High-Speed Invariant Surface Characterization: 3-D Object Recognition Using Range Data

Sandra Rios and Ernst L. Leiss
Department of Computer Science
University of Houston
Houston, Texas 77204-3475
{rios, coscel}@cs.uh.edu

Abstract

In this paper, we present a new high-speed surface characterization algorithm applied to triangularized surfaces. This characterization algorithm is based on eleven invariant surface characteristics which are calculated from the geometric relationships formed by neighboring triangles. The algorithm is used in a 3-D object recognition system that recognizes, when presented with synthetic range data, ten different, but quite similar, rectangular boxes with between zero to four corners cut off. The recognition system starts with the decomposition of the range data into a network of triangles to obtain a polyhedral representation of the surface. Each triangle of the surface is labeled by one of eleven invariant surface characteristics based on the angles formed by the triangle in question and each of its edge neighbors. Next, the system segments the surface of the object into planar surface patches using the characterization information. The last step of the recognition system extracts geometric features from the surface patches to establish a correspondence with our models. We present and discuss the results and performance of the recognition system based on our new high-speed surface characterization algorithm.

1 Introduction

Three-dimensional (3-D) object recognition is a fundamental part of computer vision systems. Computer vision attempts to understand a given environment from visual and perceptual information. Many practical applications require a computer vision system to be capable of identifying 3-D objects that appear in the environment. Originally, 3-D object recognition research focused on intensity images [2]. A digitized intensity image is a matrix of numbers which represent the brightness at equidistant grid points. Subsequent studies [2] showed that range images are more appropriate for object recognition because they contain depth information. A digitized range image (or depth map) is a matrix of numbers where the numbers represent the perpendicular distance between a sensor focal plane and equidistant points on the surface of an object. Therefore, range data approximate the 3-D shape of the objects in the scene very accurately. Range data can be obtained from range sensors with a high degree of accuracy; techniques for obtaining range data include Light Stripping, Spot Ranging, and Ultrasonic Ranging [1].

Recognition is performed by establishing a correspondence between a stored model and the perceived description of an object. Thus, the two major tasks of an object recognition system are shape description and matching. The shape description process extracts features of 3-D objects that describe the physical properties and their spatial relationship with other objects in the scene. The matching process establishes a correspondence between the objects' physical properties and the descriptions of stored models in order to recognize objects.

This work presents a new high-speed invariant surface characterization algorithm which is an intermediate step towards the goal of obtaining a shape description for the objects in the scene. A *surface characterization* algorithm is an algorithm capable of describing many types of surfaces based on shape [2]. The process of obtaining an object description from range data is considered a very expensive task for a recognition system; many techniques have been presented in the literature that require complex mathematical computations [4, 6, 10, 11]. Motivated by the need for more efficient algorithms, we propose our new surface characterization algorithm which makes the task of obtaining a description more efficient. Our algorithm uses a direct geometric approach to compute efficiently eleven invariant surface types that obtains good performance and results.

Surface triangulation is the first step in our approach to surface characterization. The range data are decomposed into a network of triangles to form a polyhedral representation of the surface. By using the geometrical properties formed by neighboring triangles, we devise a surface characterization algorithm where each triangle of the surface is classified as one of eleven surface types which are view-independent descriptors. The three angles formed between one triangle and each of its three edge-neighbors are used to obtain the eleven surface types.

To develop a comprehensive object recognition system, we use the surface characterization information to obtain the segmentation of the image. The image segmentation algorithm uses the characterization information to group triangles into regions that form meaningful surface patches which resemble the surface of the objects. Related work includes [4, 6, 7, 8, 10]. Furthermore, we implemented a simple matching system to obtain the correspondence between the object description and the object models. Recognition was performed on the synthetic range data of ten different, but quite similar, rectangular boxes with between zero to four corners cut off. The matching system uses specific geometric features of the rectangular boxes, such as angles and distances, to uniquely recognize each of them. We present and discuss the implementation, results, and performance of the object recognition system based on our new high-speed surface characterization algorithm. The results indicate that our algorithm is realistically capable of distinguishing rather similar objects in a highly efficient way.

