines and labeled (a) to (i). Due to the projective similarity of (e), (f) and (h), eight classes are sufficient to represent the local shapes of the spanners. The correspondence between the $\mathcal{M}$ curves and the classes is given in the table. The global shape of each spanner is also required for recognition, this includes the geometric constraints between each class (see section 3.1.3 for details), and also the entire set of edge locations and orientation data; this is used for verification.*

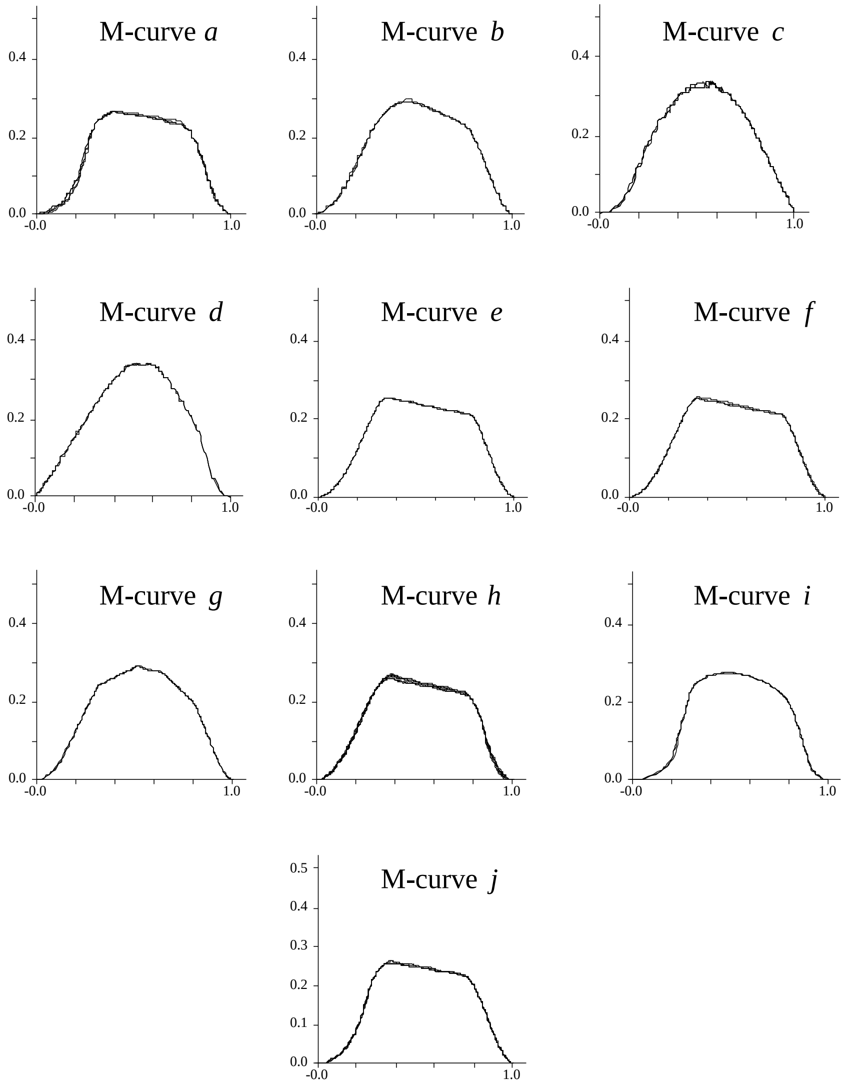


Figure 30: *The canonical curves for the four models shown in figure 29. Three images of each object were used and the curves superimposed; the very close match between each curve highlights the stability of the construction. Note the similarities between signatures (e), (f) and (h); these are essentially the same and are therefore represented by the same class. All the other signatures are in their own class.*

Table 7: *Matching statistics for figures 34 to 37. The number of $\mathcal{M}$ curves extracted from the images and how many class hypotheses result from indexing are shown. The class hypotheses are used to form joint hypotheses that are verified or rejected by the following tests: if a larger subsuming joint hypothesis has already been accepted; if a good model to image projectivity cannot be computed; if backprojection results in an impossible pose.*

Figure	#$\mathcal{M}$ curves	#classes	#jh	#no verification	#poor proj.	#poor pose
34	42	13	18	0	1	5
36	79	18	23	2	0	3
37	99	24	39	4	1	2

metallic. Another cause of edge segmentation failure is due to the finite thickness of the objects, as discussed in section 2.5.1. Frequently, an edge extracted from the image can swap between portions of the outline on the upper and lower surfaces of an object. As the canonical construction is local this does not present a major problem, though its effects are occasionally noticeable.

In figure 32 the recognition system is tested on a more complex scene where there is clutter and occlusion. A single class is found for model 3 (class 5), which is then localised correctly in the image to give 55.5% edge support. Although a total of 16 class hypotheses were formed, yielding 22 joint hypotheses, only the correct hypothesis was given sufficient confidence by backprojection (over 50% projected edge support).

The canonical frame construction works very well under significant perspective distortion. This is demonstrated in figure 33. For this relatively simple scene three classes are found, and only one produces a hypothesis that passes through the object verification procedure. This gives an 83.6% edge match. As may be seen from figure 29, within the range of typical signature variation, model 2 (which is the one identified in figure 33) is projectively 2-cyclic[6]. Thus, the spanner will always be projected into the image in two different poses differing by the equivalent of a 180^o rotation, and still match correctly.

Figures 34 to 37 show further recognition examples in which the correct objects are always recognised. No false positives were found in any of the images, though this is not always the case. Although some instances have sufficient edge support the hypothesis is rejected based on size, as described in section 3.1.4. For example, model 0 was identified as subsequently rejected as shown in figure 35. Full details of the recognition performance are given in table 7.

The algebraic invariant and canonical frame representations can be independently applied to an image to recognise objects of both types. Figure 38 shows an example of recognition for both index methods together.

3.5 Complexity

The grouping cost incurred in forming the invariants was discussed in section 2.4. Here we first propose a simple model for recognition complexity, and then verify this experimentally.

[6] An object is projectively 2-cyclic if there is a view of the object for which it can be mapped onto itself with a 180^o rotation.

