MATCHING FEATURES FROM EDGE-PROCESSED IMAGE SEQUENCES

M J Stephens
Plessey Research Roke Manor

ABSTRACT

This paper discusses the problem of matching two edge-vertex lists, extracted from two similar images after edge processing. The matching is intended to provide feature-flow information for a Structure From Motion algorithm. The matching is primarily via vertices, but the topology of the structures is used to ensure consistent vertex matches, and to achieve matching of complete edges. Results are shown for real images of an indoor scene.

1 INTRODUCTION

Understanding three-dimensional (3D) scene geometry from a sequence of images requires careful selection and management of the information they offer. Many techniques are conceivable, offering different trade-offs between complexity of implementation and detail of 3D scene representation. A suitable technique for computer vision must provide a compact representation, which is robust and easy to update as further information is acquired from subsequent images.

Our approach to Structure From Motion is via feature matching, exploiting edge and vertex information in images. This provides a list-based representation of a scene, which meets the above criteria, and which maintains 3D information for features closely related to the real world.

This paper discusses the problem of matching (finding the correspondence of) two edge-vertex lists, which is required in order to provide observed feature motion from image sequences. The observed feature motion is then available to cast features into 3D, and later update their positions. The problem is restricted to static scenes incorporating corners and generally straight edges.

Section 2 briefly describes the information available to the matcher from edge-vertex decomposition of an image. Section 3 covers the two phases of matching (initial vertex matching and topological matching). Finally, section 4 discusses other aspects of feature matching.

2 EDGE-VERTEX DECOMPOSITION

Edge-vertex decomposition operates independently on each image, and outputs a list of edges and associated vertices. Vertices are located at edge end-points, sharp bends and junctions. They are classified according to their degree - the number of attached edges. For example, edge end-points are degree one vertices, bends are degree two and T-junctions are degree three.

The edge-vertex decomposition is a list created by analysis of an image resulting from edge detection. The requirements for the edge detection are: accurate location of edges and vertices, low rates of edge drop-out, and consistency from image to image. The first two requirements arise because the features representing the scene must correspond closely to real-world features, if 3D scene geometry is to be understood. The third requirement arises because features have to be matched from image to image. The temporal consistency of real features (in a static scene) means that the decomposition will tend to be consistent, if it produces accurate edges and vertices.

Unfortunately, the stated requirements are severe, when compared to the level of achievement of local edge detectors. Typically, there is an unsatisfactory trade-off between signal-to-noise ratio and accuracy of vertex extraction. Operators such as those of (Canny, 1983) and (Spacek, 1985) provide good performance on isolated edges, but may fail at vertices, for example where a weak vertical edge meets a strong horizontal edge.

Our approach has been to use a minimum of smoothing in order to preserve accuracy, and to employ a second, heuristic stage which aims to remove spurious short edges, and to repair junctions. Figure 1 shows an example image, viewing a box standing in a corridor. Figure 2 shows a thresholded edge image, after heuristic improvement.

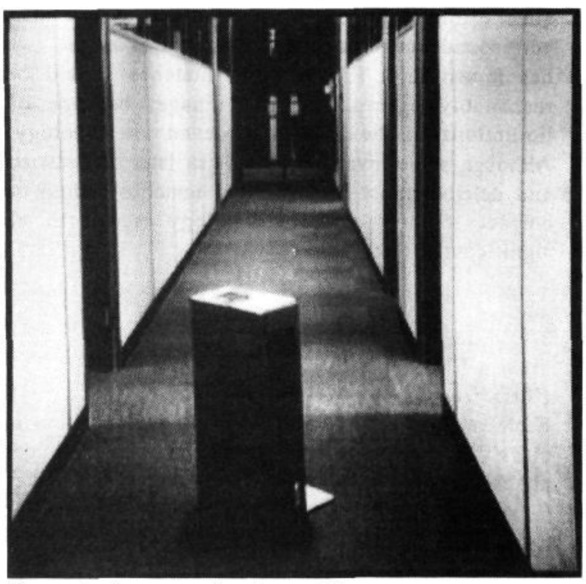

Figure 1. An example of a real image.

A second image, not shown here, views the scene after translating the camera about 15 cm backwards. The resulting edge image is similar to that of figure 2, but the matching process must be robust against some level of inconsistency in edge-vertex connectivity.

