Efficient Model Library Access by Projectively Invariant Indexing Functions

C.A. Rothwell, A. Zisserman, J.L. Mundy and D.A.Forsyth
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ
England

Abstract

Projectively invariant shape descriptors allow fast indexing into model libraries without the need for pose computation or camera calibration. This paper describes progress in building a model based vision system for plane objects that uses algebraic projective invariants. We give a brief account of these descriptors and then describe the recognition system, giving examples of the invariant techniques working on real images.

1 Introduction.

A major unsolved problem in model-based vision is the construction of fast recognisers for large model bases. Current practice involves trying to place each object from a library in a scene, and then evaluating a hypothesis based on the consistency of the best projection of the model onto the image features. This constitutes simultaneously finding pose and performing recognition, and is generally of a complexity linear in the number of models in the library.

Ideally the first steps in the generation of a model hypothesis should not require specific information about the identity of the object being recognised, but most model-based vision systems exploit the pose consistency of image feature to model feature assignments in order to form hypotheses. Such systems fall into two main categories: interpretation tree techniques (such as those of Ayache and Faugeras [1] or Grimson and Lozano-Pérez [6]); or transformation determination methods (those of Huttenlocher and Ullman [7], Lowe [10] or Mundy and Heller [11]).

For an object recognition system with a large number of models, testing for the presence of each model becomes impractical. Instead, it is necessary to introduce the concept of indexing functions. These provide direct access to a certain model in the data base without using specific information about the model, or model pose in advance. Consequently, recognition complexity is superior to pose based systems.

An index is an image measure that remains constant (invariant) for a particular object over all viewpoints. *Each index may generate a model to image match hypothesis which is confirmed or rejected by projecting the model onto the image.* This verification step is identical to that used by the interpretation tree or transformation determination techniques. However, in this case, the cost of generating the model hypothesis does not have the same dependency on the size of the model library.

Indexing functions are becoming a familiar tool for recognition. Many examples are already available in the literature: Forsyth, *et al.* [5], Nielsen and Sparr [13] (both projective), Lamdan, *et al.* [9] Wayner [16], Clemens and Jacobs [4] and Huttenlocher [8] (all affine).

The index functions we use are *algebraic invariants* computed from algebraic curves fitted to Canny [3] edge data. For each object the invariants are combined into *index vectors*; these are lists of indexes as well as a topological relationship between the constituent geometric features. Each component is used to generate a recognition hypothesis. Hypotheses are combined through connectivity prior to verification.

In the following sections we describe the model based system, and demonstrate recognition from real images using a library of over thirty models. A study of the increase in recognition time with the size of the library is also reported.

2 Invariant Indexing Functions.

The invariants we use are constructed from plane algebraic curves; ie. lines and conics. The actual expressions for the invariants are familiar in the literature and are omitted here (see [12]). Three different invariants have proved to be both reliable and stable:
1. Five Coplanar Lines: Five lines in general position give two independent invariants. For degenerate configurations of the lines one of the invariants is trivial and, in this case, there is only one useful index.
2. A Conic and Three Lines: Three indexes can be computed for a conic and three lines. Each invariant is computed from the conic and a pair of lines.
3. A Pair of Conics: A pair of coplanar conics yield two invariant indexes; this invariant was tested extensively in the work reported in [5].

3 The Model Based System.

The major components of the object modeling and recognition process are:
1. Feature extraction: The conics and lines needed to form the invariants are extracted from image edge data.
2. Model Construction: A set of features is associated with a particular object by providing one or more images of the object by itself, from which the model invariants can be computed. The model's edge data is also stored in the library for use during verification.
3. Hypothesis generation: The invariants for groups of features in a scene are computed. We index the measured invariants against invariant values in the library using a hash table, and if they match, produce a recognition hypothesis. Hypotheses for the same object are then combined into index vectors before verification.
4. Hypothesis verification: When a potential match is found we confirm it by projecting model features (algebraic) and also edge data from an acquisition image into the test scene. Should the projected and scene data be sufficiently similar the match is confirmed.