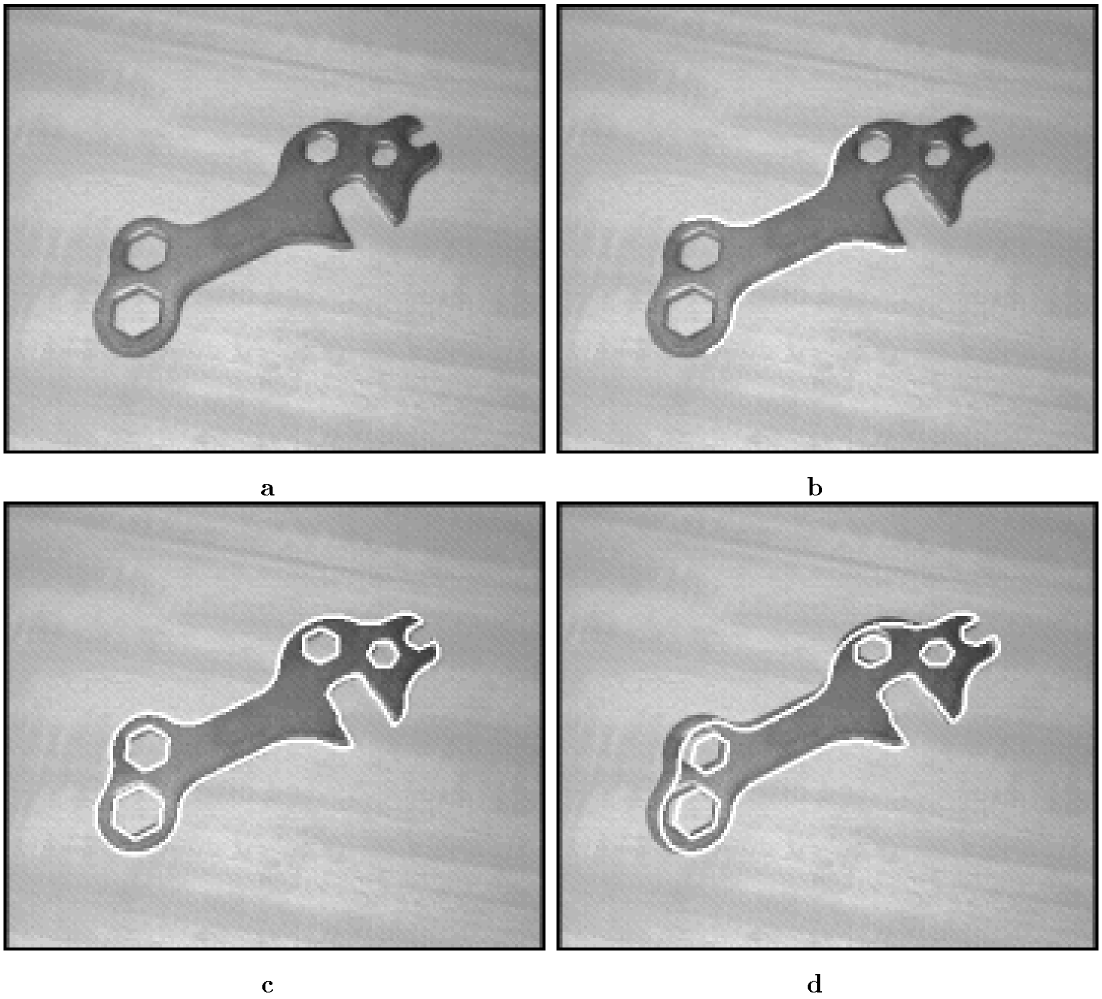


Figure 31: *(a) shows an unoccluded view of model 0. The classifier correctly locates classes 0 and 1 in (b). These are used to form a joint hypothesis for model 0 which is verified using back projection that finds* 92.8% *image support for the model. This is the only model match found that has a reasonable pose (that is, the object is not too small). Note that very good registration of the object is achieved in (c), this is when both* $\mathcal{M}$ *curves are used to compute the model to image transformation. Sometimes, as in (d), if a single* $\mathcal{M}$ *curve is used the registration is good in the region of the curve, but extrapolates poorly over the rest of the object. In this case a single* $\mathcal{M}$ *curve is still sufficient for recognition as a* 68.9% *projected edge match was found.*

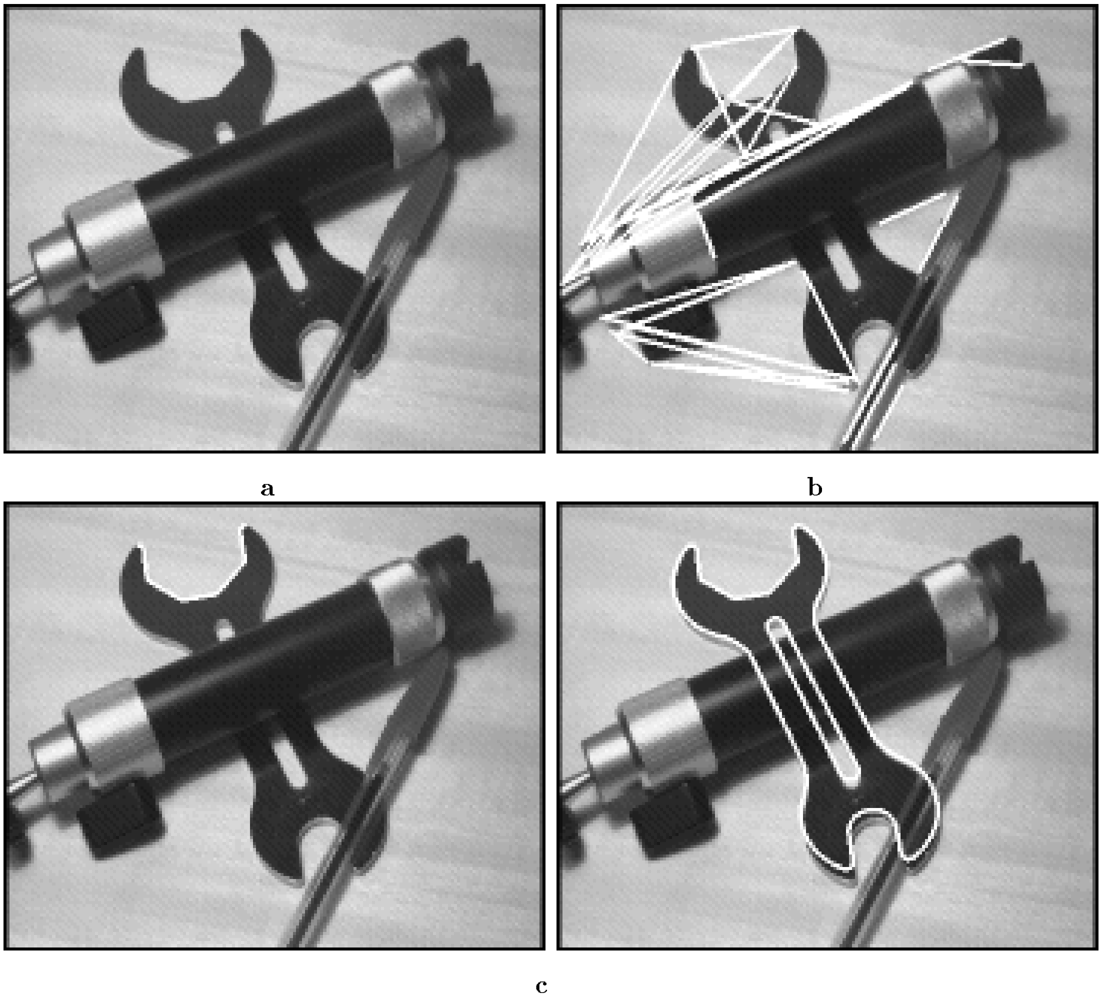


Figure 32: *For the occluded and cluttered view of the spanner (a), there are a large number of bitangents, (b). Note that bitangents are computed only along single continuous edgel chains and not between distinct curves. This further ensures a linear grouping cost. After $\mathcal{M}$ curve formation and indexing, a total of 16 potential class matches were found. However, the one that correctly identifies model 3, marked in (c), was the only one that produced a sufficiently high verification score (55.5%) to be accepted.*

3.5.1 Complexity Model

A major concern with the effectiveness of an indexing function is the probability that an image measurement taken from background clutter actually indexes a model. Often, it is suggested that the number of clashes produced within the hash table is important, but this is not the case. The hash table is simply an implementation of the index space, and should be designed so that only objects with matching image measurements are returned rather than those having only matching hash key values[7].

Here an informal argument is given that determines the likelihood that a random measurement will index an actual model; it shows that the indexing paradigm is (non-asymptotically) constant time, or at least can be made so with judicious use of the indexes. Consider a measure for a set of features that forms an n

[7] The function mapping index values onto hash keys is many-to-one.

