Extracting Polyhedral Models From A Range Image: A Hybrid Approach

Kamal Kant Gupta*
Xiao Ming Zhu
Simon Fraser University
Burnaby, British Columbia, Canada

In this research, we provide a new hybrid approach to extract 3-dimensional models of the visible surfaces of a polyhedral object in a range image. These models are in the form of 3-dimensional faces, edges, vertices and their geometry and topology (similar to B-Rep models in solid modelling).
Our approach, called "combine and compare" combines the use of edge-based and region-based low-level segmentation to obtain more robust and reliable analytical descriptions of faces, edges and vertices of the polyhedral object than the current purely edge-based approaches. Using region-based information also facilitates a simpler subsequent geometric analysis (for instance, classification of the vertices) than existing pure edge-junction based approaches. We view this work as a step toward the larger goal of obtaining complete B-Rep models of objects in a scene from multiple range images.

Introduction

For many robotic tasks such as automatic path planning and grasp planning, a robot needs a 3-dimensional geometric model of its environment in terms of faces, edges and vertices and their geometry and topology For instance, a basic paradigm in path planning, called the configuration space approach, explicitly computes the constraints on the joint angles of the robot arm [13], [6]. For such computations, it needs the following information (i) which edges form a face, and the outward normal to the face, (ii) coordinates of the vertices. Similarly, to find a grasp position on a polyhedral object, one needs to reason about the relationship among the faces of the object. Such representations are very close to what are called boundary representations (B-Rep) in Solid Modeling literature. A boundary representation is a method of representing the 3-D geometry and topology of the vertices, edges and faces an object [10]. We will also use the term geometric (or polyhedral) models for such representations.

In this research, our aim is to extract boundary representations of visible surfaces of a polyhedral object from its range image. The boundary representation, in our case, is simply the vertices in a counter-clockwise order (with the inside of the face on the left hand side as the edges are traversed) on each face. This simple representation is adequate for tasks such as motion planning and grasping. Note that our methods do recover the face adjacency relationships, however, it is not made explicit in this representation. Furthermore, boundary representation schemes such as winged edge [10] be derived from our simpler representation.

Figure 1 shows the input to the system, a range image, and Figure 2 gives the output, a boundary representation – each face of the object is listed along with the vertices (their x, y, z co-ordinates that form the face). The vertices are listed in counter-clockwise order with the inside of the face on the left hand side as the vertices are traversed.

There are several reasons for restricting our domain to a polyhedral world. First and foremost is that although a great deal of vision research has been devoted to representing shape in general and [12], e.g., generalized cones, Superquadrics, etc., there exists no single shape respresentation that is suitable in all situations. In fact, there has been such a gap between CAD models and vision representations of shape that CAD-based approaches to build representations that are particularly suitable for object recognition are themselves an active area of research [3]. Our main motivation comes from the robotic domain. Most industrial objects are polyhedral; others that are not, can be approximated as polyhedra as far as planning collision-free paths is concerned. Second, most planning algo-

*This research has been supported by a grant from the Centre for Systems Science, Simon Fraser University. Some previous publications of the first author have appeared under the name Kamal Kant.

rithms assume that the objects in the robot's environment are polyhedral [13]. Furthermore, representation of polyhedral objects is fairly well understood in the solid modeling literature[10]. Hence, our choice of polyhedral world. Of course, later these models will be extended to include curved surfaces.

Background and Relation to Other Work

There are two broad sub-areas in our research work: (i) low-level range image processing techniques that extract geometric primitves such as jump edges, roof edges, and planar regions, and (ii) intermediate-level geometric analysis that uses the results of low-level segemntation to extract more symbolic boundary representations. Most vision literature is motivated by object recognition as its goal, and the same is true of recent literature in range images [11], [2], [4], [8], etc. However, a few researchers have emphasized the need for obtaining complete scene descriptions for a variety of other tasks, e.g. [15], [9] for polyhedral objects, and [14] for curved objects. See [7] for a brief review of these approaches.

