Implicit model matching as an approach to Three-Dimensional Object Recognition.

Kenneth M. Dawson,
David Vernon,
Department of Computer Science,
Trinity College, Dublin
Ireland

Abstract

A general purpose approach to 3D model matching is described in the context of object recognition for a computer vision system. The mechanism is independent of the type of model used.

A 3D model is extracted from a series of images of the same scene using shape from stereo and motion techniques. Stored object models are compared with the extracted 'scene model' in a three stage process. Firstly, feasible object poses are determined through the use of histograms of 3D orientations (i.e. Extended Gaussian Images). In the second step, possible object positions are calculated using, as an initial estimate, the 'scene model' position within the image and then, 'tying' appropriate significant 3D vertices together between the two models (in order to fully determine position). Finally, to accomplish object recognition, resultant object poses/positions are compared with the 'scene' model through point-by-point comparison of $2\frac{1}{2}$D sketches.

1 Introduction.

Matching is a complex computationally expensive problem. In the domain of three dimensional object recognition, 'standard' approaches (such as graph or

tree matching) encounter much difficulty. As long as an object model consists of complex component parts (ie. surface or volume primitives), it is extremely unlikely that subsequently extracted components will ever be identical to those of the original model. Hence simple graph matching methods have to be made extremely adaptive to be able to function within the 3D object recognition domain. This leads us to a search for another approach.

The approach detailed in this paper is one of 'implicit matching' as it is independent of the object representation. Instead of comparing object models directly, representations of the *models* are used:

- Extended Gaussian Images: These are used in order to build histograms of the three dimensional orientations of an object model.
- $2\frac{1}{2}$D Sketches: These are simply 'images' of object models from a given viewpoint with every point representing a three dimensional orientation vector rather than the scene illuminance/reflectance.

2 Object Pose estimation.

Extended Gaussian Images (EGIs) [5,6,7] of objects are spatial histograms of the objects surface normals (i.e. 3D orientations). In order to make the EGI as evenly quantised as possible (for the sake of mapping 3D orientations to it in a uniform manner) the tessalations of the EGI are defined by the planar surfaces of a regular polyhedral model of a sphere[1] (in which the surfaces area as similar in shape and size as possible). The largest perfect geodesic (ie. polyhedron with identical surfaces) is the dodecahedron (which has twelve pentagon surfaces) and so this is normally used as the starting point in the generation of the sphere. The dodecahedron can be broken down into smaller surfaces in an iterative fashion (ie. firstly each pentagon is broken into five triangular surfaces[2], and then the triangular surfaces are themselves broken into four triangular surfaces[2] as many times as necessary[3]).

[1] A 3D orientation is mapped to the sphere tessellation through which it would pass if placed at the center of the sphere.

[2] Common apexes are 'raised' to the surface of the sphere.

[3] The number of times the triangular surfaces are broken down into four component triangular surfaces determines how accurately the EGI represents a sphere, and hence how accurately the three dimensional orientations are quantised.

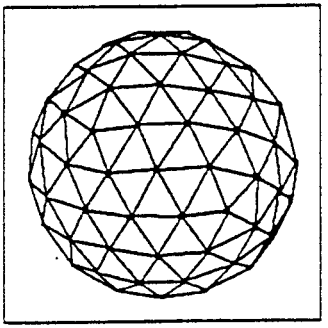

Figure 1: The front view of an Extended Gaussian Image with 240 triangular surfaces.

Assuming that we can obtain a 3D model for a potential object from images of a scene[4], and that we have stored 3D models of possible objects, it is a simple matter to generate EGI histograms for the 3D orientations of both. To determine object poses which are possibly present in the 'scene model' we must rotate the EGIs representing the stored object models so that they correspond to that of the 'scene model'. Now, the EGI generated from the 'scene model' is not complete as it only represents visible surfaces within the scene. Both occluded surfaces (i.e. those surfaces which face the viewpoint but are not visible due to occlusion by another surface), and those surfaces which face away from the viewpoint are not included. To solve these problems, we must consider only the visible hemisphere of the EGIs (as this will allow surfaces not facing the viewpoint to be ignored), and only attempt to match all orientations from the 'scene model' EGI with those from the stored object models and not vice versa (as this deals occluded surfaces, by allowing them not to be matched).