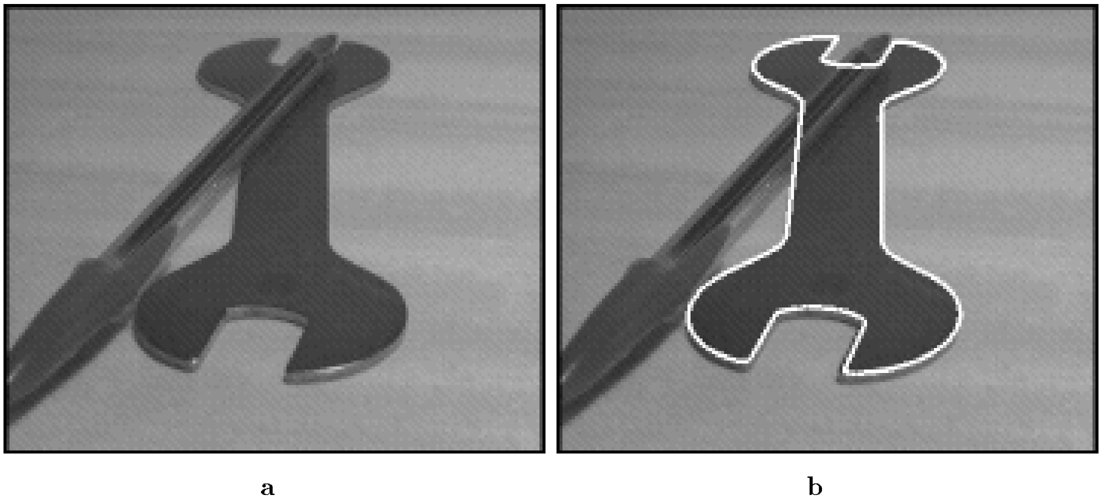


Figure 33: *Even under severe perspective distortion the recognition system performs well and finds model 2 with* 83.6% *confidence. Note that an affine description, such as the footprints in [Lamdan88], would fail in this case.*

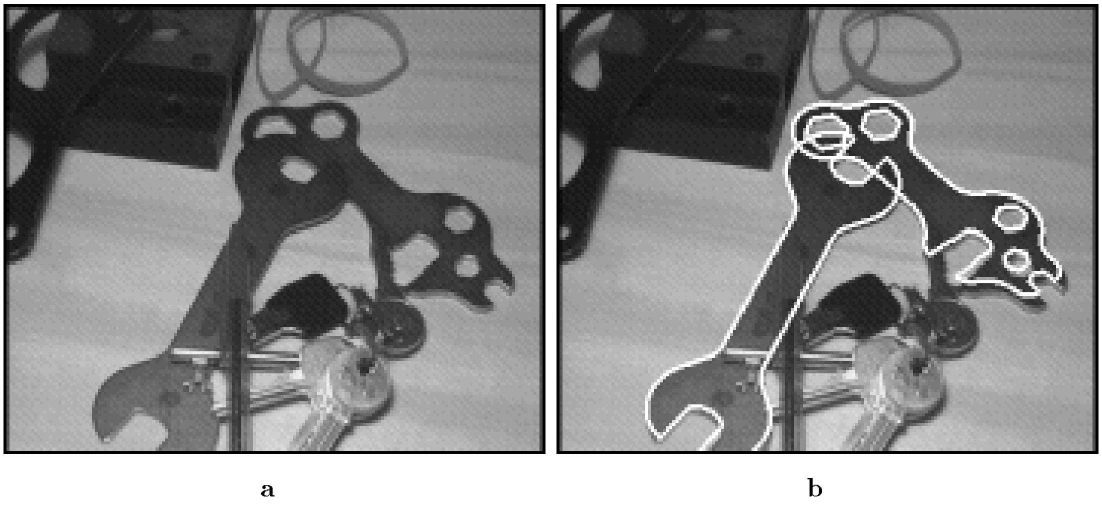


Figure 34: *Single classes are sufficient to recognise the two model instances shown in (b). The redundancy of the canonical frame representation gives much better tolerance to occlusion than global shape methods. The left hand object gained* 67.1% *boundary support, and the right object* 81.6%.

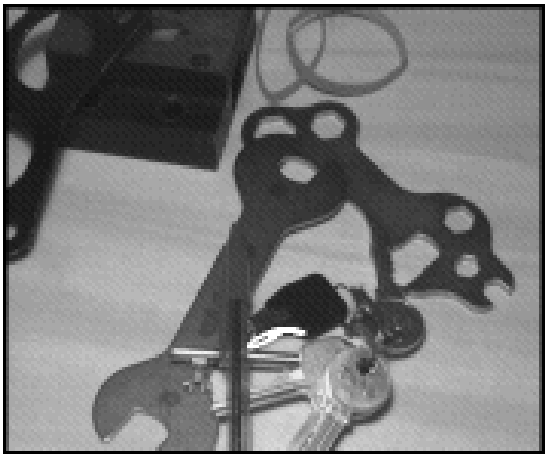


Figure 35: *In figure 34 incorrect objects were identified. Here, model 0 receives* 77.9% *edge match, but has a total projected object width of 24.7 pixels, which is too small to be a reasonable object projection.*

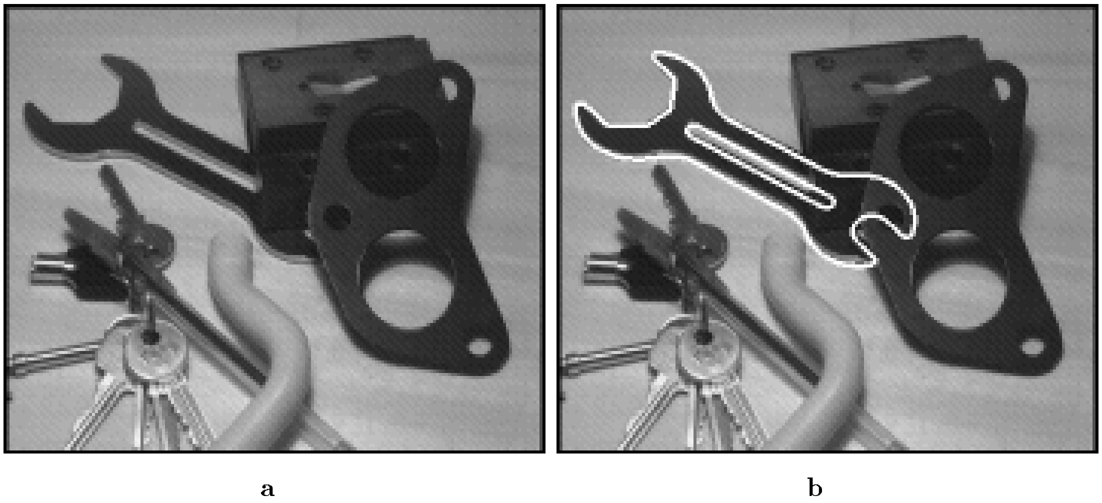


Figure 36: *Two classes are recognised and joined into a single joint hypothesis to recognise model 3 with* 68.0% *edge support.*

dimensional index; assume that each dimension has the same behaviour. Let each index cover a segment on the real line from i_0 to $i_0 + L$ (figure 39), and the quantisation along the line be δ, a constant quantity over the line segment[8]. There are $b = L/\delta$ buckets along the line, and so for n indexes and *assuming* that the measured invariants have a constant PDF over the invariant space[9], the probability of hitting any cell at random is $1/b^n$. If there are λ models in the library, each with α shape descriptors, and each invariant can be measured up to an error of $\pm\delta\epsilon/2$, $\epsilon \in \mathcal{N}$ (the set of natural numbers), there will be $\alpha\epsilon\lambda$ entries in the table[10]. If it is assumed that these entries are spread uniformly over the hash table, the chances of indexing

[8] More exactly a logarithmic scale should be used as the errors in invariant indexes tend to be proportional to the invariant values [Forsyth91].