Most of these approaches to extracting boundary representation adopt the edge-based approaches as the low-level segmentation, i.e., they use edge-detection to extract edge segments that form the basic primitives to be used for the geomterical analysis. This is mainly because these edge-based approaches directly provide the edges and vertices for deriving a boundary representation of polyhedral objects. Infact, even the surface information is then derived from these edge-based primitives. However, inaccuracies in valid edge segment declarations are very common and sometimes almost inevitable. To achieve a boundary representation, especially in order to get the information on which edge belongs to which surface and which edge separates which two regions, accuracy is very important. A small error here may lead to significantly different results later on and may cause wrong inference. Although the information derived from region-based approach is not in an explicit edge-vertex form, our thesis is that region-based information helps us get robust and accurate description of what is in the image, for example, surface (planar, in our case) equations. Our approach uses information from both edge-based and region-based methods. Edge descriptions are derived directly from edge processing (as usual), however, surface information (analytical descriptions of planes that form the faces of the object) is obtained directly from raw image data using region-based approaches. Such descriptions not only help extract more precise edge descriptions (roof edges), but they also help the subsequent geometric analysis to extract valid vertices. In the next few sections, we first give a brief overview of our approach and then give details for each part.

Overview

Refer to Figure 3. The input data is a range image, an array of depth values, $z(i,j)$, with its corresponding binary flag image which indicates where the range image is valid. First, we extract a jump edge map and a roof edge map from the range image. Note that the term map implies that the representation is still at pixel level, i.e., each pixel is characterized as a valid or invalid edge pixel. Next, Hough transform technique is used to extract symbolic jump edge segments. Independently, surface segmentation techniques (based on surface normals) are then used to extract planar regions in the image. Least-squares-fit techniques are then used to determine analytical representation of each planar region. Each of these plane equations are then intersected ("combined") to determine feasible roof edge segments. These roof edge segments are then validated against ("compared") with the roof edge to determine valid roof edge segments. at this stage, we have all the edge sements (roof and jump) in the image. A very similar "combine and compare" method is used to extract valid verices. A crucial point here is that vertex extraction is done on each plane separately. Having extracted the vertices, they are then grouped in a closed loop and ordered in a counter-clockwise fashion (with simple geometric tests) to determine the final description of each face.

There are several novel features in our approach. First and foremost, this combine and compare approach to extract the roof edge segments is quite unique to the best of our knowledge. The main distinguishing feature of this approach is that it uses region-based segmentation to extract roof edge primitives. We believe that this hybrid approach is more reliable, since, if edge processing results in many spurious edges, this method provides a mechanism to select the valid edge segments. Our approach also applies the above "combine and compare" idea to the process of getting vertex information. That is, vertex positions are derived by combining edge functions in all possible ways and then comparing the results with already known information on edges, such as its start and end point positions, to get rid of false vertex position declarations. In addition, this geometric analysis is carried out one face at a time, i.e., thereby greatly simplyfying the analysis. Further, the vertices are classified according to their pixel location and its relationship with the other pixels on the same surface. This is quite different from most existing classifications, such as junction dictionary, which is based on how lines (of different faces) intersect . According to the classification used in this thesis, as few as four types of vertices exist. False vertex declarations are removed by checking to see if its vertex type is allowed or not.

Description and Implementation of Our Hybrid Approach

Low Level Segmentation: Jump Edge Map, Roof Edge Maps, Planar Regions

Our segmentation schemes are simple, however, more sophisticated schemes can be used [1]. Our emphasis is on how the low-level segmentation results, e.g., roof edge map are used later (to extract symbolic roof edge segments) and not on how they are derived.

The jump edge map is derived (using the Sobel edge operator) wherever the edge magnitude exceeds the desired threshold. The gradient of each jump edge pixel is stored for later processing (such as Hough transform).

The roof edge map is derived from surface normal discontinuities [16]. A surface normal map could be calculated using either the gradients in x and y directions, or, by fitting planes in $n \times n$ window. The roof edge magnitude, denoted by M_{roof}, is computed as the maximum angular difference between adjacent unit surface normals. A roof edge is derived wherever the roof edge magnitude (maximum angular difference between adjacent surface normals) exceeds a desired threshold.

For surface segmentation, we group pixels according to their surface normal vectors. Pixels are classified in the same plane if the angular difference between their surface normal vectors is less than a threshold. Furthermore, each region is shrunk by peeling off the boundary pixels (i.e., pixels that do not have all eight neighbours). This shrinking process is iterated a few times (normally, three of four).