We now present these steps in more detail.

3.1 Curve Segmentation and Grouping.

Continuous edge curves, or curves with single pixel breaks, are taken from Canny output [3]. Edge segments that are approximated by lines and conics are represented by the corresponding algebraic form. Full details are given in [14]. The remaining edge data is still useful for verification or for forming other projective invariants (see figure 1).

The segmentation process produces a disjoint set of lines, conics and higher order plane curves. We group straight lines into cyclic edge chains such that lines from single edge chains are put in the same group, and the ordering around the curve is preserved in the chain. We compute invariants only for lines that are adjacent in the edge chains, and so reduce the overall recognition complexity. For example, to compute the five line invariant for a set of n lines we might consider $O(n^5)$ different groupings, however, if the lines are part of a chain we consider only $O(n)$ invariants (interestingly there are only $O(n)$ independent invariants for n lines). Even without chaining, forming invariants from higher order curves reduces complexity. For example only two conics are needed to form an invariant, and so the complexity is $O(n^2)$.

3.2 Model Acquisition.

When projective invariants are used, models can be acquired directly from images. We compare the invariants measured for a pair of images of an unoccluded object, and all invariants that are similar in both views (essentially constant) are stored in the library. Edge data from one of the images is also stored for use during verification.

The library consists of a number of sub-libraries; each one is in the form of a hash table with the invariants as keys. The entries in the tables are duplicated so that invariants still index the correct models even when their values have been perturbed by image noise. Much work remains to be done on the understanding of the effects of image noise on invariants; an introduction to the analysis is given in [5], but a full investigation of the reliability of the indexes remains a goal of future research.

3.3 Recognition.

Recognition is based on the groups of lines and conics constructed by the segmentation process. These groups are used to form the same invariant indexes which were used for defining the model. The indexes are used to retrieve model hypotheses from the library. All hypotheses are combined into joint hypotheses before verification.

For each feature group we compute a list of invariants. Each invariant is then matched via the hash table to an object name in the model library if the invariant value (within a 5% error bound) is stored in the library. Every time an invariant indexes a model,

a hypothesis is formed. Because many invariants may actually correspond to the same object, and should therefore be part of the same hypothesis, we form *joint hypotheses* which are lists of 'compatible' hypotheses.

The reasons for forming the joint hypotheses are:
1. Combining individual measures provides better discrimination between objects.
2. Two hypotheses indexing the same object in a single part of the scene significantly increases confidence that the match is correct.
3. Multiple hypotheses give more matching model and image features than a single hypotheses and allow the determination of a more accurate model to image map.
4. It is more efficient to validate two correct hypotheses from the same object together, rather than separately, since the verification process is applied once only.

Two (joint) hypotheses are combined into a single index vector if the hypotheses are compatible; this is based on topological adjacency of the features used to compute the invariants.

Since topological relations are often unreliable it is possible that two hypotheses could be combined into a single joint hypothesis even though they may be totally unrelated (for example one may represent a correct match and the other may have been caused by clutter). We therefore maintain a list of all the original hypotheses and all possible combinations of compatible hypotheses. The list is ordered by descending number of simple hypotheses per joint hypothesis. Those with more hypotheses are verified first, and if the match is confirmed, other joint hypotheses that represent partial versions of the hypothesis are deleted. The joint hypothesis formation stage can potentially cause an exponential number of hypotheses to be formed. We find, in practice, that deleting verified hypotheses keeps the process under control.

3.4 Verification.

Once a potential match to a model has been found, the hypotheses are verified to remove recognition false positives and yet prevent false negatives. Verification is both expensive and hard for occluded scenes. Verification is performed in two stages:
1. Confirm that the model algebraic curves can be projected onto their image counterparts.
2. Search the image for further edge support for the model hypothesis.