[9] This claim is a current topic of research, and should be compared to the work of Hopcroft, *et al.* [Mundy92] and Maybank [Maybank93].

[10] For efficiency reasons during recognition only a single cell will be read. Models are not stored in single cells, but in as many as defined by the range $\delta\epsilon$ which is the expected measurement error. This contrasts with storing models in single cells and then

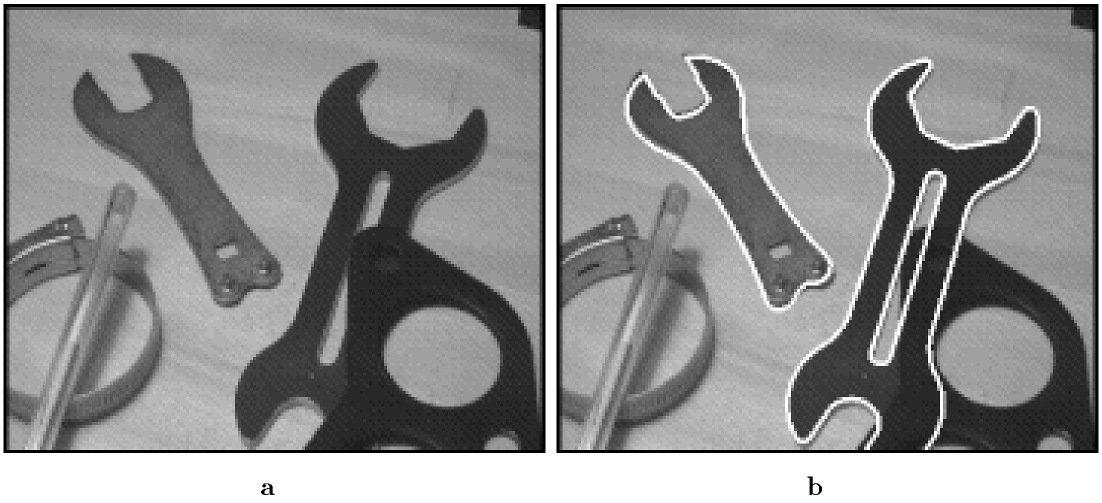


Figure 37: *Both models 1 (*91.4% *support) and 3 (*75.7%*) are correctly recognised and projected into the image as shown in (b).*

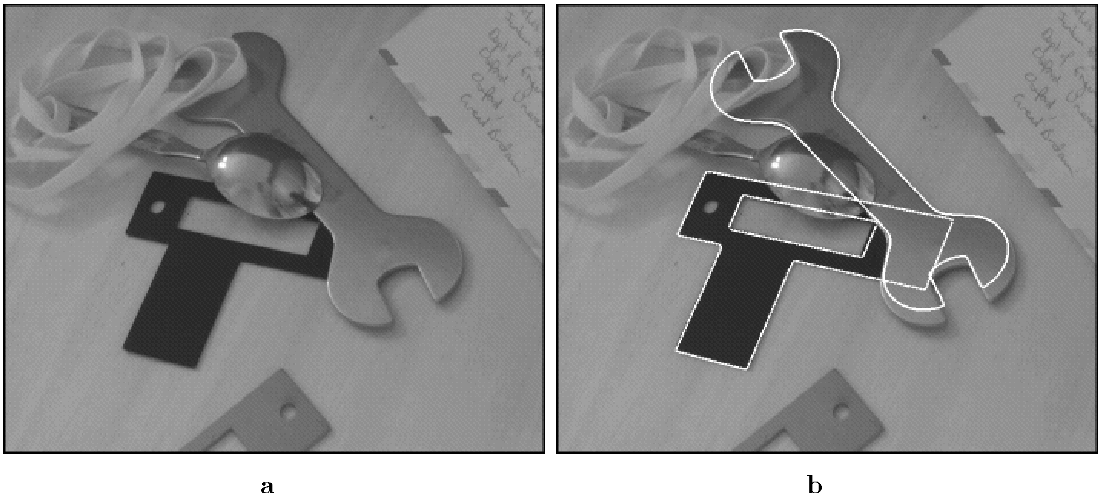


Figure 38: *A demonstration that both types of invariant index can be used to recognise objects in a single image (by applying the invariant constructions independently within LEWIS). The bracket is indexed using algebraic invariants and the spanner is indexed using the canonical frame signature.*

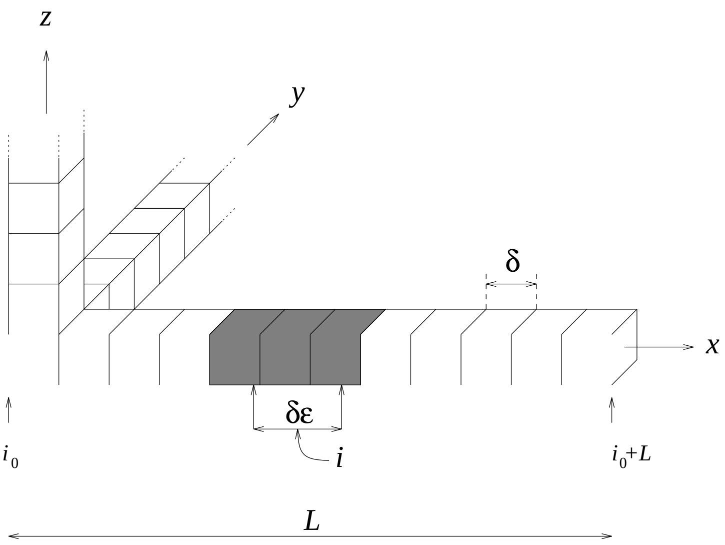


Figure 39: *Although the index space is multidimensional, it can loosely be assumed to be isotropic in all directions. Considering only the x direction: the index space ranges from i_0 to i_0+L, with bucket quantisation of size δ. On measurement of an index i in a scene, all the buckets within a range $\pm\delta\epsilon/2$ must be searched in the index space. These cells are shaded grey.*

a model through noise is $(\alpha\epsilon\lambda)/b^n$.

This analysis means that there is an algorithmic complexity of $O(k_1 + k_2\alpha\epsilon\lambda/b^n)$, where k_1 is the cost of edge detection, feature extraction and grouping (essentially constant), and k_2 another constant dependent on the form of the invariants, etc. It can be seen immediately that by making n large, the term dependent on the number of models λ, becomes arbitrarily small, and so recognition time tends towards k_1, a constant.

There are two problems associated with making n large:

1. For algebraic invariants there is little control over n. If a minimal feature group is used there is no control, but by using larger structures n can be increased. However, the grouping task may then become harder. Alternatively one could index using less discriminatory invariants and then group using results of this first indexing stage before forming higher order invariants and indexing a second time. For the invariants of other structures, such as canonical frame invariants, n can be made large (subject to the noise present in the curve).

2. Making n too large means that the problem of constructing an efficient hashing function must be considered.

During the development of LEWIS [Rothwell94] it was found that an invariant composed of a conic and two lines gave insufficient discrimination between objects. However, as an example of the above argument, when

indexing over a range.

an extra line was used to make $n = 3$ rather than $n = 1$ the invariant increased in utility. Because of the grouping heuristics used in the system there was no loss of efficiency but rather a marked improvement in performance.