This paper is structured as follows. Section 2 presents the surface triangulation method used to obtain a polyhedral representation of the surface. The center piece of this section is the presentation of our new surface characterization algorithm. Section 3 discusses the segmentation of the range data into planar surface patches. Section 4 describes the objects used for recognition and discusses the matching process. Section 5 gives a brief overview of the approaches taken by Pieroni and Tripathy and by Besl and Jain to obtain a segmented surface description. Section 6 presents our results; Section 7 concludes the paper.

2 Description of the new method

The shape description mechanism of a recognition system is of critical importance. This description mechanism should be robust, application-independent, data driven, and view-independent. To develop an invariant and appropriate description of the surface we use a surface characterization algorithm applied to triangularized surfaces.

2.1 Surface triangulation

Surface triangulation is the decomposition of the surface of a scene into a network (or mesh) of triangles. In this research, surface triangulation is used to obtain a representation of the surface from range data where an *equilateral triangle* is the basic element. The reason for choosing equilateral triangles over others is that they have the greatest symmetry, thus alleviating orientation bias. They exhibit greater computational efficiency, and a multiresolution [11] technique can be easily implemented. Also, triangles are geometrically very simple, their geometric features can be computed easily, a network of them approximates the surface arbitrarily well, they uniquely determine a passing plane, and conveniently approximate neighborhood of pixels. The triangulation technique presented in this work has two important neighbor relations: Two triangles are considered neighbors if they have at least a point in common; triangles are *edge-neighbors* if they share a complete side and *point-neighbors* if they have one point in common.

The approach used to obtain a surface triangulation of the depth map is to first triangularize the xy-plane and then project these triangles onto the surface of the image. The triangulation (Γ_d) of the xy-plane is accomplished by generating a mesh of equilateral triangles. The triangles in Γ_d have sides of length d and are generated by connecting the following points in Cartesian coordinates (x,y):

$$[(2i\frac{\sqrt{3}}{2}d, jd);\ \ ((2i+1)\frac{\sqrt{3}}{2}d, (j+\frac{1}{2})d) \mid i, j \in Z]$$

These two points represent the two types of nodes ($\bigcirc, \square$) in Γ_d; they are connected to six other points in the following way (see Figure 1.a):

$(2i\frac{\sqrt{3}}{2}d, jd)$ (node $\bigcirc$) is connected to:

$$((2i-1)\frac{\sqrt{3}}{2}d, (j-\frac{1}{2})d); \qquad ((2i-1)\frac{\sqrt{3}}{2}d, (j+\frac{1}{2})d); \qquad (2i\frac{\sqrt{3}}{2}d, (j-1)d)$$

$$(2i\frac{\sqrt{3}}{2}d, (j+1)d); \qquad ((2i+1)\frac{\sqrt{3}}{2}d, (j-\frac{1}{2})d); \qquad ((2i+1)\frac{\sqrt{3}}{2}d, (j+\frac{1}{2})d)$$

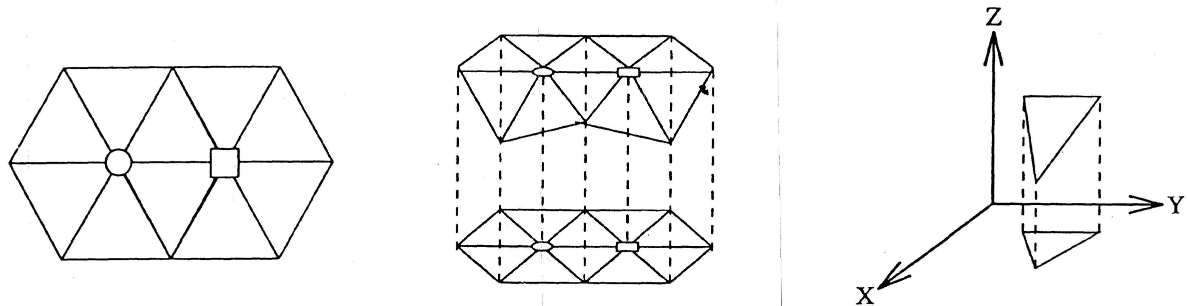
$((2i+1)\frac{\sqrt{3}}{2}d, (j+\frac{1}{2})d)$ (node $\square$)is connected to:

$$(2i\frac{\sqrt{3}}{2}d, jd); \qquad (2i\frac{\sqrt{3}}{2}d, (j+1)d); \qquad ((2i+1)\frac{\sqrt{3}}{2}d, (j-\frac{1}{2})d)$$

$$((2i+1)\frac{\sqrt{3}}{2}d, (j+\frac{3}{2})d); \qquad ((2i+2)\frac{\sqrt{3}}{2}d, jd); \qquad ((2i+2)\frac{\sqrt{3}}{2}d, jd)$$

To decompose a given depth map into a mesh of triangles, we need to project the equilateral triangles of Γ_d onto the surface of the given scene. The projection is parallel and in normal direction to the (x,y,0)-plane.