The first stage is the computation of the model to image projection. This can be found from the matched features used to form the invariants. Except in the case of an isotropy [12], invariants always provide a sufficient number of constraints to determine the projection. This stage can be used to eliminate a proportion of the recognition hypotheses. For example, four line correspondences determine the projection uniquely. For the five line invariant, we have five line correspondences, and so the projectivity is over constrained. If the match is incorrect, the computed projectivity will not map the model lines onto the image lines, and so the hypothesis can be rejected. In a similar manner other configurations of features can be eliminated.

The second stage involves searching the image for features not used to form the hypothesis. Using the transformation computed above, model edge data (from an acquisition scene) is projected into the image. If the projected edge data lies close (within 5 pixels) to image edge data of the same orientation (within 15^o), we assume that the object caused the image edge data in the image, and thus counts as support for the object actually being visible. If more than a certain proportion of the projected model data is supported (we use 50%), we assume that there is sufficient support for the model, and the recognition hypothesis is confirmed. This part of the process is very expensive as $O(10^3)$ edgels must be mapped into the image; this forms one of the slowest parts of the entire process.

Computing the Euclidean distance from a point in the image to the nearest edge location is expensive, so we compute an approximation to the distance using the 3-4 distance transform of Borgefors [2]. The assumed edge orientation of image edgels is that of the Canny output, whereas that of projected features is based on the actual feature that is projected.

3.5 Complexity.

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. There are two contributions to the number of hypotheses formed during indexing:
1. Whether the invariant indexes a model.
2. The number of clashes expected during hashing.

The first is affected by whether the invariant genuinely corresponds to a model, or whether it is due to 'noise'. Here we give an informal argument for the likelihood that a noisy invariant 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 invariant indexes.

Consider a single invariant for a set of features that forms an n dimensional index, of which each dimension is considered to have the same behaviour. Let each index cover a segment on the real line from i_0 to $i_0 + L$, and the quantisation along the line be δ, a constant quantity over the line segment[1]. We require $b = L/\delta$ buckets along the line, and so for n indexes and *assuming* that the measured invariants have a constant PDF over the invariant space[2], 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 number, there will be $\alpha\epsilon\lambda$ entries in the table. If we assume that these entries are spread uniformly over the hash table, then the chances of indexing a model through noise is $(\alpha\epsilon\lambda)/b^n$.

This analysis means that we have 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, which is essentially constant. We immediately see 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.

The problems associated with making n large are:
1. For algebraic invariants we have little control over n. If we use minimal feature groups we have no control, but by using larger structure such as a conic and three lines we can increase n. For the invariants of other structures, such as the plane curves reported in [15], we can make n large.
2. Making n too large means that the hash table occupies a large portion of machine memory, and so the cost of accessing cells is large. We are currently investigating (through the work of [14]) the most effective number of indexes for recognition.

4 Experiments.

In this section results from the recognition algorithm are reported. Figures 1 to 3 show the system operating on a few test scenes with some of the match statistics shown. For figure 1, 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 and 78 needed the second stage. For figure 2, 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.

The graph in figure 5 shows how the number of hypotheses that have to be verified increases with the number of models in the model base. The cost does slightly increase with the number of models but is significantly less than hypothesising and testing each model in the library.

5 Discussion.

This paper describes the development of a model based vision system which uses projectively invariant descriptors to expedite recognition. Segmentation and grouping still pose a significant problem in the construction of a real recognition system, and so more intelligent pre-processing algorithms are required. For example, many of the conics with little support in figure 1 could be eliminated which would reduce the number of indexes computed. The use of the canonical frame construction for projective invariants [15] may overcome this problem because is does not place such stringent requirements on grouping.

During the development of this system, we originally used as an index the single invariant of a conic and a pair of lines. However, although this was stable over views, it provided insufficient discrimination between models in the data base. Adopting the conic and three line invariant has overcome this by increasing the dimension of the hash table.

