PROCEEDINGS

Volume 595

Computer Vision for Robots

O. D. Faugeras, Robert Kelley
Chairmen/Editors

Organized by
SPIE—The International Society for Optical Engineering
ANRT—Association Nationale de la Recherche Technique

2-6 December 1985
Cannes, France

Determining the Pose of an Object

R. M. Dolezal, T. N. Mudge, J. L. Turney and R. A. Volz

Robotics Research Lab
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor
Michigan 48109, USA.

ABSTRACT

We present an algorithm for determining the position and orientation (pose) of an unoccluded three-dimensional object given a digitized grey-scale image. A model data base of characteristic views is generated prior to run-time by merging perspective views containing the same feature points, such as points of sharp curvature in an edge map, into common characteristic views. The run-time algorithm consists of (1) extracting an edge map from the image; (2) locating feature points in the edge map; (3) using intrinsic properties of the feature points in the image, such as signs of curvature, to rank the characteristic views for the object according to their likelihood of correspondence to the image; (4) for each characteristic view in the ranking, matching properties of the image feature points and object feature points in order to generate potential correspondences; and (5) verifying the most likely correspondences by examining a least-squares fit in each correspondence. The fit yields a rotation matrix that defines the pose of the object.

1. Introduction

The ability to automatically determine the location and orientation (the pose) of a three-dimensional object from a digitized intensity image is essential in many tasks of automation, including navigation and parts handling. However, determining the pose of an object is difficult for two reasons: (1) there are six independent degrees of freedom for the pose of the object (three degrees of freedom for rotation, three for translation); and (2) if the resolution of the digitized image is to be sufficiently fine to permit unambiguous recognition of the object, then there is a sizeable quantity of data to be handled during the course of recognition.

To make the problem tractable, the quantity of data must be reduced as early as possible during the recognition process, subject only to the constraint that sufficient data must be retained to allow reliable recognition of the object being sought in the image. One way to accomplish this reduction is to extract an edge map from the digitized image. Extracting an edge map essentially reduces the image data from dimensionality two (i.e., an $M \times N$ pixel array of intensities) to dimensionality one (i.e., chains of pixel coordinates). If there are many edges in the image, however, the edge map still represents a formidable amount of data.

Further reduction of the data can be accomplished by extracting feature points from the edge map. A feature point, which might correspond to a vertex or other point of extreme curvature in the edge map, typically consists of the coordinates of the point in the image as well as some additional information characterizing the point. A particular feature point, for example, might consist of the image coordinates of a point of extreme curvature and a value denoting the radius of curvature. The concept of a feature point generalizes to the concept of a feature point vector, or feature vector, which contains two or more feature points and certain values describing relationships between sets of the feature points. For example, a particular feature vector might consist of an ordered triple of feature points and three values denoting the pairwise distances between the feature points in the image.

If the characteristics of the feature points are robust and their locations are insensitive to noise in the image, then the feature points are convenient starting points for hypothesizing correspondences between an image and a known model of an object. The hypothesized feature point correspondences can be combined to form hypothesized feature vector correspondences. Once a reasonable set of hypothesized feature vector correspondences is obtained, successful recognition depends solely on the robustness of the algorithm for verifying the feature vector correspondences. Since judicious choice of the types of feature points to be used in recognizing a particular object should result in relatively few feature vector matches between the model and the image, the extraction of feature points and feature point vectors represents a significant reduction in the amount of data to be processed during the later stages of recognition.

In the following sections, we discuss a general algorithm which utilizes feature point vectors to determine the location and rotational orientation of an unoccluded three-dimensional object from a digitized grey-scale image.

2. Details of the Current Implementation

2.1. Off-line Data Base Generation

The algorithm depends upon a model data base produced prior to run-time. The data base is generated as follows.

The object is assumed to have an intrinsic Cartesian coordinate system, so that each point on the object has an associated triplet of coordinates. Centered about the Cartesian origin, (0,0,0), is an imaginary sphere which completely encloses the object. We tessellate the surface of the sphere into many regions of approximately equal area. Each region in the tessellation corresponds to a perspective from which the object may be viewed, hence we refer to the regions as *perspective regions*. We specify a perspective region with a pair of angles, elevation and azimuth, similar to the geographic coordinates for latitude and longitude (see Fig. 1). For each perspective region, we obtain the corresponding isometric projection of the object by rotating the object with an orthogonal rotation matrix derived from the appropriate elevation and azimuth angles.

After the rotation, a silhouetting subroutine extracts the edges corresponding to the outline of the object, as it would appear from that perspective. Another subroutine parametrizes the silhouette boundary in coordinates of arclength vs. tangent angle. The parameterized boundary forms a function that is smoothed by convolving it with a Gaussian. From the smoothed function we extract a list of points corresponding to extrema in curvature (see Fig. 2). A point of large positive (negative) curvature in the image corresponds to a point of large positive (negative) slope in the tangent angle-arclength function. Thus, our feature points correspond to points on the silhouette at which the tangent angle is changing rapidly as a function of arclength. Some of the feature points in the silhouette correspond to unique, well defined points on the object; we call these *real* feature points. Other feature points arise because one part of the object is occluding another part; we call these *virtual* feature points.

