

Combining Geometric Invariants with Fuzzy Clustering for Object Recognition

Ellen L. Walker
Mathematical Sciences Department
Hiram College
Hiram, OH 44234
walkerel@hiram.edu
<http://hirame.hiram.edu/~walkerel>

Abstract

Object recognition is the process of identifying the types and locations of objects in the image. Earlier work has shown the desirability of using fuzzy compatibility for local feature correspondence and fuzzy clustering for pose estimation of two-dimensional objects. This paper extends the methodology to images of three dimensional objects by applying geometric invariants, specifically the cross ratio of four collinear points. The recognition process is divided into three subtasks: local feature correspondence, object identification, and pose determination. Algorithms are described for each subtask.

1. Introduction

When processing an image, it is important to be able to identify the types and locations of objects contained within that image. These objects might be subjects in an image database, landmarks for navigation, or targets for a robot's actions. Unless the objects can be correctly identified and located, the object-related tasks cannot be carried out. This paper will discuss methods for finding *known* objects in *unknown* orientations.

Previous work [8, 9] divided the object recognition task into two subtasks: (1) finding local correspondences between model features and image features, and (2) collecting correspondence information into hypotheses for the pose (position and orientation) of a single object. The features used were object segments, with correspondence based on similarity of segment length and angle between consecutive segments. For each feature correspondence, a pose, consisting of translation (x and y) and rotation (about z) was hypothesized. While successful, this work was limited to a single model and to an essentially two-dimensional problem, i.e. the camera was required to be looking straight down on the flat objects.

This paper describes extensions to the previous work that allow multiple objects to be recognized in the same image, and provide for much less restriction in allowable viewpoint. To allow multiple objects in the same image, we add a new subtask after local correspondence determination: object identification, or deciding which of the available models should be matched to the object. To allow unrestricted viewpoints, we change the

correspondence determination subtask to take advantage of projective geometric invariants.

Invariants for Object Recognition

An *invariant* is a measurement that does not change under a given class of transformations. More specifically, a geometric (or shape) invariant depends only on the shape of an object. In computer vision, invariants are "shape descriptors computed from the image which are independent of the viewpoint, that is, they are the same regardless of which point of view the image was taken from." [10] The range of allowable viewpoints determines the transformation class of invariants that is required.

When the viewpoint is restricted to a single point and the objects are planar and perpendicular to the camera, the shapes are transformed by a Euclidean transformation (rigid rotation in one dimension, plus translation). If a zoom lens can be used, then the transformation class is similarity (Euclidean plus change in scale). In the general case, however, the class of transformation between the object and the image is projective. (See [7] for a good explanation of the projective transform and its relation to the imaging process.) Therefore, for maximum generality of viewpoint, projective invariants should be used as image features.

In our previous work, the restriction to two-dimensional problems is directly due to the choice of length and angle as primary features. Angle is invariant under similarity transforms, but length is invariant only under Euclidean transforms. Thus, for length to be a useful feature, the transformation between the model and the object to be recognized must be Euclidean. This was enforced in the earlier work by defining the model parameters from an image taken with the same camera at the same location as the test image, using two-dimensional objects, and orienting the camera so that the viewing direction was perpendicular to the objects.

To remove the restrictions on objects and viewpoints, it is necessary to choose projective invariants for our features. Many projective invariants have been identified for use in computer vision [7, 10], but we use a very simple one: the cross ratio. The cross ratio is defined by four collinear points, as in Figure 1. Let AB denote the distance from point A to point B . Then the cross-ratio of points A, B, C, D

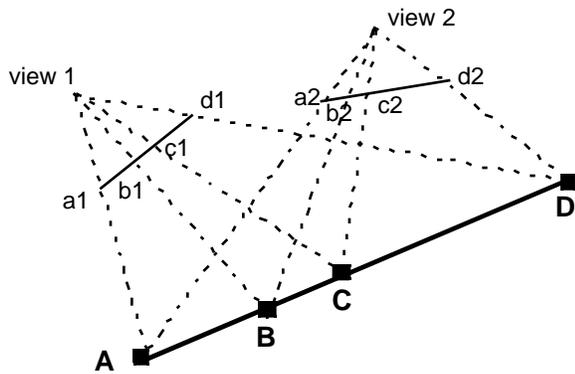


Figure 1: Cross ratio of four points in two views

is $(AB)(CD) / (AC)(BD)$. Figure 1 shows a set of four collinear points in the world, labeled A, B, C, and D, and two images of those points ($a_1 \dots d_1$ in view 1 and $a_2 \dots d_2$ in view 2). Because the cross-ratio is invariant, the image measurements $(a_1 b_1)(c_1 d_1) / (a_1 c_1)(b_1 d_1)$ and $(a_2 b_2)(c_2 d_2) / (a_2 c_2)(b_2 d_2)$ are the same as the actual (world) measurement $(AB)(CD) / (AC)(BD)$. Therefore, if a sequence of four collinear points is a feature, with the cross-ratio of the four points as the constraint on that feature, then the feature can be detected in an image from any viewpoint where all four points are visible, regardless of relative orientation of the camera and the object.

