MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

Relaxation Based Modeling and Recognition of 3D Surfaces from Range Data

Chang Y. Choo and Nasser M. Nasrabadi
Worcester Polytechnic Institute
Department of Electrical Engineering
Worcester, Massachusetts 01609
U. S. A.

William I. Kwak
Digital Equipment Corporation
500 Donald Lynch Blvd
Marlboro, Massachusetts 01752
U. S. A.

Abstract

Modeling and recognition of 3D objects by surface is an important problem in machine vision. Given a large number of range data points of an object surface, we present a relaxation based technique to construct a piecewise triangular patch model using a small number of knot points. We also describe how the model can be used for extracting surface features and for recognizing the object surface.

1. Introduction

Surface modeling and recognition of 3D objects from range data is an important problem in machine vision because range images provide an insight to the surface geometry and are not distorted by reflections or lighting conditions. A typical approach to this problem may be divided into five consecutive steps [1, 2]. First, local discontinuities, e.g., edges and vertices, are found by applying a local edge operator such as zero-crossings and extrema. Second, the local surface patches defined by the discontinuities and surface orientation are approximated linearly or by second- or higher-order polynomials in 3D space. Third, to find global surface patches, the local patches are merged by using a region growing algorithm. Fourth, the object surface is represented as a graph with nodes and links representing geometric entities such as global patches and their relationships, respectively. Finally, object recognition is performed by graph matching.

The first two steps from the above are known together as the surface segmentation. Many researchers have presented techniques for finding edge lines from range data. They include Inokuchi et al. [6], Langridge [7], and Fan et al. [4]. Often, segmentation of surface points using the edge operator is undesirable or difficult. For example, if surface data points are not given as a matrix, or if they are not uniformly distributed, the edge operator cannot be directly applied.

One of the objectives of surface segmentation is to use as small a number of patches as possible, mainly for data reduction. Another is to make the points and lines of patches align with surface features. It was pointed out in [3] that triangular patch approximations are not so satisfactory with respect to the latter objective. However, the computational advantage of using simple patches is often too attractive to employ other sophisticated techniques.

In this paper, we present an efficient surface segmentation technique based on the idea of relaxation. Also, we present a technique for finding the high-level features from the surface model. We also discuss the construction of graphs from the features. Although it is assumed in this paper that only single-view surface data are given, our technique can be extended to the general case, i.e., 3D neighborhood graph [5], in straightforward manner.

In Section 2, we describe how the surface of a 3D object is segmented and modeled. Section 3 describes a procedure for finding surface features from the triangular patch model. In Section 4, we describe a procedure for constructing a graph for the object surface based on the features found. Section 5 shows some experimental results involving a library of simple 3D objects. Finally, Section 6 contains concluding remarks.

2. Surface Segmentation

The segmentation technique presented in this section is based on relaxation, and searches iteratively and exhaustively for the optimal location of the small number of knot points. Although the technique does not always produce the globally optimal solution, it does find a set of reasonable knot point locations in short time. The technique does not require that the range data be in matrix format. In addition, the surface patches need not be triangular, but may be rectangular or curved. However, the segmentation becomes much simpler by using triangular planar patches, and results in a high degree of data compression.

Suppose that we are given a set S of M surface data points, i.e., $P_i = (x_i, y_i, z_i)$, where $i = 1, \ldots, M$. Each of them represents an object surface point where the z coordinate is the depth value at (x, y). At the finest level, we can obtain a Delaunay triangulation for all the points in S and construct the corresponding triangular patch model. However, because of redundancy of data points and/or limited memory as well as finite computational capacity, we often select a subset V of S and approximate the surface based on V of N knot points. Thus, the objective is to find the optimal placement of $N(\ll M)$ knot points $P_i' = (x_i, y_i)$, $i = 1, \ldots, N$, such that the object surface is best approximated with planar triangular patches formed using the N points. In addition, the triangular patches will allow efficient extraction of edge line features.