Matching of the EGIs is basically a template matching problem. Both position and orientation of the template are variable so the simple approach of trying all possibilities would be computationally too expensive. Instead some method of improving the efficiency of the matching process is needed. For naturally

[4]The subject of three dimensional model extraction is not dealt with in this paper, as it is not important what type of model is extracted. (The recognition method does not makes use of the object model, but rather it exploits secondary representations of the model).

polyhedral objects (ie. most man-made objects) the orientations on the EGIs are typically sparse, so it is possible to use significant orientation 'features' (i.e. single strongly weighted EGI tessellations or groups of tessellations) to guide the matching of two spheres. The simplest version of this is that of using the three most strongly weighted tessellations from the 'scene' model and tying them to all possible appropriate tessellations from the object model EGIs.

3 Determining object position.

Having determined a possible object pose, it is necessary to determine its position in space (as the matching of EGIs only allows pose to be estimated). If the object in the scene is not occluded, then the position that the object model should be modelled in, can be determined quite simply by using object boundaries as viewed in the image of the scene (or even general position information such as centroid, visible area, etc.). If the object is occluded (or could be occluded) then another mechanism is necessary. Again a simple approach has been taken (although it should be possible to improve this approach) in which significant vertices are used to tie the two models together. In fact, a single vertex would suffice, but the use of several has the advantage of minimising possible errors. It is quite possible that at this stage several different positions of the object may be equally feasible for a given object pose, and in such an instance it is necessary to calculate a degree-of-fit for each position; this is addressed in the next section.

4 Object Recognition - Calculating a final degree of fit.

Having estimated an object pose and position which could represent the 'scene' model, it is necessary to calculate a degree of fit between the two models[5]. Obviously there is little point in returning to graph matching approaches, as the problems associated with them still apply. Instead, an iconic matching strategy is

[5]The comparison of Extended Gaussian Images alone is insufficient to determine an object model match (ie. recognition) as a given EGI histogram may represent many different views of the various possible object models (See figure 2).

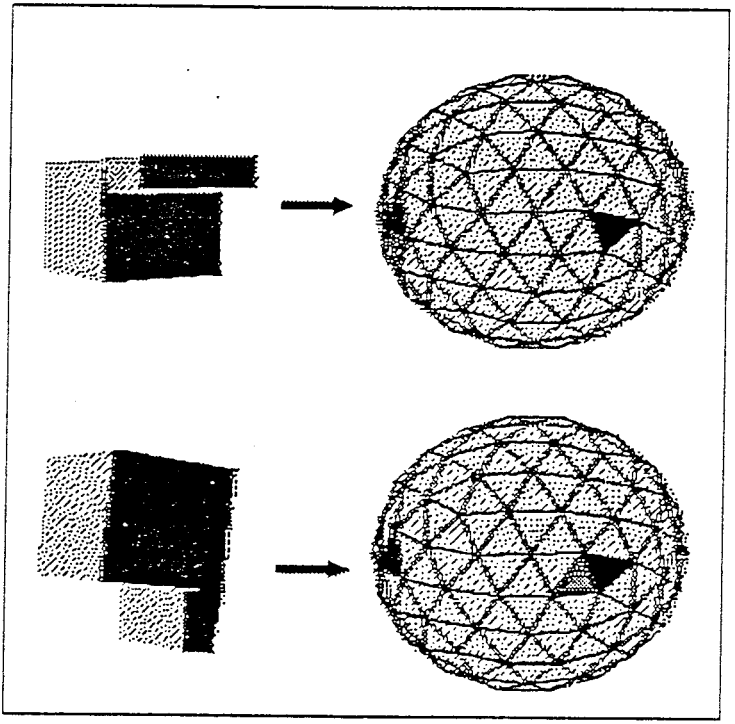

Figure 2: 3D orientations quantised from two different views of an object model onto Extended Gaussian Images. Note how the EGIs are extremely similar, although the views are obviously visually very different.

used in which $2\frac{1}{2}$D sketches are reconstructed[6] of the two models to be compared using the camera model relevant to the view of the 'scene' model being used.