Each cross-ratio indicates that a set of four points in the image can correspond to one or more sequences of four collinear points in a model. If there are multiple models, then only models with similar cross-ratios to the detected one need to be considered. Therefore, each cross-ratio hypothesizes that the object in the image can match one or more models in the database (those models that have cross-ratios equal to the detected one).

Although cross-ratios make good features and aid in object identification, by their very nature they are useless for pose determination. Since the cross-ratio is invariant to the elements of the pose (translation and rotation), the pose cannot be determined from the cross-ratio. Therefore, once a model has been identified and correspondences have been hypothesized, alternative techniques must be used for pose determination.

2. Finding Local Correspondences

There are two aspects to finding local correspondences between the image and the model: finding appropriate features in the image, and determining the degree to which each image feature corresponds to a model feature. The first uses image-processing to locate points (in this case point sequences) of interest. The second uses matching (fuzzy compatibility) to determine to what degree each point sequence matches each model feature.

Finding cross-ratio features in an image is not significantly different from finding vertex features as used

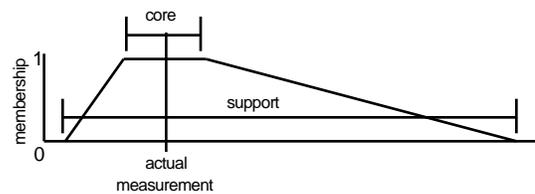


Figure 2: Fuzzy set for image length

in [8]: first standard line detection algorithms find line segments in the image, then those segments are grouped into longer line segments and junctions (vertices) [5]. Each sequence of four consecutive junctions on the same line segment is a feature whose cross-ratio can be determined.

Image and model features correspond when the constraint or measurement associated with the features are the same. Because images contain noise, occlusion, and spurious features, and because the feature extraction process itself is imperfect, it is very unlikely that the image and model measurements will be identical. Therefore, we model each measurement as a fuzzy set whose shape reflects the likely variation of the measurements. With Euclidean measurements, it is easy to develop such sets. As an example, Figure 2 shows a fuzzy set for the Euclidean length measurement, based on the actual measurement. The set is biased toward lengths that are longer than the measured length, because occlusion can shorten a segment's apparent length, but never lengthen it.

When using cross ratios, the effect of occlusion is not as straightforward. Consider a simple case where point D (in figure 1) is occluded. When the segment is detected, the four points will be A, B, C and a point E that results from the occlusion. E will fall somewhere between C and D, so $CE < CD$ and $BE < BD$. If we define k as the portion of segment CD that is visible ($k < 1$), then $CE = kCD$ and $BE = BC + kCD$. Therefore, the cross ratio can be expressed as: $(AB)(CD)k / (AC)((BC) + k(CD))$. The effect of occlusion depends not only on the amount of occlusion (k), but also on the relative lengths of segments BC and CD. As BC approaches 0, the cross ratio approaches the cross ratio of the unoccluded segment. As BC gets larger, the cross ratio becomes smaller. Therefore, the shape of an appropriate fuzzy set for cross ratio will be similar to that for image length, but the degree to which the support exceeds the core on the upper side will depend on the distance between points B and C. If there is no occlusion (as in the model measurement), then the set can be symmetric.

Each object model will contain a cross-ratio for every set of four collinear points on the model. When a cross-ratio is measured in the image, its degree of compatibility with every ratio of every model is recorded. This information is used for both object identification and pose determination.