Our relaxation based technique works as follows:

1. For a given surface and N, place N points spaced uniformly or in some predefined manner.
2. Perform a triangulation on these N points. If the N points are placed on a uniform grid, the triangulation is implied by the grid. Delaunay triangulation may have to be performed if the N points are not initially placed on a uniform grid.
3. For every polygon made up of a set of triangles with only one internal point P_i, as shown in Figure 1, place the internal point optimally such that the sum of approximation errors of all triangular patches within the polygon is minimized.
4. Go to 3 unless there is no change in the placement of points or unless a predetermined limit of iterations is reached.

Note that the maximum error is used as the measure of error for each triangular patch. The maximum error is defined as the maximum perpendicular distance from the triangular patch to the surface data point.

The time complexity of this technique is given in the following theorem.

Theorem: For a fixed number of iterations, the time complexity of the algorithm is $O(\frac{M^2}{N})$.

Proof: There are M surface points and N polygons. Each iteration checks at most N polygons. A point in a polygon belongs to at most two other polygons because a triangular patch within a polygon is a member of the other two neighboring polygons. In other words, an internal triangle belongs to at most three neighboring polygons. Each iteration, when every polygon is searched for a better internal point, evaluates at most $3M$ points. Each polygon has, on the average, $\frac{3M}{N}$ points. Thus, the evaluation of each point in a polygon takes $O(\frac{M}{N})$ operations because it computes the error on every surface point. Therefore, each iteration takes about $O(3M \times O(\frac{M}{N})) \equiv O(\frac{M^2}{N})$. Note that the time complexity becomes $O(M)$ when N is proportional to M (e.g., $N = \frac{M}{2^4}$). □

Figure 2(a) shows an object surface triangulated on a uniform grid, and Figure 2(b) shows the object surface obtain by our segmentation technique. More examples will be shown in Section 5.

3. Surface Feature Extraction

In this section, we present a technique for finding edge lines of 3D surfaces using the intrinsic information on each triangular patch. The intrinsic information available includes the unit normal vector, side lengths, and area of each patch. Edge lines can be found by assigning to each triangle side, a feature index which is based upon the inner product of the unit normal vectors of the patches which bound that line.

Edge lines of a 3D object surface include valley lines, ridge lines, and cliff lines. By using the maximum error criterion, the points will tend to align with the edges of the object. Patches which fall on either side of an edge will have large angles between their unit normal vectors. The inner product of these vectors is used to locate the edge lines. In the special case of a vertical edge, the points will align themselves on either side equidistant from the edge. Using the inner product, the two bounding lines can be located and the feature is determined to lie directly between them. Detection of edge lines proceeds in following steps:

1. Overlay a uniform grid of knot points onto the surface to be segmented.
2. For the four points defined by every two adjacent triangular patches, pick the triangulation with the smaller error.
3. Apply the segmentation technique until it converges or reaches a predetermined limit of iteration.
4. Repeat Step 2.
5. Calculate a "feature index" to be associated with each line. This index will be the inner product of the unit normal vectors of the two patches which bound the line. Area of each patch may also be considered.
6. Remove all lines with an index value which exceeds a threshold. This will result in a few clusters of line segments.

The process of locating the edge lines is now reduced to a 2D problem. The 2D projection of those triangular patch edges with significant feature indices indicates where the edge lines of the surface lie. These patch edges, which are short lines, must be combined to enable an edge description of the surface as seen from this particular view. To simplify this process, we have assumed that all of the surfaces are made up of straight edges so that the task becomes polygonal approximation.

The polygonal approximation used is based on the rules which considers various situations and applies whichever rule may be appropriate. This method may require many rules and there is a chance of the situation which cannot be dealt with correctly and will require a new rule. Six representative rules are described in Figure 3. These rules are applied iteratively until no changes are made.