3.5.2 Empirical Assessment

The indexing technique computes a number of invariants that is entirely dependent on the number of image features, though only a few of these will be turned into hypotheses on indexing. Indexing dramatically reduces the time taken for the entire recognition process. It was argued above that there should be a small linear growth in the number of hypotheses created as the size of the model base grows.

The linear growth is demonstrated in figure 40. The graph shows data collected over fifty evaluations of the recognition system in which a single model from the model base was placed in a scene and partially occluded by other objects that are not in the model base. Other non-library objects were also placed in the scene as clutter; figure 26 shows a typical scene. The average number of hypotheses computed as more objects were added to the library is plotted. The first model added to the library always corresponded to the actual model in the scene. Although 15.8% of the hypotheses were for the correct model (this is for when a total of 33 objects were present in the library), as predicted by the theory, the shape of the graph is predominately linear. The real benefit of indexing becomes apparent when one considers how many hypotheses would be produced if an alignment technique were used (maintaining the same grouping methods). On average, over 2000 feature groups existed for each image, and so 2000 hypotheses would be produced for each model feature group in the library (generally there are four or five feature groups per object and so the situation would be far worse). This would result in about 7×10^4 hypotheses for the entire model base compared to less than the 60 produced when indexing is used. As these all have to be verified it is clear that indexing produces a dramatic improvement in the system efficiency.

4 Discussion

We have shown how the use of invariants as index functions avoids search at two stages of the recognition process. First, indexes generate hypotheses which give direct access to models, avoiding a search through the model library. Second, at the hypothesis combination stage, invariants of the geometric relationships between feature groups, for instance a pair of $\mathcal{M}$ curves, permit the efficient construction of extended feature groups.

4.1 Verification

The final stage of recognition in most model-based systems [Huttenlocher87, Lowe87] is to verify model-to-image hypotheses. In the system described here, this is a layered process: first determine if there is

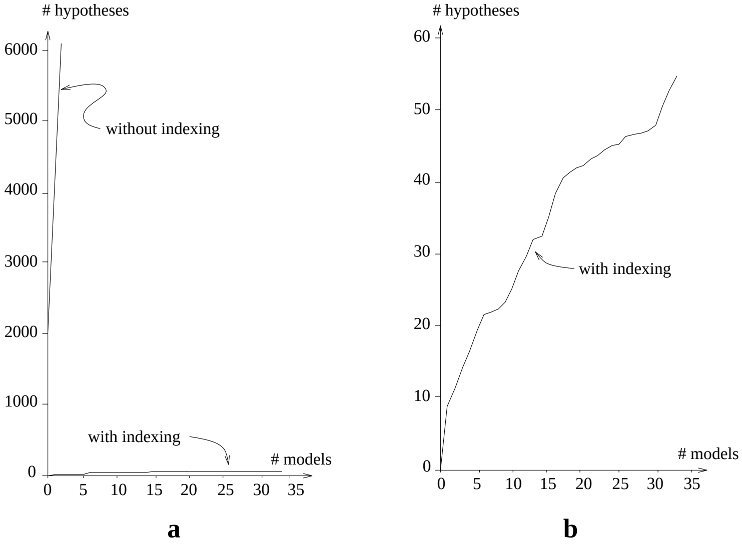


Figure 40: *(a) the number of hypotheses that have to be verified varies with the number of models from the model base. The results show an average over fifty scenes containing only one object in the model base, but with other clutter and occlusion present. Over 2000 indexes are created for the scene, which corresponds to the number of hypotheses that would have to be verified* **per model feature group** *if an alignment paradigm were used. Therefore, there is a rapid linear growth in the number of hypotheses created as the model base is expanded. However, the number of hypotheses created through indexing remains substantially lower; the detail depicted in (b) demonstrates that approximately a low constant of proportionality linear growth is observed. This ties in with the theoretical prediction of section 3.5.1.*

a common projective transformation for all geometric components (lines, conics, $\mathcal{M}$ curves) of the joint hypotheses. Second, back project geometric features and measure image support.

This strategy can fail, generally as a false positive, for two principal reasons. First, only projective geometric structure is used and many object boundary shapes are equivalent up to a projective transformation. In order to discriminate further, it is necessary to assume viewing conditions where an affine transformation is valid, or to use a calibrated camera which enables scaled Euclidean reconstruction. The second type of failure is associated with incomplete image support, which is discussed in more detail in the next section. Examples of both these failure cases are given in figures 42 and 43.

4.1.1 Image Support

Hypothesis validation based on image support is faced with two opposite failure mechanisms: too little support; or too much. When the object boundary exhibits little image contrast, a significant fraction of the achievable perimeter is unrecoverable by edge detection algorithms. On the other hand, when the background is highly textured or cluttered, high support can be achieved for an incorrect hypothesis. Two examples of the latter failure mechanism is shown in figure 45.

Both of these problems are symptomatic of having too sparse a description for the object. Thus far we

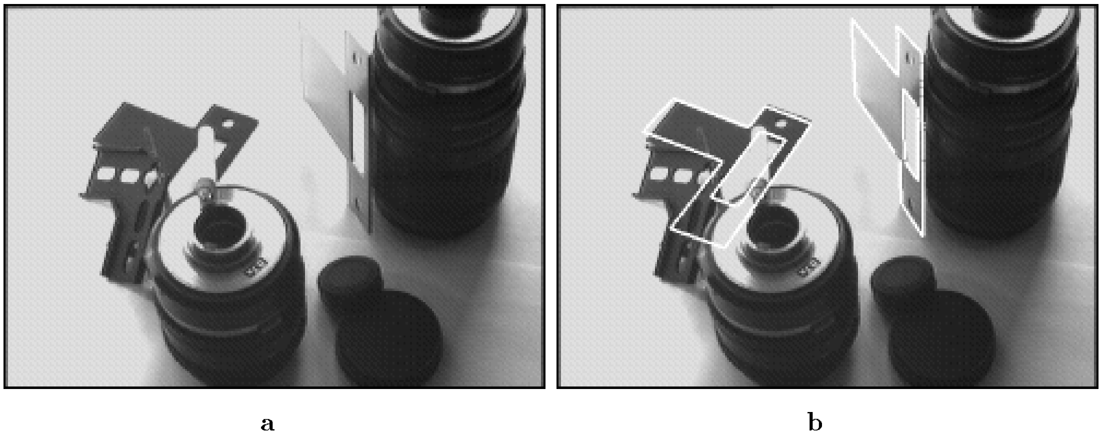


Figure 41: *Two objects from the model base are recognised correctly despite strong perspective distortion.*

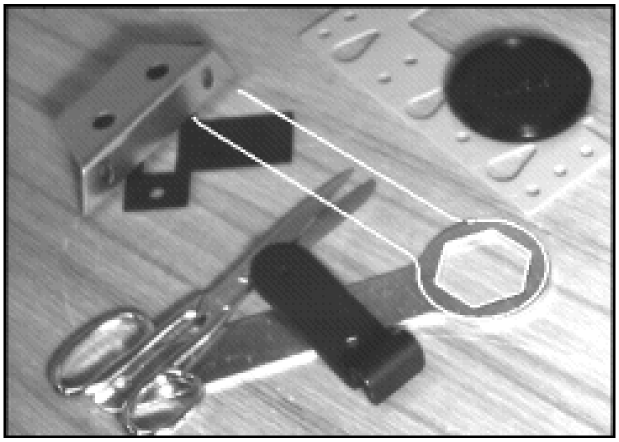


Figure 42: *The spanner from figure 27 is shown and recognised, but with the wrong orientation; due to texture in the image a* 52.1% *edge match is still found.*