Extracting Symbolic Primitives

Jump Edge Segments

Jump edge pixels are grouped into edge segments using Hough Transform with two modifications. First, we extract the startpoint and the endpoint, in addition to the ρ and θ parameters. This, start point and end point description is more explicit and easier to use when trying to find which surface each edge lies on. The other modification is to use a two-step-threshold method. With this technique, we can eliminate much of the noise while maintaining the ability to detect the short edge segments. Each resulting jump edge segment is defined by its 2D and 3D start and end points, and its 3D line parameters.

Planar Equations

We use a least-squares-fit approach to get the analytical equations for the planes corresponding to the faces of the object. In particular, the best fit plane $z = ax + by + c$ on each region is evaluated to get the a, b and c co-efficients.

Roof Edge Segments

The next step is to get a symbolic description of all the roof edge segments in the roof edge map. A novel approach is proposed and used here in order to get a better result than the typical Hough transform type methods. The approach is as follows:

1. Combine all the plane equations (say, n) into groups of two in all possible ways (C_2^n possible combinations). Each combination represents a potential roof edge segment and the (unique) line equation that corresponds to the intersection of the two planes is derived.

2. The potential roof edges are validated by verifying if a significant number of roof edge pixels in the roof edge map (derived earlier) are within a given threshold distance from it. If no corresponding pixels in the roof edge map are found, that line will be declared as a false roof edge segment and will be removed. Also, during this process, the start and end points of the edge are recorded by finding the largest contiguous set of edge points along the line that contains the roof edge segment. Again, each resulting jump edge segment is defined by its 2D and 3D start and end points and its 3D line parameters.

The reason that this method is preferable to the Hough Transform is that very small inaccuracy of roof edge symbolic description derived from the Hough Transform will result in very big error when trying to know which plane they are on. However, in our case, the symbolic edge description is derived from the intersection of two planar equations, which are quite robust, since they have been derived using a relative large set of pixels in the region using least-squares-fit. Now we have got symbolic descriptions of all the roof edges and which two planes form each of these roof edges; all the jump edges, and which plane a jump edge lies on.

Geometric Analysis

Edge Segment Labeling

The edge segments are now examined to determine which surface do they belong to. For roof edge segments, this is already accomplished. The start poin, the end point, and several intermediate points of each jump edge are analyzed to see if they lie on one of the planes. We associate a jump edge with one visible surface. Therefore, only the plane that gives the minimum error is chosen as the plane that the jump edge belongs to.

Extracting Vertices

Next step in our method is to extract vertices explicitly. A distinguishing feature of our approach is that vertex extraction is done one face at a time. We first determine all the edge segments on each plane, and then the following algorithm extracts vertices on each of the plane. Note, however, that the vertex processing is done in 2D, in the image plane, i.e., the 2D projections of the jump and roof edge segments are used intead of the 3D segments.

1. Combine the edge segments (say, m) in all possible combinations to get all potential vertices (C_2^m).

2. Each potential vertex is then classified into one of the following five different categories:

   • valid vertices I that lie within a given threshold of the already known start and end points of an edge. • false vertices I that lie completely outside a region. • false vertices II that lie completely inside the region. • false vertices III that lie on the edges. • valid vertices II. Any potential vertex that does not fall under the previous four categoies.

Figure 4 illustrates four of these cases (For lack of space, we can't illustrate all the cases. Please see [7] for details).

Since the vertices are extracted plane by plane, there are some minor difference among them. Say for example, if a vertex position is actually $(10,10,10)$, on one surface it may be claimed as $(10,9,9.5)$, on the other it may be claimed as $(9.9,9,9.4)$, we have to assign a single value to the same vertex. We call this vertex unification, and all vertices that fall within a threshold distance of each other are coalesced to a single vertex. Now we have a set of edge segments (on each plane), each described by two vertices (end points) that form it. Note that the same vertex may appear in many different planes and may form an endpoint of different edge segments on different planes.

Extracting Oriented Faces

The next step is to obtain the oriented faces, i.e., a closed loop description of the vertices on each plane, traversed in a counter-clockwise direction. This provides the full description of an oriented face, and is achieved as follows:

1. Start randomly with a vertex on a plane. Pick either one of the edges that formed this vertex to be the next edge. Traverse this edge to the other vertex that forms this edge. Continue this way until the starting vertex is reached. The edges have thus been ordered into a loop.