4. Model-Based Surface Matching

Once the high-level edge lines of the object surface are extracted, the surface can be described by an attributed connection graph. Since complexity of the graph depends on the number of the high-level edge lines, it is important that the rule-based line connection procedure described in the previous section generate as few edge lines as possible. In the attributed connection graph, a node represents an edge line, and a link connecting two nodes represents the relationships between the two connected edge lines. The node attributes for each edge line include the 3D coordinates of the two end points and the feature index of the edge line. Such attributes as the length, direction, and midpoint of the edge line can be easily calculated from the 3D coordinates of the two end points and need not be stored separately. The link attributes include the angle between the two connected edge lines. An example of the attributed connection graph is shown in Figure 4.

After an attributed connection graph is constructed from an input range data, we should select the model graph which best matches the input graph. This graph matching begins with finding the nodes with a distinguished attribute, for example, the node representing the longest edge line, from both the model and input graphs. The attributes of the adjacent nodes in the input graph are compared with those of the model graph. The cost of matching as a matching metric is the dissimilarities between the model and input graphs with respect to the node and link attributes. While matching the nodes of the model and input graphs, if the matching cost exceeds a threshold, the current model graph is rejected. Otherwise, the matching continues until all nodes and links are tried and the current model is determined as the same as the input surface range data. If the current graph is rejected, then the next model graph is selected and the above procedure is repeated.

5. Experimental Results

We implemented the whole procedure in C on a DEC Station 3100. Experiments were performed on several different object surfaces, each defined by 128x128 surface data points. They are shown in Figure 5. Initially, 184 knot points were placed uniformly and triangulated. Of this 184 points, 144 (N) are internal and can be relocated by the segmentation technique. In all of the cases, the segmentation converged in a few iterations, typically requiring about 90 CPU seconds in the DEC Station 3100, and was independent on the object.

Figure 6 shows the results of the segmentation in the 2D plane. Objects (a) and (b) are made up of planar sections with straight edge features. These edge features are found as described before. The feature index is evaluated for each triangle edge and a threshold is set. All lines with an index above this threshold are removed. In all of these cases, the index threshold was chosen to be 1.0. In other words, all of the lines which were bounded by patches with parallel normal vectors were removed. This is appropriate in this case because we know that the object surfaces are made up of planar sections. For object (d), we examined a histogram of the feature indices to find that many adjacent patches have parallel normals. For this object we again choose the threshold to be 1.0.

The lines remaining after the thresholding are shown in Figure 7. These results show that the edge lines of the objects can be easily extracted from the triangular patch model created by our segmentation technique. These edge lines must be described clearly to be useful for surface description and recognition. The rule-based polygonal approximation discussed earlier is now used and the results for some of the objects are shown in Figures 8.

6. Concluding Remarks

The relaxation based segmentation and the rule based polygonal approximation have shown to be effective and efficient in finding high level edge lines from range data. The experimental results show that our approach is indeed applicable to a class of objects. Construction of graph from the high level edge lines facilitates efficient recognition of objects. Future investigation needs to be done in various stages of the procedure for recognizing objects from range data. For example, while applying the segmentation technique "intelligent" initial placement of the knot points may results in better approximation and less number of iterations.

Acknowledgments

This work was supported in part by Engineering Foundation Grant RI-A-89-12 and the IEEE Life Member Fund Committee.

References

[1] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, 1982.

[2] P. J. Besl, *Surfaces in Range Image Understanding*, Springer-Verlag, 1988.

[3] T. Fan, G. Medioni, and R. Nevatia, "Segmented Descriptions of 3-D Surfaces," *IEEE Trans. Robotics and Automation*, vol. RA-3, no. 6, pp. 527–538, 1987.

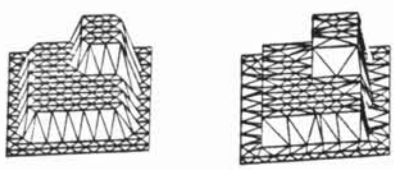
[4] T. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-D Objects Using Surface Descriptions," Proc. Second Intern. Conf. Computer Vision, Tampa, Florida, December 1988, pp. 474–481.

[5] T. C. Henderson, "Efficient 3-D Object Representations for Industrial Vision Systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 6, pp. 609–618, 1983.