Verification is hard for occluded scenes because an incorrect match may have as much image support as a heavily occluded correct match; that is, for scenes where there is dense edge data it is quite likely that a large number of edges may be close to, and have the same orientation as, the projected model edges (see the example in figure 4). As industrial scenes are very structured, only one or two erroneous straight lines of the right orientation may be sufficient to give over 50% support for a model and so render a false match. Obviously, any object which is over 50% occluded can not be found by the recogniser, and so there is a certain tradeoff required when setting the support threshold. As the threshold is lowered, an occluded object is more likely to be found, but there will also be more false positives.

Therefore, our work has shown that the two stage verification process is insufficient in general, and that a third stage is required. This could be based on partial pose information that would eliminate unrealistic

[1] More exactly we should work on a logarithmic scale as the errors in the invariants are proportional to the invariant values [5], but this just complicates the analysis.

[2] This claim is a current topic of research, and should be compared to the work of Hopcroft, *et al.* [12].

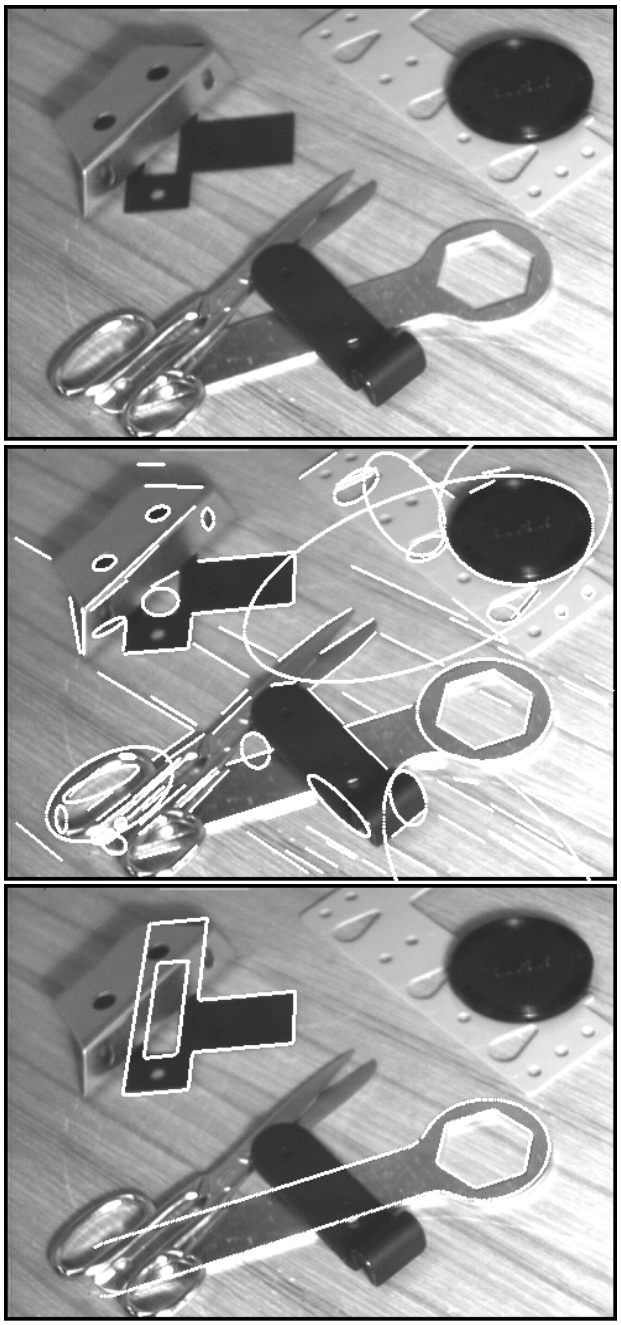

Figure 1: The upper image shows a scene containing two objects from the model base, with fitted lines (100 of them) and conics (27) superimposed on the middle image. Note that many lines are caused by texture, and that some of the conics correspond to edge data over only a small section. The lines form 70 different line groups. The lower image 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.