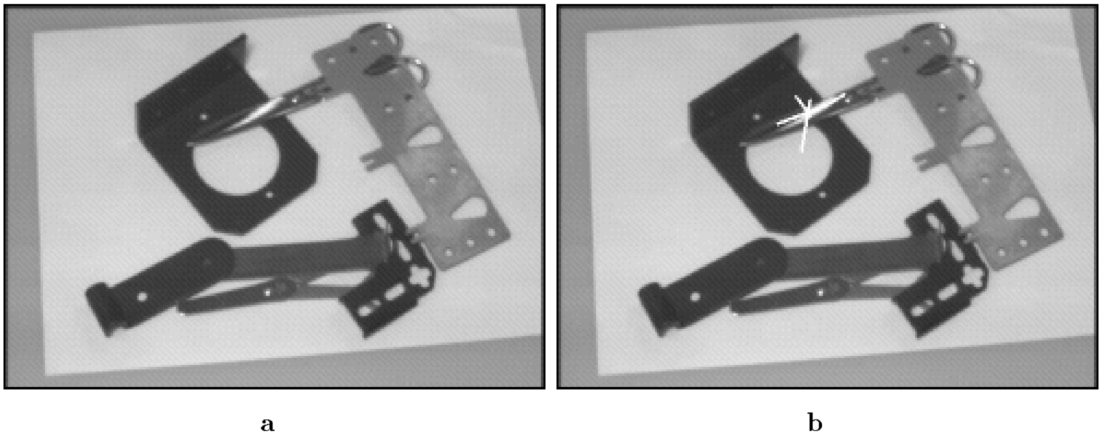


Figure 43: *An object from the model base which is superimposed in (b) can be recognised with over* 50% *edge support from the specularity on the pair of scissors in (a).*

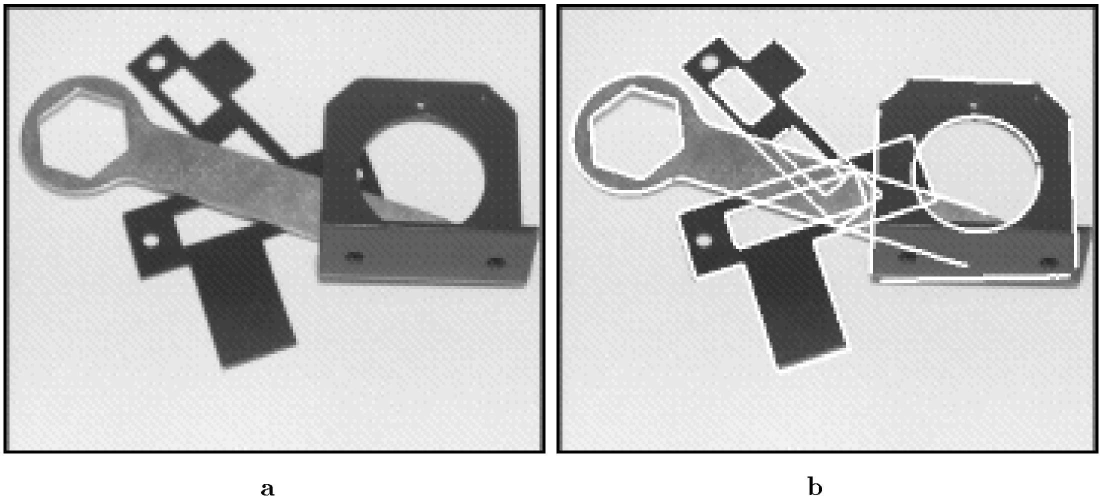


Figure 44: *Six objects were recognised from image (a). The four correct matches are shown in (b), with the two incorrect given in figure 45. The worst of the four correct identifications had two invariants and* 60.3% *match.*

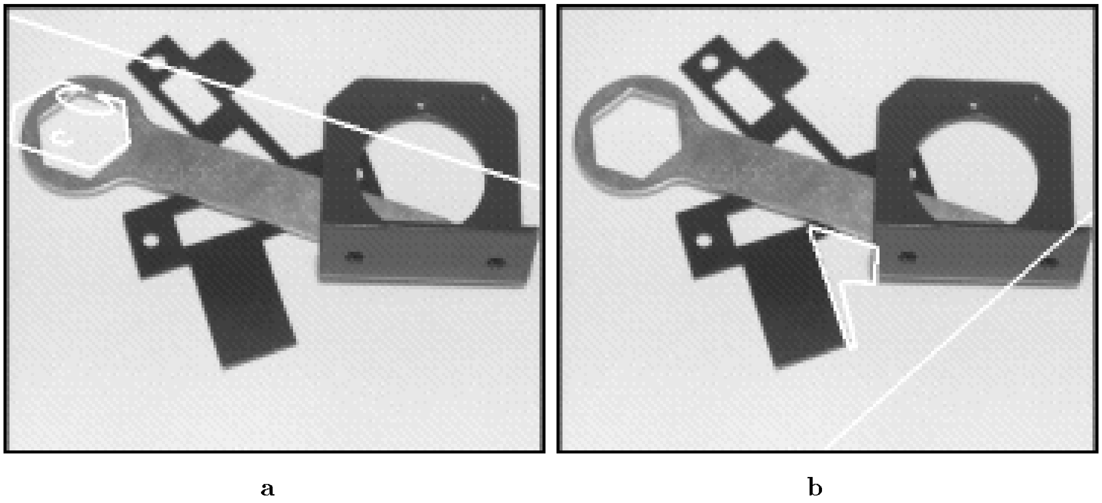


Figure 45: *The two incorrectly recognised objects from the image in figure 44(a). Unlike the matches shown in figure 44(b) these two objects were hypothesised by only a single invariant, and had less than* 52% *image support. In both cases image support is provided by features that have already been used to verify hypotheses in figure 44(b). The straight line across each image is the projection of the line at infinity from the acquisition images. Its closeness to the image center indicates an unlikely object pose (see the discussion in section 4.1.2 for details).*

have relied just on the boundary curve of the object and have ignored any properties of the interior region. It is certainly reasonable to use our knowledge of the model coordinate frame in the image to extract viewpoint independent texture measures. Even very simple measures would have eliminated the false positive shown in figure 42. It is also possible that very simple intensity measures on the internal object surface can be used. For example, the ratio of intensities in the neighbourhood of step discontinuities is a reliable measure of albedo ratio [Nayar93]. Even very weak intensity measures for discrimination can be used to increase confidence in a hypothesis or to break ties between two very similar geometric configurations.

There is also the open question of whether a feature should be used to support a hypothesis when it has already been used in a confirmed hypothesis, as illustrated in figure 45.

When insufficient image support is found it will be necessary to invoke additional understanding of scene of the form "this object is on top of another, and therefore occludes it". An understanding of this kind can be used to guide a search for further object features which support the explanation for missing features in the first object.

4.1.2 Projective Transformation

One limitation of the full projective transformation is that unreasonable perspective projections are allowed. An example is shown in figure 43 where the entire object is backprojected onto a thin specularity. To eliminate such projections, we propose a stratified solution involving progressively more knowledge of affine followed by Euclidean structure and ultimately, full perspective camera calibration.

1. **Real cameras**

   For a physical camera the object must lie in front of the image plane. More precisely the object plane cannot intersect the focal plane (a plane parallel to the image plane, containing the optical center). This is captured by projecting the ideal line, that is the line at infinity, of the model image onto the target image. Any case in which the ideal line passes through the convex hull of the hypothesised image features can be ruled out, since objects are assumed to be finite. More generally, if the model ideal line is observed within the finite bounds of the image plane, the object pose must be sufficiently extreme that the hypothesis can be rejected. In our experiments, about 25% of false positives due to poor poses ruled out by this constraint (see figure 45 and [Rothwell94] for details).