Each point (x,y) of Γ_d is given a z-coordinate value π(x,y) that lies on the surface of the observed scene: thus, if (x,y) is a point in the plane, (x,y,π(x,y)) will be the three-dimensional coordinate of its projection (see Figure 1.b). Looking at Figure 1.c, we note that the projected triangles need no longer be equilateral and need not lie on the same plane; however, triangles which were edge-neighbors (point-neighbors) in Γ_d will remain edge-neighbors (point-neighbors) in the projection.



(a) Triangulation Γ_d (b) Projection of Γ_d (c) Projection of one triangle

Figure 1: Triangulation of the surface.

For the implementation of our program, each triangle is represented by a node of a graph. Each node contains the three points that form the triangle and is linked to three other nodes that are its edge-neighbors. The computation of the z-coordinate is performed by an interpolation technique since the (x,y) coordinates of the range data are not necessarily the same as those of our triangulation. The use of interpolation effectively adds noise to the data; therefore a good interpolation technique should be chosen. We used the following weighted interpolation that provided good results:

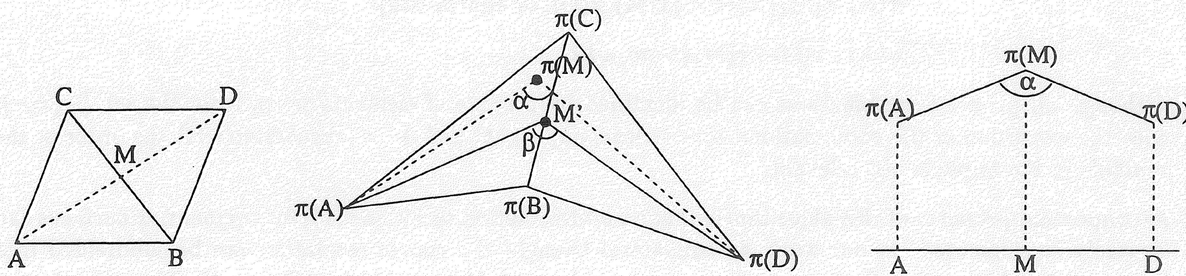
$$\pi(x,y) = \frac{(d_{max}-d_1)R_{i,j} + (d_{max}-d_2)R_{i+1,j} + (d_{max}-d_3)R_{i+1,j+1} + (d_{max}-d_4)R_{i,j+1}}{4d_{max} - d_1 - d_2 - d_3 - d_4}$$

where R is the matrix that contains the range data, d_1, d_2, d_3 and d_4 are the distances from (x,y) to the closest four corner points of the matrix that surround (x,y), and d_{max} is the maximum distance from one corner point to the diagonal point. Figure 6.b illustrates the results of surface triangulation.

2.2 Surface characterization

Consider the triangle T1 (ABC) and its edge-neighbor T2 (BCD) in the triangulation Γ_d of the xy-plane (see Figure 2.a). Also, consider the line (AD) that intersects the edge BC at its midpoint M (see Figure 2.a). If T1 and T2 are projected onto the surface of the scene, the triangles $T1_p$ and $T2_p$ will be generated, where $T1_p$ is defined by the three points (A,π(A)), (B,π(B)), (C,π(C)) and $T2_p$ is defined by the three points (B,π(B)), (C,π(C)), (D,π(D)). It follows that the angle (β) between the two triangles $T1_p$ and $T2_p$ is directly related to the angle (α) between the lines defined by (A,π(A)), (M,π(M)) and (M,π(M)), (D,π(D)). Note that the angles β and α are not necessarily equal because the midpoint of the line defined by (A,π(A)), (D,π(D)) might not be the same as (M,π(M)). However, both angles will be either > 180°, or

= 180°, or < 180° assuming that we have a smooth surface (see Figure 2.b). It is also important to note that since A, D, and M lie on the same line on Γ_d, the six points (A,0), (M,0), (D,0), (A,π(A)), (M,π(M)), (D,π(D)) all lie on the same plane defined by π (see Figure 2.c).