The two $2\frac{1}{2}$D sketches are compared by a simple point-by-point comparison (in effect, global template matching with only one possible position of the template, as position and orientation of the object have already been determined, and as the same camera model was used to generate both $2\frac{1}{2}$D sketches). This comparison implicitly checks each visible object surface and, by using cross-correlation[8] as the measure of similarity, results in a single overall degree of fit between the 'scene model' and the view of the stored object model[7].

5 Advantages, Disadvantages and Conclusions.

The matching approach presented may be used with 'any' type of 3D model, and in fact it is not even necessary for the model extracted from the scene to be of the same type as the stored models. These factors obviously allow flexibility within the system and also allow the method to be used as a testbed for various different object model types. On the other hand the method is constrained as it is most appropriate for naturally polyhedral objects, because of the use of vertices, although it should be possible to use other object features in a similar fashion.

The situation involving multiple objects has not yet been adequately addressed and it has been assumed that it will be possible to segment potential objects within a scene, on the basis of features such as spatial proximity, similarity of surfaces, grey levels, etc. While this is a major assumption, the recognition method described does allow for the recognition of occluded objects. Thus if part of the object can be segmented successfully from the scene it should still be possible to perform recognition, and perhaps improve the segmentation during

[6]It is quite possible that only a $2\frac{1}{2}$D sketch of the stored object model need be reconstructed, as a $2\frac{1}{2}$D sketch of the 'scene' model may well have been determined during the extraction of the model.

[7]It should be noted that, in order to allow for a partial scene model, it is necessary only to consider points in the two $2\frac{1}{2}$D sketchs for which the scene model generates a surface normal. In fact, at this stage it is reasonably simple to check as to whether or not any more of the scene (given that we have or can generate a $2\frac{1}{2}$D sketch for the entire scene) corresponds to the proposed object model, and hence improve the confidence in the model while, at the same time extending the 'scene model'.

the comparison of $2\frac{1}{2}$D sketches (given that a $2\frac{1}{2}$D sketch for the entire scene can be generated).

Perhaps the most useful tool described in this method is that of the comparison of $2\frac{1}{2}$D sketches. This provides a simple method of determining a degree of fit between two object models (or at least between views of them), and allows any differences between the models and any occlusions of the object in the scene to be identified and investigated.

It should be noted that to date we have not attempted recognition using models extracted from real images, as the 3-D polyhedral models are not yet sufficiently accurate. The theory of the method has been tested however using arbitrary views of a stored object model, and did perform well. Essentially this paper describes the current state of our research on this topic, rather than a tried and tested method. However, with improved camera calibration and more reliable techniques which yield structure from motion, we should be in a position to pursue this empirical investigation in the near future.

References

[1] D. Marr, "Vision.", W.H. Freeman and co., San Francisco, 1982.

[2] G. Sandini, and M. Tistarelli, "Recovery of Depth Information: Camera Motion Integration Stereo.", Internal Report, DIST, University of Genoa, Italy, 1986a.

[3] G. Sandini, and M. Tistarelli, "Analysis of Camera Motion through Image Sequences." in "Advances in Image Processing and Pattern Recognition.", V. Cappellini and R. Marconi (editors)., Elsevier Science Publishers B.V., (North-Holland), pp.100--106, 1986b.

[4] D. Vernon and M. Tistarelli, "Using Camera Motion to Estimate Range for Robotic Part Manipulation.", (to be published in the) IEEE Journal of Robotics and Automation.

[5] K. Ikeuchi, "Determining Attitude of Object from Needle Map using Extended Gaussian Image.", MIT, AI Lab, AI Memo No. 714, 1983.

[6] P. Brou, "Using the Gaussian Image to Find Orientation of Objects.", The International Journal of Robotics Research, Vol. 3, No. 4, 1984.

[7] B.K.P. Horn, "Extended Gaussian Images.", Proceedings of the IEEE, Vol. 72, No. 12, December 1984.

[8] D. Vernon, "Machine Vision.", (to be published by Prentice-Hall), 1990.