2. **Similarity structure**

   If the acquisition view is taken with the object in a fronto-parallel plane, one can calculate slant and tilt of the plane of the object in the target image. This pose calculation does not require full calibration of internal camera parameters. In the perspective case, the computation does not require focal length and in the affine case only the image pixel aspect ratio is required.

3. **Size**

   Two additional calibration parameters are essential for the computation of size. The first is distance from the camera. In order to estimate this distance, an approximate knowledge of object area, a Euclidean measure, and focal length are required. The second calibration parameter is the physical size of pixels on the image plane.

4.2 Future Work

1. For non-algebraic curves there are other invariants available which do not require $\mathcal{M}$ curves. For example, Van Gool *et al* [VanGool91] exploit single inflections as distinguished points; Carlsson [Carlsson92], fits conics tangent to the curve at four points. The latter procedure is applicable even to convex curves. There are numerous other covariant constructions (for example tangents between two curve segments) that can be utilised to generate distinguished points and hence invariants. The natural stage for integrating the various categories of invariant is at hypotheses combination. Again joint invariants between features involved in more global invariant groupings can efficiently be used to build larger model hypotheses. This integration strategy is currently under investigation [Rothwell93b].

2. Feature grouping based on sequential connectivity is a somewhat fragile process. It is easy to encounter large gaps in the object boundary due to low image contrast and occlusion. Any recognition algorithm will be adversely affected by these effects, however it is impossible to recover from when the index is constructed based solely on the assumption of boundary connectivity. An immediate way to overcome this problem is to use as many feature groups as possible for a given object to derive a redundant description, however many object shapes do not have sufficient complexity to define more than a few independent feature groups.

   Current work is investigating how grouping can be improved for applications where segmentation provides poor boundary connectivity, for instance in aerial reconnaissance scenes. The primary grouping relations are proximity and collinearity. By constructing a Delaunay triangulation of the set of line segment endpoints, it is computationally feasible to establish line segment sequences which are not actually connected topologically. Similarly, line segments which are reasonably close and collinear can also be grouped efficiently.

3. We observe that a useful goal for image feature segmentation and grouping is to provide feature groups which support invariant computations, for example the algebraic curves and $\mathcal{M}$ curves used in the current system. As a consequence, the evolution and testing of new segmentation and grouping algorithms can be tested by an evaluation of the accuracy and stability of resulting invariants. Additionally, the discovery of new invariant constructions will require the development of associated feature extraction algorithms. Since we know that the robustness of recognition is largely dependent on the success of such group constructions, we can profitably focus research on this stage of the system.

4. We have demonstrated a recognition complexity of low gradient linear growth with the size of the model base, and developed a statistical model of this performance. These results are still preliminary,

firstly because the model base is still relatively small (less that 50 models), and secondly because the objects are fairly similar. It is an open question as to whether this is simply clustering behaviour and if for a large model base (several thousand objects), recognition would remain asymptotically constant time.

5. A number of recent papers have demonstrated that invariants of 3D structures, under 3D projective transformations, can be extracted from image projections of the structure. These can be obtained from multiple views [Demey92, Faugeras92, Hartley92, Koenderink91, Mundy92, Quan91] or from a single view [Forsyth92, Forsyth93, Liu93, Rothwell93a, Rothwell94, Wayner91].

   We propose to employ our experiences with LEWIS by building an improved recognition system called MORSE (Multiple Object Recognition by Scene Entailment) for 3D structures. To recognise such structures an improved architecture is required. For example: first, we require interactions between different types of invariant working simultaneously in a single image; second, fine-grained communication loops are required between the different processing layers than provided within the current grouping-indexing-correspondence architecture. These must be implemented in such a way as to ensure that the implications of each local conclusion are understood by all other layers. Furthermore, multiple representations of objects must be allowed by the model library. For instance, curve g on the spanner in figure 29 could be represented both as a concavity curve, and as a five line sequence. Both representations should be included in the model library.

Acknowledgements

We acknowledge discussions with various members of the Oxford University Robotics Reseach Group, especially Paul Beardsley, Mike Brady, Ian Reid, Mike Taylor and David Sinclair. CAR and JLM were supported by General Electric working under the following contracts: ARPA grant MDA97291C0053 and a grant from the United States Air Force Office of Scientific Research, AFOSR-91-0361. DAF was supported in part by the National Science Foundation under award no. IRI-9209729, and in part by a National Science Foundation Young Investigator Award with matching funds from GE, Eugene Rikel, Rockwell International and Tektronix. AZ acknowledges support of Esprit Basic Research Action 6448 VIVA. The final version of the text was improved through the reviewers' comments.

References

[Ayache86] Ayache, N. and Faugeras, O.D. 1986. HYPER: a new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-8(1), 44-54.

[Ayache87] Ayache, N. and Faugeras, O.D. 1987. Building a consistent 3D representation of a mobile robot environment by combining multiple stereo views. In *Proc.* IJCAI, pp.808-810.

[Binford81] Binford, T.O. 1981. Inferring surfaces from images. *Artificial Intelligence* 17, 205-244.

[Binford93] Binford, T.O. and Levitt, T.S. 1993. Quasi-invariants: theory and explanation. In *Proc.* DARPA IUW, pp.819-829.

[Bolles87] Bolles, R.C. and Horaud, R. 1987. 3DPO: a three-dimensional part orientation system. In *Three Dimensional Vision.* Kanade, T. editor, Kluwer Academic Publishers, 399-450.

[Bookstein79] Bookstein, F. 1979. Fitting conic sections to scattered data. *Computer Vision Graphics and Image Processing* CVGIP-9, 56-71.

[Borgefors88] Borgefors, G. 1988. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-10(6), 849-865.

[Brooks83] Brooks, R.A. 1983. Model-based three-dimensional interpretations of two-dimensional images. *Pattern Analysis and Machine Intelligence* PAMI-5(2).

[Califano92] Califano, A. and Mohan, R. 1992. Multidimensional indexing for recognizing visual shapes. *Visual Form*, 109-118.

[Canny86] Canny J.F. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-8(6), 679-698.

[Carlsson92] Carlsson, S. 1992. Projectively invariant decomposition of planar shapes. In *Geometric Invariance in Computer Vision.* Mundy, J.L. and Zisserman, A.P. editors, MIT Press.

[Cass92] Cass, T.A. 1992. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In *Proc.* ECCV, pp.834-842.

[Clemens91] Clemens, D.T. and Jacobs, D.W. 1991. Model group indexing for recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-13(10), 1007-1017.

[Cox92] Cox, I.J., Rehg, J.M. and Hingorani, S. 1992. A Bayesian multiple hypothesis approach to contour grouping. In *Proc.* ECCV, pp.72-77.

[Demey92] Demey, S., Zisserman, A. and Beardsley, P. 1992. Affine and projective structure from motion. In *Proc.* BMVC, pp.49-58.

[Duda73] Duda, R.O. and Hart P.E. 1973. *Pattern Classification and Scene Analysis.* Wiley.

[Ettinger88] Ettinger, G.J. 1988. Large hierarchical object recognition using libraries of parameterized model sub-parts. In *Proc.* CVPR, pp.32-41.

[Faugeras92] Faugeras, O. 1992. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc.* ECCV, pp.563-578.

[Forsyth91] Forsyth, D.A., Mundy, J.L., Zisserman, A.P., Coelho, C., Heller, A. and Rothwell, C.A. 1991. Invariant descriptors for 3-D object recognition and pose. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-13(10), 971-991.