AVC 1987 doi:10.5244/C.1.25

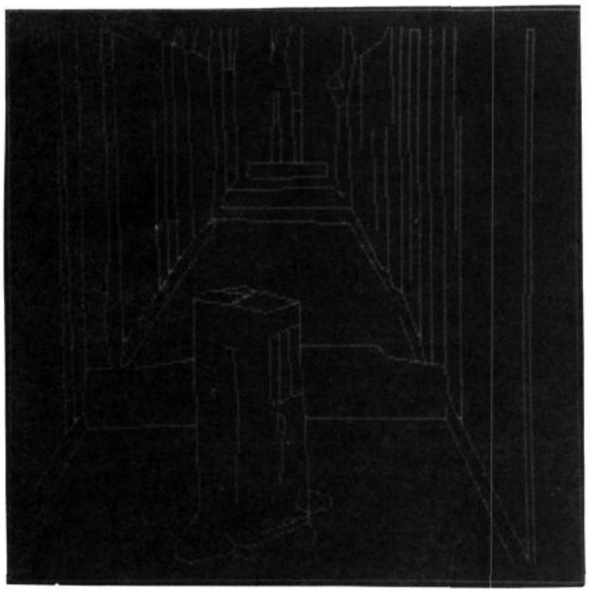

Figure 2. Thresholded and improved edge image of figure 1

3 VERTEX, DIRECTION AND EDGE MATCHING

The information extracted from each single image by edge-vertex decomposition is in the form of a list of edges and associated vertices. The vertices are single pixels labelling junctions, sharp deviations and end-points of edges. The matching of two edge-vertex decompositions (from similar images) is achieved primarily via the vertices, as they are more reliable than edges, which may fragment between images. However, the topology of the decomposition (that is, the directions of edges at vertices, and vertex connectivity) provides a powerful test of the global consistency of the vertex matches, and also allows matching of new vertices and complete edges.

Before topology may be used, some initial vertex matches must be obtained, in order to establish correspondences between sub-structures in the two decompositions. These initial matches should be reasonably spread over the image, because of limitations in the growth achieved via topology. Although some invalid matches can later be detected and deleted, most of the initial matches should be correct if the perceived topology is not to be significantly distorted.

3.1 Initial Vertex Matching

Classification of Vertices

Each vertex from the edge-vertex decomposition is classified by its degree - the number of edges attached to it. Vertices of different degree are treated separately during initial matching. Vertices of degree one are not considered at all, since positions of edge end-points tend to be dictated by detector drop-out and noise, rather than by real scene structure. Vertices of degree two are not considered as primary vertices, because positions of bends in edges are similarly ill-defined.

Vertices of degree three are examined first, being the simplest vertices which are likely to correspond to real structure. These are first matched to other vertices of degree three (from the other decomposition). Then, remaining vertices of degree three are matched to vertices of degree two, in order to allow some tolerance for failure of edge extraction near a vertex. Vertices of degree higher than three are rare, but may be treated similarly to those of degree three if they occur.

Vertex Attributes

Vertices of appropriate degree are matched according to similarity of attributes; quantities which, ideally, discriminate between different vertices, but are similar for the same vertex in different images. The most important vertex attributes in this context are the directions (absolute angles on the image-plane) of attached edges. This assumes that, at least for the first pair of an image sequence, the edge directions on the image plane remain approximately constant.
In order to measure a unique match of direction attributes for a given vertex pair, a particular correspondence of the edge directions must be chosen. The best choice is that which yields minimum total mismatch in these directions. In this manner, matching of edge directions is performed at the same time as vertex matching. The direction matches are important in the topological phase, as discussed in section 3.2.

Another possible vertex attribute is the length (on the image-plane) of an attached edge. However, edge drop-out means that apparent edge lengths can change abruptly from image to image, so length is avoided as an attribute. The other attribute which has proved useful is the local vector image gradient, which is available from the edge detector. However, this is used only as a supplementary measure to the primary mismatch, based on edge directions.

Use of Prior Knowledge

Because the vertex attributes are imperfect, match success rate is improved by reducing the number of candidate vertices which have to be considered for a given match. This may be achieved on the basis of image-plane proximity: only those vertices which are within a given neighbourhood are considered as candidates. Naively, this region could be chosen as a circle of sufficient radius to encompass expected feature flow. However, if some knowledge of the camera motion, which causes feature flow, is available then a more appropriate region may be defined. This is similar to the use of motion estimates to constrain expected optical flow fields (Thompson and Pong, 1987), but is applied to discrete features.