2. Next, we make sure that the loop is in counterclockwise order. This is easily achieved by checking if the determinant of the edge vectors at any convex vertex is positive. If not, we simply reverse the direction of traversal of the loop.

Thus, we have extracted the visible surfaces in the object in the form of oriented faces. However, since all the vertex processing is done in $2D$ in the image plane, the z-component (depth) of the vertices needs to be extracted. Here, we simply refer back to the surface equations to get the z-component.

Figure 5 shows our results for an "L" shaped nonconvex polyhedron (obtained from the PRIP Lab in Michigan State University). The darker the gray level is, the closer it is to the viewer. The same figure also shows the discretized versions of the symbolic jump edge segments and the roof edge segments. Region segmentation and region shrinking experiments are shown in Figure 6 respectively. The final result of the processing, a simple boundary description, listing oriented faces, is shown in Figure 2.

Extensions and Discussion

We have successfully extracted the boundary representation of the visible surfaces of a convex polyhedral object from its range image. The main emphasis and novelty of our research has been on the geometric analysis aspect and the use of surface-based infromation (that has been directly derived from the range data) in this geometric analysis. Admittedly the test examples are quite simple, since, our main aim is to illustrate our methodology to recover complete boundary descriptions of completely visible surfaces in a single view. Our next steps are (i) to generalize our approach for more general polyhedral objects, e.g., objects with multiple holes and partially visible surfaces in a single view, and (ii) integrate multiple views of the same object from different viewpoints.

References

[1] D.H. Ballard and C.M. Brown, **Computer Vision**, *Englewood Cliffs, N.J.: Prentice-Hall, c1982*

[2] P. Besl, **Surfaces in Range Image understanding**, *Springer-Verlag New York Inc. 1988*

[3] B. Bhanu and T. Henderson, **CAGD-based 3D Vision**, *IEEE Conf. on Robotics and Automation*, 1985. pp. 411-417.

[4] Robert Bolles and Patrice Horaud, **3DPO: A Three-Dimensional Part Orientation System**, *The International Journal of Robotics Research*, Vol 5, No 3, pp. 3-26.

[5] R.O. Duda, D. Nitzan and P. Barrett, **Use of Range and Reflectance Data to Find Planar Surface Regions**, *IEEE Transcation on Pattern Analysis and Machine Intelligence*, PAMI-1, 3, pp. 259-271, July 1979

[6] K.K. Gupta, **Fast collision avoidance for manipulator arms: a sequential search strategy**, *IEEE Transactions on Robotics and Automation, Vol. 6, No. 5, 1990*

[7] K.K. Gupta and X. Zhu, **Extracting Polyhedral Models from a Range Image: A Hybrid Approach**, Technical Report CSS-IS-TR 91-10. Centre for Systems Science. Simon Fraser University. 1991.

[8] O.D. Faugeras and M. Hebert, **Segmenting Range Data into Planar and Quadratic Patches**, *CVPR, Washington*, June, 1983

[9] M. Herman, **Generating Detailed Scene Descriptions from Range Images**, *85 IEEE International Conference on Robotics & Automation, pp 426 - 431*

[10] M. E. Mortenson **Geometric Modeling**, John Wiley & Sons, 1985.

[11] M. Oshima and Y. Shirai, **A Scene Description Method Using Three-Dimensional Information**, *Pattern Recognition 11 (1979)*, pp9 - 17

[12] A. Pentland, **From Pixels to Predicates**, 1987.

[13] T. Lozano-Perez, **A simple motion planning algorithm for general robot manipulators**, *IEEE Journal of Robotics and Automation. Vol.3 No. 3, June 1987*

[14] David Smith and T. Kanade, **Autonomous Scene Description with Range Imagery**, *CVGIP, Vol 31*, pp. 322-334, 1985.