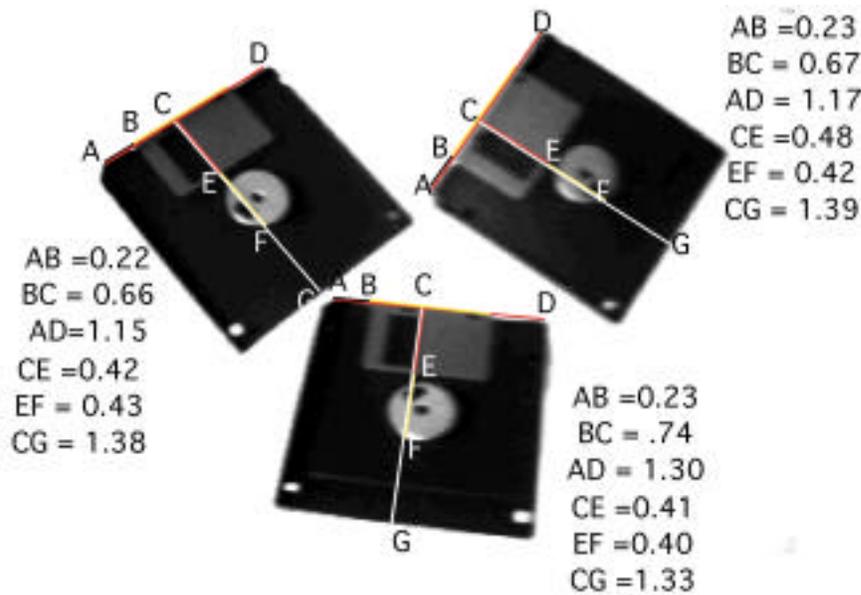


Figure 3: Cross ratio example

3. Object Identification

If more than one model is possible, the local correspondences can aid in object identification. Essentially, each membership of a feature in an object is a weighted vote for that object. These votes need to be constrained by the fact that a each point set can only correspond to a single feature, so for object identification, only the strongest membership of each feature in each object is considered. As an example, consider two objects with two features each. Object A has features A1 and A2, and object B has features B1 and B2. Point set F1 has memberships A1: 0.8, A2: 0.2, B1: 0.6, B2: 0.6. Point set F2 has memberships A1: 0.2, A2: 0.8, B1: 0.3, B2: 0.3. For model A, point set F1 contributes 0.8 support and point set F2 also contributes 0.8 support. For model B, point set F1 contributes 0.6 support and point set F2 contributes only 0.3 support. Therefore, model A is the model most compatible with features F1 and F2.

In addition to considering individual features, it is also helpful to consider pairs of features where appropriate. Specifically, pairs of cross ratios that share a point, such as features ABCD and CEFG in Figure 3, can be linked as a binary constraint: the object containing the pair of line segments is supported as the minimum support of the two cross ratios (using minimum as the fuzzy AND operator). Feature pair constraints are more specific than individual feature constraints, so they are weighted more heavily in object identification.

4. Pose Determination

The goal of pose determination is to find the three-dimensional position and orientation of the object that

appears in the image, given the correspondences and object identification determined earlier. An alternative way to consider the problem is as a stereo problem: given a set of correspondences between two views, determine the three-dimensional structure of the points in the image. We do not know the relationship between the cameras in advance, so we cannot take advantage of constraints other than the point correspondences.

In the context of uncalibrated stereo, it has been shown [3, 6] that eight point correspondences are sufficient to recover the three-dimensional structure of the scene up to a projectivity. The eight-point algorithm [4, 6] recovers the fundamental matrix, which entirely captures the relationship between the cameras, from which the 3D scene is reconstructed. In the case of pose determination, the fundamental matrix itself serves as a description of the pose. Since the model view is fixed, the matrix depends only on the pose of the second camera, or alternatively, on the pose of the object with respect to the second camera.

As in the two-dimensional case, point sets that arise from the same object should yield the same pose, and point sets that arise from different objects should yield different poses. Thus, correct object hypotheses will appear as clusters in pose space, and incorrect hypotheses as isolated points. The pose space for three-dimensional recognition is of much higher order (7 degrees of freedom, because the fundamental matrix has determinant zero) than the equivalent pose space for two-dimensional recognition (3 degrees of freedom). Initially, we will use a simple distance metric in the 7-dimensional space defined by Csurka et. al. [1]. After fuzzy C-Means clustering [2], the strongest clusters, i.e. those fuzzy sets with highest cardinality, will correspond to the best object hypotheses.

If the space is too sparse, it might be necessary to adjust the distance function to take into consideration the covariance matrix of the fundamental matrix [1].

5. Example

The image shown in Figure 3 contains 3 floppy disks, essentially identical. One disk is lying flat, one is tilted horizontally, and one is tilted vertically relative to the camera. As can be seen in the bottom disk, the tilting causes noticeable skewing in the image: the perpendicular edges of the disk do not appear perpendicular in the image. Thus, even if the edges were extracted perfectly, the features of the three disks would not match.