Consider first that a good estimate of the 3D position of a vertex to be matched is available. Uncertainties in the six degrees of freedom of camera motion are assumed to follow a multivariate normal distribution, and hence may be specified by an appropriate covariance matrix. The uncertainties can be projected onto the image-plane as uncertainties in the match position of the vertex, and the normal form retained. The region in which candidates for the match must fall is then specified by a given Mahalanobis distance (for example, within the 3-sigma contour of the distribution function). In general, this will result in an elliptical search region.

If a good estimate of the range of a vertex is not available, then the above region should be extended. Consider the variation of the elliptical region as the assumed range of the vertex is varied between minimum and maximum limits. A new region will be swept-out on the image-plane, which defines the allowed candidate positions.

Finally, consider an uncertainty associated with the position of the observed vertex. This will give rise to a normal distribution on the image plane. This is combined with the camera motion uncertainty by combining the image-plane covariance matrices.

These considerations are described in more detail in (Harris and Pike, 1987). In practice, a simple approximate region can usually be chosen, considering only the dominating factors. In the case of the first pair of an image sequence, where no 3D scene structure is yet known (but an estimate of camera motion is available), the dominating factor is the uncertainty in vertex depths. In this case, the region may be reasonably approximated as a thin 'sausage' shape, which is the envelope of a circular uncertainty (due to observation and motion errors), varied between minimum and maximum depth limits.

Assigning Matches

The intial matches are assigned in a single-pass process, as follows. For each vertex to be matched in one edge-vertex list, candidates are selected from the second list according to a proximity criterion, as discussed above. A mismatch measure is computed for each candidate, by comparing its attributes with those of the first vertex. The candidate with the best (smallest) attribute mismatch is selected as the plausible match from vertex list 1 to list 2. The process is repeated with this selected vertex as the primary vertex, taking candidates from list 1. If the selected candidate in this reverse direction is the same as the original vertex, the match is accepted; otherwise no match occurs.

The process is repeated for each vertex in the first list. It is only modified where no candidates are found in the proximity region, or the attribute mismatch of the best candidate exceeds an allowed threshold - again, no match is assigned. The two-way matching ensures that the final match set is one-to-one, and improves match integrity (since an invalid match is less likely to operate in both directions).

3.2 Edge-Vertex Topology

Once an initial set of vertex matches has been established, the connectivity of vertices via edges may be used to improve and extend the matches. We refer to this as topological matching.

For a given vertex-to-vertex match across two images, traversal to a neighbouring vertex via a given attached-edge direction in one image, and a similar traversal via the same edge direction in the other image, should arrive at corresponding (matchable) vertices in each of the two images. In practice, this is only expected to hold where the attached edges are of similar lengths, since edges are prone to drop-out and fade. Hence, traversals are only made where the difference in edge lengths is less than a few pixels.

Three cases may arise from the vertex-to-vertex traversals. Firstly, the implied topological match between vertices may conflict with existing matches for one or both vertices; the existing match or matches are refuted by topology. Secondly, the implied match may already exist; the existing match is confirmed by topology. Thirdly, the vertices in the implied match may not have been matched before; a new match is indicated by topology. Hence, topology may be used both to delete inconsistent matches and to add new matches. The deletion, or pruning, should be performed first, since errors will be propagated by growing from invalid initial matches.

3.4 Topological Pruning

Topological pruning refers to the deletion of matches which are found to imply a local edge-vertex topology in one image which conflicts with that in the other image. It is not satisfactory merely to delete all matches which are refuted by topology, because a valid existing match will sometimes be refuted by a topological match which itself arises from another, invalid match. The technique used is to accumulate a consistency measure for each existing match, scoring +1 for each confirmation by an implied topological match and -1 for each refutation.

After all consistency measures are accumulated, any existing matches with a consistency below a given minimum are deleted. For instance, if the minimum allowed consistency is zero then all existing matches which score more refutations than confirmations will be deleted. Whilst it may be possible to design more sophisticated techniques which iteratively refine their estimates of consistency, the more simple approach has proved to be effective.

Note that none of the topological matches are saved during the prune phase; they are merely used to accumulate the consistency scores for existing matches.