[15] K. Sugihara, **Range-Data Analysis Guided by a Junction Dictionary**, *Artificial Intelligence Vol 12*, pp41-69, May, 1979

[16] N. Yokoya and M.D. Levine, **Range Image Segmentation Based on differential Geometry: A Hybrid Approach**, *IEEE Transcation on Pattern Analysis and Machine Intelligence*, Vol. II, No. 6, pp. 643-649, June 1989

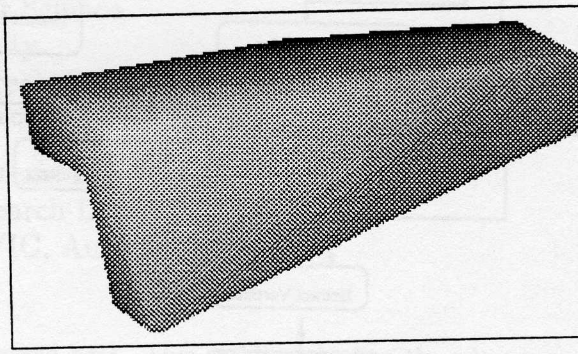
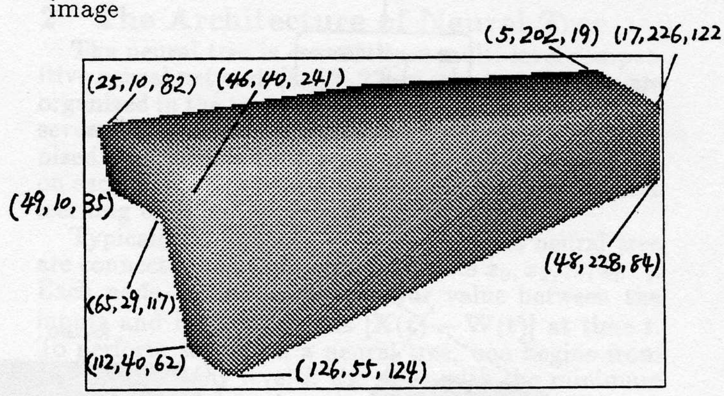


Figure 1: The input to the proposed system: a range image




```
3 surfaces

on surface 0, there are altogether 4 vertices.
in counter-clockwise order they are:
25 10  82; 46 40  241; 17 226 122; 5  202 19

on surface 1, there are altogether 4 vertices.
in counter-clockwise order they are:
48  228 84; 26  229 111; 46  40  241; 126 55  134;

on surface 2, there are altogether 6 vertices.
in counter-clockwise order they are:
65  29  117; 112 40  62; 126 55  124; 46  40  241; 25  10  82;
49  10  35;
```

Figure 2: The output of the proposed system: a boundary representation. The numbers represent the x, y, z co-ordinates of the corresponding vertex.