For each disk, two cross-ratios were computed using seven points labeled A through G on each disk. Points A through C are along the edge of the disk containing the slider that exposes the disk. Point A is the corner of the disk that is cut off. Point B is the left edge of the slider. Point C is the right edge of the window in the slider, and point D is the far corner of the disk. Points C, and E through G are along the line segment that extends the right edge of the slider. Point E is the boundary of the slider and point F is the second intersection of the segment with the circle. Point G is the far edge of the disk. To increase accuracy, points were extracted and measurements were taken in a magnified image.

The results of the measurements are recorded in tables 1 and 2. Each table shows the three measurements taken directly from the image, the cross ratio, and the percentage of error, computed by subtracting the cross ratio from the average cross ratio and dividing by the average cross ratio. For the first cross ratio $X(A,B,C,D)$ the error remained below 1% in all cases. The second cross ratio $X(C,E,F,G)$ was somewhat worse, but even there, the worst error was only 3.27%. As a comparison, when length error is computed in the same manner, AD has 8% error, and CE has 10% error, which is significantly larger than errors in the cross ratio. It is clear from these results that the cross ratio features can be matched more accurately than length features in this image, where objects are at different three-dimensional orientations.

6. Conclusion

In this paper, we have described a system for three-dimensional object recognition using a projective invariant and fuzzy clustering. The cross ratio of four collinear points, a projective invariant, is a useful feature for object recognition. Four-point sets are easily detected, and the ratio is easily computed. Because the cross ratio is a projective invariant, values in different images of corresponding segments are similar. Therefore, the ratio can be used to match four-point sets between images and object models. We have hypothesized an appropriate fuzzy set for modeling the uncertainty in these matches due to occlusion. The cross ratio can also aid in object identification, but not pose estimation. Pose hypotheses can be made directly from point correspondences, using the

Table 1: Cross ratios for points A,B,C,D

	Disk1	Disk2	Disk3
AB	0.22	0.23	0.23
BC	0.66	0.67	0.74
AD	1.15	1.17	1.30
X(A,B,C,D)	0.07258065	0.07340426	0.07312843
% Error	0.63%	-0.50%	-0.12%

Table 2: Cross ratios for points C,E,F,G

	Disk1	Disk2	Disk3
CE	0.42	0.48	0.41
EF	0.43	0.42	0.40
CG	1.38	1.39	1.33
X(C,E,F,G)	0.28609769	0.28717949	0.2860979
% Error	3.27%	-1.83%	-1.44%

8 point algorithm, and correct hypotheses form clusters in the 7-dimensional pose space. Experiments are planned to show how well these algorithms will work in practice.

7. References

1. G. Csurka, C. Zeller, Z. Zhang and O. Faugeras, "Characterizing the Uncertainty of the Fundamental Matrix," *Computer Vision and Image Understanding*, vol. 68, No. 1, October 1997, pp. 18-36.
2. J.C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *J. Cybernetics*, vol. 3, 1973, pp. 32-57. Reprinted in *Fuzzy Models for Pattern Recognition*, J.C. Bezdek and S.K. Pal, eds., IEEE Press, 1992.
3. O.D. Faugeras, "What Can Be Seen in Three Dimensions with an Uncalibrated Stereo Rig?," in *Proceedings of the 1992 European Conference on Computer Vision*, 1992, pp. 563-578.
4. R. I. Hartley, "In Defence of the 8-point Algorithm," in *Proceedings of the 1995 International Conference on Computer Vision*, 1995, pp. 1064-1070.
5. H-B. Kang and E.L. Walker, "Characterizing and Controlling Approximation in Hierarchical Perceptual Grouping," *Fuzzy Sets and Systems* vol. 65, August 1994, pp. 187-223.
6. H.C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature*, no. 293, 1981, pp. 133-135.
7. J.L. Mundy and A. Zisserman (eds.), *Geometric Invariance in Computer Vision*, MIT Press: Cambridge, MA, 1992.
8. E.L. Walker, "Fuzzy Relations for Feature-Model Correspondence in 3D Object Recognition," in *Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society*, June 1996, pp.28-32.
9. E.L. Walker "A Fuzzy Approach to Pose Determination" in *Proceedings of the 1997 Conference of the North American Fuzzy Information Processing Society*, Syracuse, NY, Sept. 1997, pp. 183-187.
10. I. Weiss, "Geometric Invariants and Object Recognition," *International Journal of Computer Vision*, vol. 10, no. 3, 1993, pp. 207-231.