(a) (b) (c)

Figure 2: (a) Two edge-neighboring triangles. (b) Projection of two edge-neighboring triangles. (c) The angle α between two projected triangles.

Now, we describe the characterization of triangles based on the angles formed with their edge-neighbors. From the triangulation of Γ_d we note that each triangle has three edge-neighbors. Therefore, each triangle will form an angle α with each of its edge-neighbors for a total of three angles. Each of these angles will be either less than (<), equal to (=), or greater than (>) 180°, which will make a total of 27 (=3^3) possible combinations. Note that rotations for (unlabeled) equilateral triangles are identity operations; therefore, only those combinations that are not equivalent under rotation will be distinguished. Of the 27 combinations shown in Figure 3.a, numbers 1, 2, 3, 5, 6, 8, 9, 14, 15, 18, and 27 are primary. This gives us a total of eleven primary combinations that are invariant under rotation. The names (labels) assigned to the eleven primary combinations are shown in Figure 3.b. We denote > (=,<) by + (=,–), with the lexicographical ordering +,=,–.

1	+++		15	==-	
2	++=		16	=-+	4
3	++-		17	=-=	15
4	+=+	2	18	=--	
5	+==		19	-++	3, 7
6	+=-		20	-+=	6, 16
7	+-+	3	21	-+-	9
8	+-=		22	-=+	8, 12
9	+--		23	-==	15, 17
10	=++	2, 4	24	-=-	18
11	=+=	5	25	--+	9, 21
12	=+-	8	26	--=	18, 24
13	==+	5,11	27	---	
14	===				

(a) Combinations

1	+++	pit
2	++=	small landing up
3	++-	up fork
4	+==	big landing up
5	+=-	cornerstep right
6	+-=	cornerstep left
7	+--	down fork
8	===	flat
9	==-	big landing down
10	=--	small landing down
11	---	peak

(b) Names of eleven primary combinations

Figure 3: Surface types.

To characterize each triangle (of the projected triangulation Γ_d) as one of the eleven types using an efficient algorithm, the following equivalence is used:

$\acute{\alpha}\{>,=,<\}180°$ is equivalent to:

$$\pi(A)\text{-}\pi(M)\{>,=,<\}\pi(M)\text{-}\pi(D), \text{ or equivalently}$$

$$\pi(A)+\pi(D)\text{-}2\pi(M)\{>,=,<\}0.$$

Although all primary combinations can be identified by a triple of elements taken from the set {+,=,–}, with the exception of the combinations +=– (cornerstep right) and +–= (cornerstep left) the order of the symbols in the triple is not relevant.

An important property of this algorithm is that a multiresolution technique [11] for segmenting surfaces can be easily implemented. In our work, an equilateral triangle of a coarse resolution can be subdivided into four equilateral triangles of a finer resolution whose sides are of length d/2 (see Figure 4). The midpoints of the triangles sides of a coarse resolution can be used for the computations of a finer resolution making the task of obtaining multiple resolutions very efficient. The four smaller triangles of a finer resolution are the children of the triangle in the coarser resolution. To characterize the surface, each triangle of the coarsest resolution can be classified (uniquely, in many cases) as one of the eleven surface types by analyzing its own type and the types of its children at multiple resolutions.

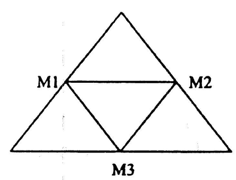


Figure 4: Multiresolution.

For our implementation, we applied the operator $\pi(A)+\pi(D)\text{-}2\pi(M)\{>,=,<\}0$ to each node (triangle) of the graph and each of its three neighbors. Each node was labeled according to the classification technique described above. Note that after characterization, neighboring triangles are labeled with the same type if they belong to the same type of surface and the types change when they reach an edge. Figure 6.b illustrates the results of surface characterization.

3 Segmentation into surface patches

The surface characterization described above provides a segmentation of the surface whose coarseness is determined by d. This section briefly outlines the algorithms used to further segment the image into surface patches that closely resemble the surface of the objects and that can provide a higher-level description of the objects. Since our objects are rectangular boxes formed by planar surfaces, we are only concerned with planar patches. However, the program can be extended to include higher degree surfaces.