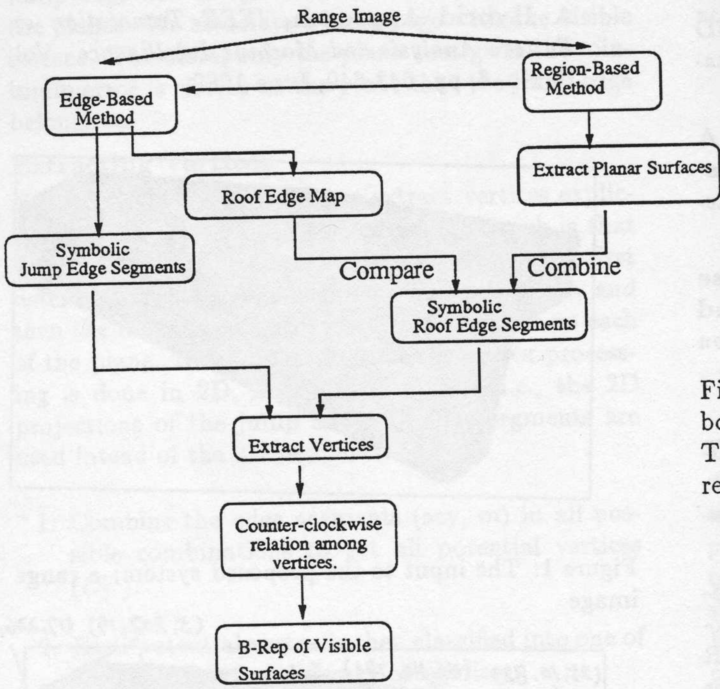


Figure 3: Overview of the proposed algorithm

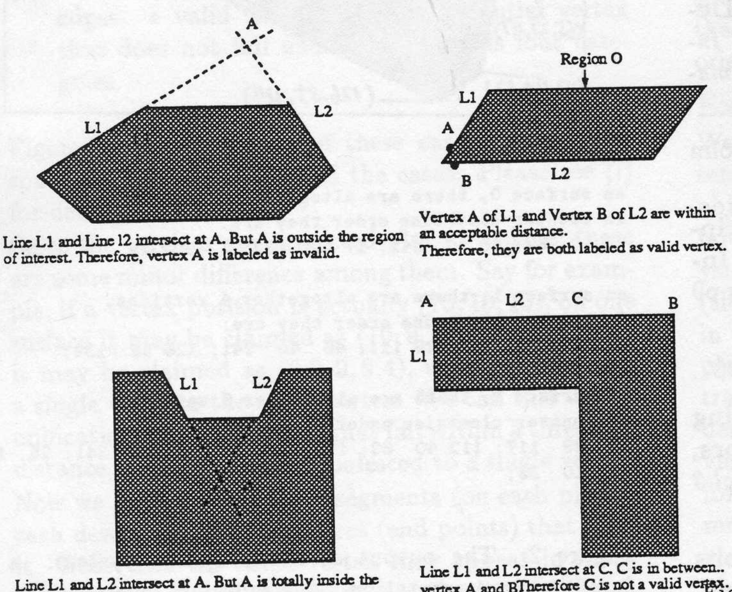


Figure 4: Illustration for different vertex types (a) valid vertex I, (b), (c) and (d) are false vertex types.

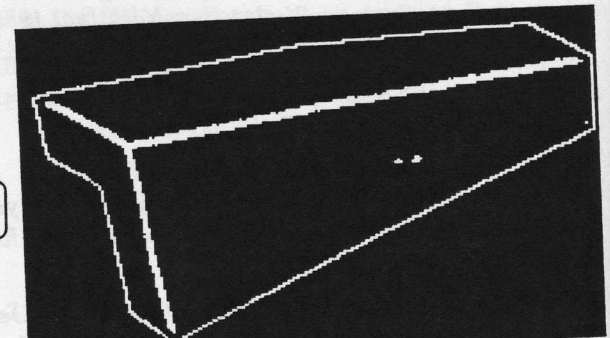


Figure 5: For the image shown in Figure 1, the symbolic Jump and Roof edge segments are shown here. Thin lines represent jump edges and the thick lines represent roof edges.

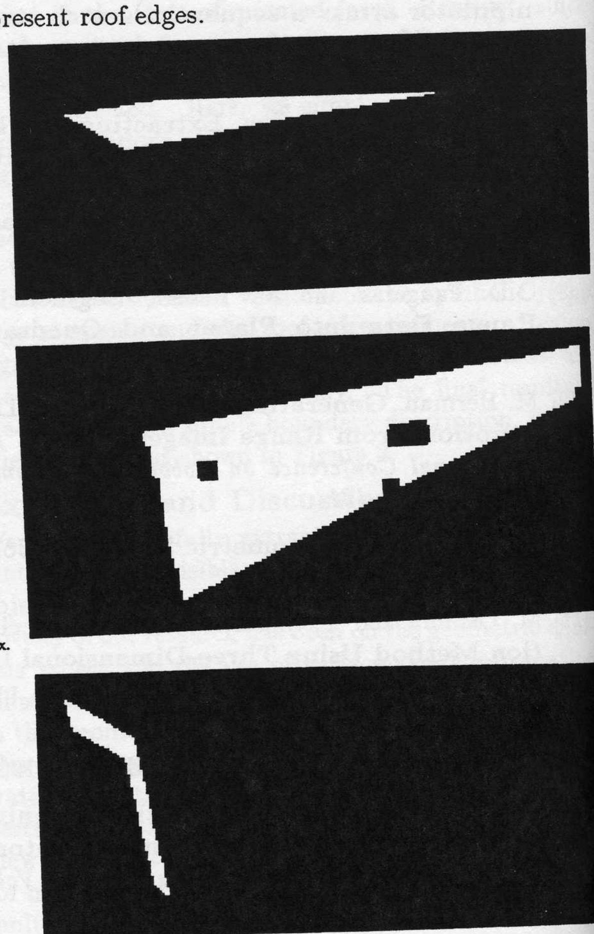


Figure 6: The planes extracted by region segmentation are shown here. Our approach then combines these planes with the edge information shown in Figure 5 to obtain the output shown in Figure 2.