We therefore acquire a list of feature points as we traverse the silhouette in counterclockwise order. The information retained for each feature point includes the type of feature point (real or virtual), the $< x, y >$ coordinates of the feature point in the plane of the silhouette, the corresponding $< x, y, z >$ coordinates of the object (if the point is real), and the sign of the curvature of the silhouette at the feature point. Finally, after all of the feature points have been identified along the silhouette we calculate the pairwise distances between feature points.

When this procedure has been completed for every perspective region in the tessellation, we merge perspective regions containing identical lists of feature points into larger regions called characteristic views. We borrow the term *characteristic view* from [ChF82], but we have redefined the notion of topological equivalence below. To satisfy the criteria for merging into a common characteristic view, the perspective regions must contain the same feature points in the same order. A characteristic view comprises this list of feature points; the averages of the elevation and azimuth angles for the perspective regions contained in the characteristic view; and, for each pair of feature points, the bounds (taken over all of the perspective regions contained in the characteristic view) on the projected distances between each pair of points. The characteristic views are therefore topological equivalence classes, where topological equivalence is defined in terms of feature point orderings.

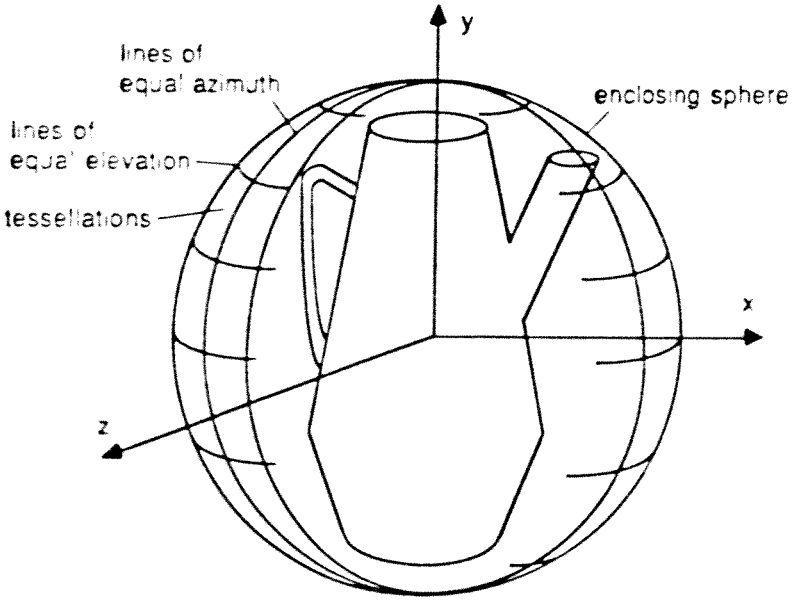

Fig. 1 Object and enclosing sphere.

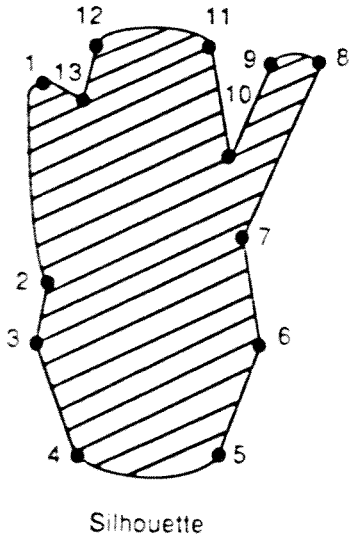

Fig. 2 Silhouette with extreme points.

2.2. Run-time Processing

Given a digitized grey-scale image of the unoccluded object as input, the recognition phase proceeds as follows.

An edge detection subroutine extracts contours from the image. For simplicity, assume that only one closed contour results from the edge detection. Another subroutine parametrizes the contour in terms of arclength and tangent angle, smooths the resulting function, and extracts a list of feature points (extrema in curvature) along the contour. During this phase of the run-time processing, it is impossible to deduce whether the feature points represent real feature points or virtual feature points. The only information available is the ordering of the feature points around the contour, the sign of curvature of each feature point, and the distance between each pair of feature points.

The perspective view of the object in the image must correspond to one of the characteristic views identified during the off-line processing. Since the extrema in curvature should be fairly robust feature points, it follows that the extrema in the image should correspond closely to the extrema in the characteristic view silhouette, of which the image is an example. Therefore, to determine which characteristic views are most likely to match the image view, we compare the number of image feature points of negative curvature and positive curvature to the number of feature points of like sign in each characteristic view. The characteristic views most likely to match the image are those characteristic views for which the numbers of negative and positive extrema most closely match the numbers of negative and positive extrema in the image. This procedure yields a ranking for attempting to match the characteristic views to the image. Since many objects have several dozen characteristic views, ranking the views in this way substantially reduces the combinatoric complexity of the matching.