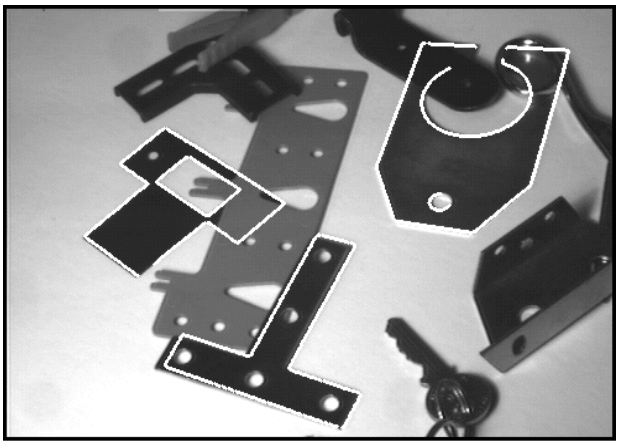

Figure 2: This image shows another typical scene containing three objects from the model base. The recognised objects are outlined with 74.7% (2 invariants), 84.6% (1) and 69.9% (3) edge matches for the objects from left to right. 58 lines and 14 conics were found.

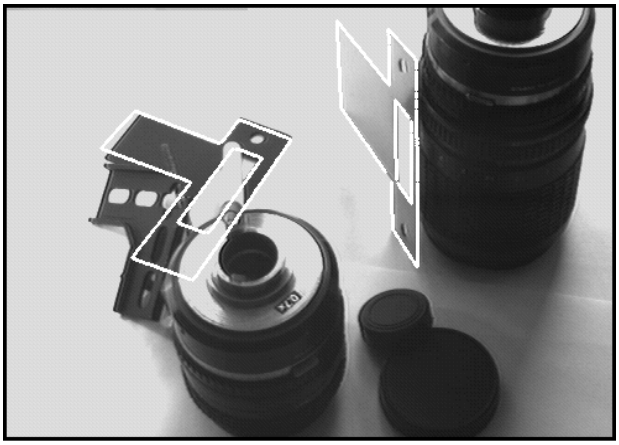

Figure 3: Two objects from the model base are recognised correctly despite strong perspective distortion.

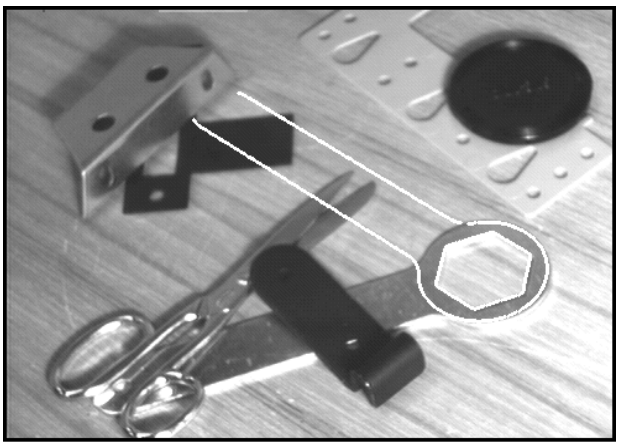

Figure 4: Here we show the spanner from figure 1 recognised, but with the wrong orientation; due to texture in the image a 52.1% edge match is still found.