[6] S. Inokuchi, T. Nita, F. Matsuday, and Y. Sakurai, "A Three-Dimensional Edge-Region Operator for Range Pictures," Proc. 6th Intern. Conf. Pattern Recognition, Munich, West Germany, October, 1982, pp. 918–920.

[7] D. J. Langridge, "Detection of Discontinuities in the First Derivatives of Surfaces," *Computer Vision, Graphics, and Image Processing*, vol. 27, no. 3, pp. 291–308, 1984.

Figure 1: Polygon with one internal point P_i



(a) (b)

Figure 2: A triangulated object surface based on (a) a uniform grid, and (b) relaxation based segmentation

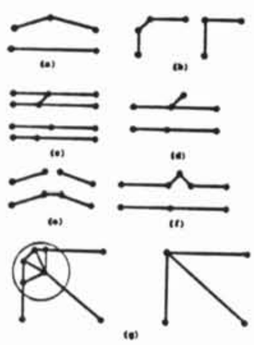


Figure 3: Rules for polygonal approximation

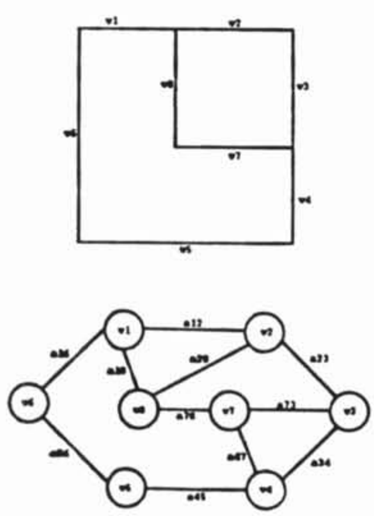


Figure 4: Attributed connection graph

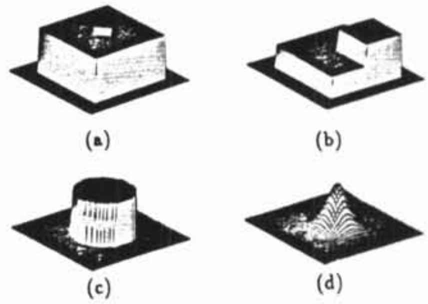


Figure 5: Object surfaces used for experiments

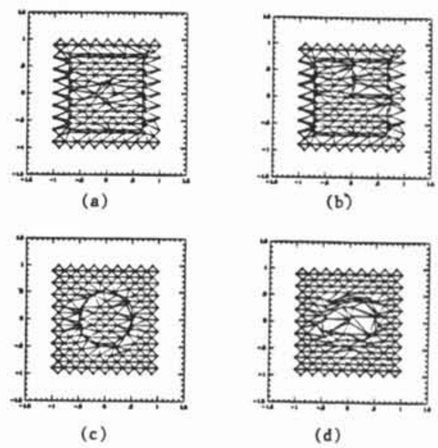


Figure 6: Triangulated results

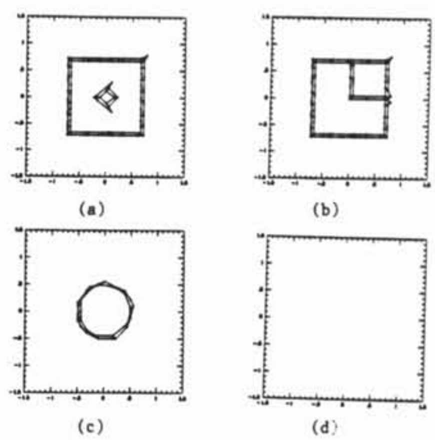
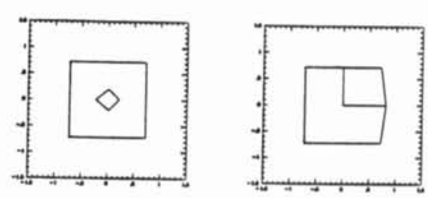


Figure 7: Low-level features



(a) (b)

Figure 8: Global (high-level) features