Once the characteristic views are ordered in this fashion, the most promising of them is examined to see how well its feature points correspond to the feature points of the image. A correspondence between a feature vector (i.e., a set of feature points) in the characteristic view and a feature vector in the image is established by verifying that the following are satisfied:

1. Each feature point in the image has the same sign of curvature as the corresponding feature point in the characteristic view silhouette.

2. The feature points in the image occur in the same order around the contour as do the corresponding feature points in the characteristic view silhouette.

3. Each pairwise distance between feature points in the image lies between the maximum and minimum distances calculated for the corresponding pair of feature points in the characteristic view silhouette. The maximum and minimum distances are softened with tolerance values that allow for error in localizing the feature points in the image.

The purpose of matching image feature vectors to object feature vectors is to match enough feature points so that a rotation matrix can be determined that gives a good fit between the coordinates of the image points to those of the rotated projected object points. A least-squares fit is employed. Since the least-squares fit is computationally expensive, we attempt it only when there is a high probability that the feature vector correspondence is valid.

We define an n-tuple correspondence as a match between a feature vector derived from n points of the object and a feature vector derived from n points in the image. The matching strategy involves generating quadruple correspondences between four consecutive points around the silhouette and four consecutive points around the image. Since there are only $O(n)$ valid quadruple correspondences of consecutive feature points, the matching is fast. The four singleton correspondences with the highest frequencies of occurrence among the quadruple correspondences are then selected for determining the rotation matrix.

Once a quadruple correspondence is selected for determining the rotation matrix, a modified version of the Levenberg-Marquadt nonlinear least squares algorithm is used to calculate the elevation and azimuth angles which are used to form the rotation matrix (see [BrD72]). The convergence parameters of the algorithm determine whether an n-tuple ($n>4$) correspondence that contains the quadruple correspondence should be generated to verify the correctness of the fit. If convergence does not occur, the available options are to attempt another match after selecting a different quadruple correspondence from the high-frequency singleton correspondences, or to select a different characteristic view for matching.

3. Conclusion

We have presented an algorithm for determining the rotational orientation of a three-dimensional object given a digitized grey-scale image. The algorithm employs feature point correspondences to index into a data base of characteristic views for the object, where feature points are defined as extrema in curvature around an edge contour and characteristic views are defined in terms of the feature points which are visible from a particular perspective.

Acknowledgment

We wish to thank Paul G. Gottschalk and Charles P. Jerian for generously contributing software used in the development of the algorithm. This work was supported in part by a grant from The Air Force Office of Scientific Research under contract F49620-82-C-0089 and a grant from the Army Research Office under contract DAAG29-84-K-0070.

4. Bibliography

The following is a brief annotated bibliography of pertinent papers.

[BrD72] Brown, K.M., and Dennis, J.E., "Derivative Free Analogues of the Levenberg-Marquadt and Gauss Algorithms for Nonlinear Least Squares Approximation," Numer. Math. 18, 289-297 (1972). [Discusses finite-difference analogues and local convergence properties of the algorithms.]

[ChF82] Chakravarty, I., and Freeman, H., "Characteristic Views as a Basis for Three-Dimensional Object Recognition," SPIE Robot Vision, Vol. 336, p. 37-45 (May 1982). [Defines characteristic views in terms of line-junction topology.]

[Goa83] Goad, C., "Special Purpose Automatic Programming for 3D Model-Based Vision," Proc. Image Understanding Workshop, p. 94-104 (1983). [Discusses tessellation of sphere into perspective regions; uses intersection of sets of perspective regions to reduce search space.]

[Mar84] Marimont, D.H., "A Representation for Image Curves," *AAAI-84*, p. 237-242 (1984). [Discusses finding an optimal set of critical points to represent a curve, selecting of natural scales for smoothing.]

[SKB82] Stockman, G., Kopstein, S., and Benett, S., "Matching Images to Models for Registration and Object Detection via Clustering," *IEEE Trans. Pattern Anal. and Machine Intell.*, Vol. PAMI-4, No. 3 (May 1982). [Discusses application of Hough transform techniques to problem of matching image features to model features when finding rotation, scaling, and translation parameters.]

[WMF79] Wallace, T.P., Mitchell, O.R., and Fukunaga, K., "Three-Dimensional Shape Analysis Using Local Shape Descriptors," *IEEE Conf. on Pattern Recog. and Image Proc.* (1979). [Discusses matching of feature vectors using angles and distances derived from local feature point information; utilizes global information to determine absolute distances and filter widths.]