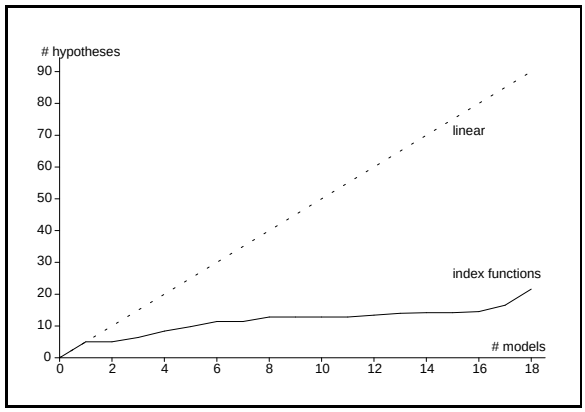

Figure 5: The graph shows how the number of hypotheses requiring full verification varies with the size of the model base. The results depict an average for scenes containing one object from the library, with clutter and occlusion. The projected line illustrates linear cost, which is worse than the measured behaviour (requiring 21.6 hypotheses to be verified for 18 models). The increase on the addition of model 18 is because it is very similar to the test object. Similar performances are observed for other objects.

solutions. Furthermore, pose arbitration between different objects in the scene should be implemented.

The size of the model base in our system is large by current standards, 18 real objects and 15 of the labels as reported in [5], but this is still too small to experience the real difficulties with large model libraries. We have, however, been able to demonstrate the benefits of using indexing functions compared to the less efficient transformation determination techniques. We have shown that hypotheses can be generated rapidly with little dependence on the size of the library. However, the number of false positives and the cost of verification depends on the collection of objects in the library. We are currently enlarging the model base to determine whether constant time indexing is possible for a large number of models. Clearly, the growth in verification cost will depend on the discriminating power of the invariant descriptors.

Future work will concentrate on the implementation of new indexing functions, such as the canonical frame construction for measuring projective invariants of general plane curves. This means that our system will be able to cope with a significant class of planar objects, though much work remains to be done on 3D shape representation and the associated invariants of 3D objects from their 2D image projection.

Acknowledgements

We thank Han Wang for his implementation of the transputer based Canny edge detector. CAR acknowledges the support of GE, AZ acknowledges SERC, JLM acknowledges the GE Coolidge Fellowship and DAF acknowledges Magdalen College, Oxford, GE and the University of Iowa. GE CRD is supported in part by DARPA contract DACA-76-86-C-007 and AFOSR contract F49620-89-C-003.

References

[1] Ayache, N. and Faugeras, O.D. "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects," *PAMI*-8, No. 1, p.44-54, January 1986.

[2] Borgefors, G. "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *PAMI*-10, No. 6, p.849-865, November 1988.

[3] Canny J.F. "A Computational Approach to Edge Detection," *PAMI*-6, No. 6. p.679-698, 1986.

[4] Clemens, D.T. and Jacobs, D.W. "Model Group Indexing for Recognition," Proceedings CVPR, p-4-9, 1991.

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

[6] Grimson, W.E.L. and Lozano-Pérez, T. "Localizing Overlapping Parts by Searching the Interpretation Tree," *PAMI*-9, No. 4, p.469-482, July 1987.

[7] Huttenlocher, D.P. and Ullman, S. "Object Recognition Using Alignment," Proceedings ICCV1, p.102-111, 1987.

[8] Huttenlocher D.P. "Fast Affine Point Matching: An Output-Sensitive Method," Proceedings CVPR, p.263-268, 1991.

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

[10] Lowe, D.G. "The Viewpoint Consistency Constraint," *IJCV*-1, No. 1, p.57-72, 1987.

[11] Mundy, J.L. and Heller, A.J. "The Evolution and Testing of a Model-Based Object Recognition System," Proceedings ICCV3, p.268-282, 1990.

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

[13] Nielsen, L. and Sparr, G. "Projective Area-Invariants as an Extension of the Cross-Ratio," Proceedings First DARPA-ESPRIT Workshop on Invariance, p.455-480, March 1991.

[14] Rothwell, C.A., Zisserman, A., Forsyth, D.A. and Mundy, J.L. "Plane Object Recognition Using Projectively Invariant Indexing Functions," *OUEL TR,* in preparation, 1992.

[15] Rothwell, C.A., Zisserman, A., Forsyth, D.A. and Mundy, J.L. "Canonical Frames for Planar Object Recognition," to appear ECCV2, 1992.

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