[Forsyth92] Forsyth, D.A., Mundy, J.L., Zisserman, A.P. and Rothwell, C.A. 1992. Recognising curved surfaces from their outlines. In *Proc.* ECCV, pp.639-648.

[Forsyth93] Forsyth, D.A. 1993. Recognizing algebraic surfaces from their outlines. In *Proc.* ICCV, pp.476-480, 1993.

[Forsyth94] Forsyth, D.A., Mundy, J.L., Rothwell, C.A. and Zisserman, A.P. 1994. MORSE: Multiple Object Recognition by Scene Entailment. GE Corporate Research and Development, NY, document in preparation.

[Fisher89] Fisher, R.B. 1989. *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis.* John Wiley and Sons.

[Goad83] Goad, C. 1983. Special purpose automatic programming for 3D model-based vision. In *Proc.* DARPA IUW, pp.371-381.

[Grimson87] Grimson, W.E.L. and Lozano-Pérez, T. 1987. localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-9(4), 469-482.

[Grimson90] Grimson, W.E.L. 1990. *Object Recognition by Computer, The Role of Geometric Constraints.* MIT Press.

[Gueziec93] Gueziec, A. and Ayache, N. 1993. New developments on geometric hashing for curve matching. In *Proc.* CVPR, pp.703-704.

[Hartley92] Hartley, R.I., Gupta, R. and Chang, T. 1992. Stereo from uncalibrated cameras. In *Proc.* CVPR, pp.761-764.

[Huttenlocher87] Huttenlocher, D.P. and Ullman, S. 1987. Object recognition using alignment. In *Proc.* ICCV, pp.102-111.

[Huttenlocher88] Huttenlocher, D.P. 1988. Three-dimensional recognition of solid objects from a two-dimensional image. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT.

[Huttenlocher91] Huttenlocher D.P. 1991. Fast affine point matching: an output-sensitive method. In *Proc.* CVPR, pp.263-268.

[Jacobs92] Jacobs, D.W. 1992. Space efficient 3D model indexing. In *Proc.* CVPR, pp.439-444.

[Kalvin86] Kalvin, A., Schonberg, E., Schwartz, J.T. and Sharir, M. 1986. Two-dimensional, model-based, boundary matching using footprints. *International Journal of Robotics Research* IJRR-5(4), 38-55.

[Koenderink91] Koenderink, J.J. and Van Doorn, A.J. 1991. Affine structure from motion. *J. Opt. Soc. Am. A.* 8(2), 377-385.

[Lamdan88] Lamdan, Y., Schwartz, J.T. and Wolfson, H.J. 1988. Object recognition by affine invariant matching. In *Proc.* CVPR, pp.335-344.

[Lowe85] Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers.

[Lowe87] Lowe, D.G. 1987. The viewpoint consistency constraint. *International Journal of Computer Vision* IJCV-1(1), 57-72.

[Liu93] Liu, J., Mundy, J.L., Forsyth, D.A., Zisserman, A. and Rothwell, C.A. 1993. Efficient recognition of rotationally symmetric surfaces and straight homogeneous generalized cylinders. In *Proc.* CVPR, pp.123-128.

[Marr82] Marr, D. 1982. *Vision.* Freeman.

[Maybank93] Maybank, S.J. 1993. Probabilistic analysis of the application of the cross ratio to model based vision. GEC Hirst Research Centre, Technical Report.

[Mundy92] Mundy, J.L. and Zisserman, A.P. 1992. *Geometric Invariance in Computer Vision.* MIT Press.

[Murray87] Murray, D.W. 1987. Model-based recognition using 3D structure from motion. *Image and Vision Computing* IVC-5, 85-90.

[Nayar93] Nayar, S.K. and Bolle, R.M. 1993. Reflectance ratio: a photometric invariant for object recognition. In *Proc.* ICCV, pp.280-285.

[Nielsen88] Nielsen, L. 1988. Automated guidance of vehicles using vision and projective invariant marking. *Automatica* 24, 135-148.

[Pollard89] Pollard, S.B, Pridmore, T.P, Porrill, J., Mayhew, J.E.W. and Frisby, J.P. 1989. Geometrical modeling from multiple stereo views. *International Journal of Robotics Research* IJRR-8(4), 132-138.

[Quan91] Quan, L., Gros, P. and Mohr, R. 1991. Invariants of a pair of conics revisited. In *Proc.* BMVC, pp.71-77.

[Reid91] Reid, I. 1991. Recognising parameterized models from range data. D.Phil. Thesis, Department of Engineering Science, Oxford University, Oxford.

[Rigoutsos91] Rigoutsos, I. and Hummel, R. 1991. Implementation of geometric hashing on the connection machine. In *Proc.* IEEE Workshop on Directions in Automated CAD-Based Vision, pp.76-84.

[Rothwell93a] Rothwell, C.A., Forsyth, D.A., Zisserman, A. and Mundy, J.L. 1993. Extracting projective information from single views of 3D point sets. In *Proc.* ICCV, pp.573-582.

[Rothwell93b] Rothwell, C.A. 1993. Hierarchical object descriptions using invariants. In *Proc.* 2^{nd} ARPA/NSF-ESPRIT Workshop on Invariance, pp.287-302.

[Rothwell94] Rothwell, C.A. 1994. *Object Recognition through Invariant Indexing.* To appear OUP.

[Schwartz87] Schwartz, J.T. and Sharir, M. 1987. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *International Journal of Robotics Research* IJRR-6(2), 29-44.

[Semple52] Semple, J.G. and Kneebone, G.T. 1952. *Algebraic Projective Geometry.* OUP.

[Shaashua88] Sha'ashua, A. and Ullman, S. 1988. Structural saliency: the detection of globally salient structures using a locally connected network. In *Proc.* ICCV, pp.321-327.

[Sinclair93] Sinclair, D.A., Blake, A., Smith, S. and Rothwell, C.A. 1993. Planar region detection and motion recovery. *Image and Vision Computing* IVC-11(4), 229-234.

[Slama80] Slama, C.C. 1980. *Manual of Photogrammetry.* American Society of Photogrammetry, 4^{th} edition.

[Stein92] Stein, F. and Medioni, G. 1992. Structural indexing: efficient 2-D object recognition. *Pattern Analysis and Machine Intelligence* PAMI-14, 1198-1204.

[Stockman87] Stockman, G. 1987 Object recognition and localization via pose clustering. *Computer Vision Graphics and Image Processing* CVGIP-40, 361-387.

[Taubin91] Taubin, G. and Cooper, D.B. 1991. Object recognition based on moment (or algebraic) invariants. *IBM TR-RC17387* IBM T.J. Watson Research Centre, P.O. Box 704, Yorktown Heights, NY 10598.

[Thompson87] Thompson, D.W. and Mundy, J.L. 1987. Three-dimensional model matching from an unconstrained viewpoint. In *Proc.* ICRA, pp.208-220.

[VanGool91] Van Gool, L. Kempenaers, P. and Oosterlinck, A. 1991. Recognition and semi-differential invariants. In *Proc.* CVPR, pp.454-460.

[Wayner91] Wayner, P.C. 1991. Efficiently using invariant theory for model-based matching. In *Proc.* CVPR, pp.473-478.

[Weiss88] Weiss, I. 1988. Projective invariants of shapes. In *Proc.* DARPA IUW, pp.1125-1134.

[Wolfson92] Wolfson, H.J. 1992. Object recognition by transformation invariant indexing. In *Proc.* Invariance Workshop, ECCV.