3.5 Topological Growth and Edge Matching

Topological growth refers to the addition of new vertex matches, implied by edge-vertex topology, to the existing match set. This has the effect of spreading out from matched vertices (where attached-edge lengths are comparable) and matching appropriate vertices which failed to match on attributes. Such failures from the initial match phase occur because of imprecise knowledge of camera motion and vertex depths, and because of imperfect feature attribute measures.

It is possible to repeat topological growth several times, to maximise the final number of matches. In practice, two or three iterations are usually sufficient to exhaust topological additions, since only edges of comparable length will be traversed. During topological growth, where similar length edges have been traversed and the destination vertices are matched (either from the initial phase or by addition), the vertices at both ends of the edges have been matched. Hence, some complete edges are matched during the topological match phase. This is indicated in figure 3, where complete, matched edges are shown as brighter lines.

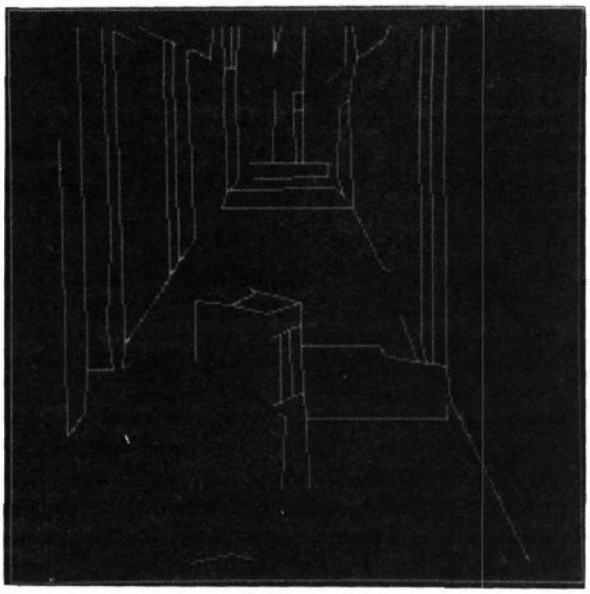

Figure 3. Edge image, with matched edges shown bright.

The effectiveness of the topological phase is also demonstrated by considering vertex matching; nine invalid vertex matches were removed and then the total number of matches increased from 50 to 112, by using topology on the box images.

4 DISCUSSION

The matching process has been described as applicable to edge-vertex decompositions of two images. This is one of the required tasks for initialising a 3D feature list from the first two frames of an image sequence. We call this the Boot Mode, and processing of subsequent frames the Run Mode.

For the run mode, the edge decomposition of a new image must be matched to the projection of the 3D features. However, the vertex matching task is similar in this case, once the projection has been achieved (to create a 2D edge-vertex list). Because previously matched vertices are associated with a good estimate of depth, the proximity regions for candidate matches are much more tightly defined, in general, so that the vertex matching will be more robust. Use of topology will also be more robust, for complete edges which were previously matched, because edge lengths and directions on the new image plane can be found by projection from 3D. The latter assumes an estimate of camera motion is available, but this can be computed from the vertex matches.

Edges present more problems than vertices in the run mode, because of their spatial extent. For instance, an edge which was seen as complete in the first two frames may be partially obscured in later frames. However, in this case the vertex at the visible end of the edge can still be updated in 3D, after successful vertex matching. If the other endpoint vertex becomes visible and is matched later, the complete edge is then effectively updated.

The matching described here operates on images which have been processed by an edge operator. Currently, in order to test the geometrical part of 3D integration from image sequences, we are testing techniques based on point-like features alone. This is discussed in two companion papers; (Harris and Pike, 1987), (Harris, 1987). In this case, edge extraction is replaced by a corner operator, and there is no information on feature connectivity. However, the feature matching then used is very similar to the initial vertex matching (that is, excluding later topological processes) described in this paper. The only difference is that edge directions are no longer appropriate as feature attributes. Instead, local grey-level and gradient attributes are used, which are simply found during implementation of the corner operator itself.

REFERENCES

J F Canny, *"Finding edges and lines in images,"* MIT technical report, AI-TR-720, 1983.

L A Spacek, *"The detection of contours and their visual motion,"* Thesis, University of Essex, 1985.

W B Thompson and T C Pong, *"Detecting moving objects,"* IEEE Proceedings from ICCV 87, June 1987.

C G Harris and J M Pike, *"3D positional integration from image sequences,"* to appear in Proceedings AVC87, 1987.

C G Harris, *"Determination of ego-motion from matched points,"* to appear in Proceedings AVC87, 1987.