3.1 Region growing and edge detection

Recall that the surface is decomposed into triangles which are labeled by one of our eleven types. The presented region growing algorithm will group these triangles according to their labels. Triangles are grouped in the following manner: we group neighboring triangles of the same type into a region (R_i) until no more triangles can be added to R_i. Our program uses a breadth-first search graph traversal algorithm to grow each region. After region growing is performed, each node of the graph will belong to a region. Edge triangles are those that form the border of a region. They are detected by testing if a triangle has at least one of its edges neighboring a triangle of a different type. A list of regions and its edges is formed after the execution of the region growing routine.

3.2 Region merging

The surface description of an object can be improved by merging similar regions to create a final description which is more accurate and meaningful. Neighboring regions are merged into larger regions if they have a similar plane equation. The program first finds a kernel region to be expanded. A region is a candidate to be the kernel region if it has the largest number of nodes and the region has not been expanded. Next, we fit a plane equation to the kernel and its neighboring regions by using a least-square-fit method. A neighboring region is merged to the kernel if it satisfies the following conditions [13]:

$$p_k^2 = (a_k - a_r)^2 + (b_k - b_r)^2 + (c_k - c_r)^2 < p_t^2 \qquad \mid d_k - d_r \mid < d_t$$

where the subscript k denotes the kernel, the subscript r denotes the neighbor region, t denotes a threshold, and a, b, c, and d correspond to the plane equation:

$$f(x, y, z) = ax + by + cz - d = 0$$

The algorithm extends the kernel region until there are no more neighbors that satisfy the merging conditions. Merging stops when all regions in the graph have been considered as a kernel.

4 Matching

Ten different boxes with between zero to four corners cut off are the models used in our recognition system. Our boxes are 10 units long, 8 units wide, and 6 units high. Specifically, a box with all four corners is defined by the following points: p1(0,0,0), p2(10,0,0), p3(0,8,0), p4(10,8,0), p5(0,0,6), p6(10,0,6), p7(0,8,6), p8(10,8,6). The cut off corners are obtained as follows: C1 is defined by the points (0,0,3), (3,0,6), (0,3,6), C2 is defined by the points (10,0,3), (7,0,6), (10,3,6), C3 is defined by the points (0,8,3), (3,8,6), (0,5,6), and C4 is defined by the points (10,8,3), (7,8,6), (10,5,6) (see Figure 5). If we cut off 0, 1, 2, 3, or 4 corners we obtain 1, 2, 4, 2, 1 boxes dissimilar under rotation, respectively, for a total of ten. Our primary reason for using these objects to measure the performance of the surface characterization algorithm is that they are composed of several planes which have similar geometric features such as size, angles, and distances. This similarity requires the characterization algorithm to be very accurate in distinguishing the differences.

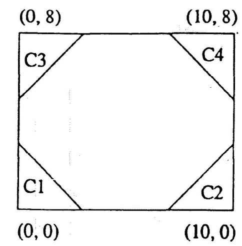


Figure 5: Top view of boxes.

Also, the boxes have many edges which requires the algorithm to correctly distinguish flat surfaces from edges.

The models used in our recognition system are represented by one view and their specific features [6]. Such a representation was chosen because our models can be distinguished by geometric features that are invariant to rotation and translation. Only one view is necessary since the top view of the objects contains enough information for recognition purposes. Each unknown object is recognized as one of the ten box models by comparing angles and distance ratios between significant points. Specifically, the matching system first finds the possible number of cut off corners. This is done by counting the number of significant surface patches that form an angle of approximately 125° with the largest surface patch, where 125° is the angle formed by the top face and a cut off corner. A significant surface patch is one that has fewer than 50% of its triangles as edge triangles. The largest surface patch will always be the top face of the box assuming that this face is visible. The possible number of cut off corners reduces the number of possibilities for comparison with the models. Next, the matching system finds the corner points of the top face by using the edge triangles. Then, the interior angles between consecutive sides of this patch are determined; they are either 90° or 135° approximately. The sequence of angles uniquely distinguishes most of the boxes. We chose the sequence of the angles rather than that of the distances because they are less sensitive to errors. If this information is not sufficient, the matching system considers the sequence of distance ratios from one corner point to the next to uniquely distinguish a box.

5 Related work

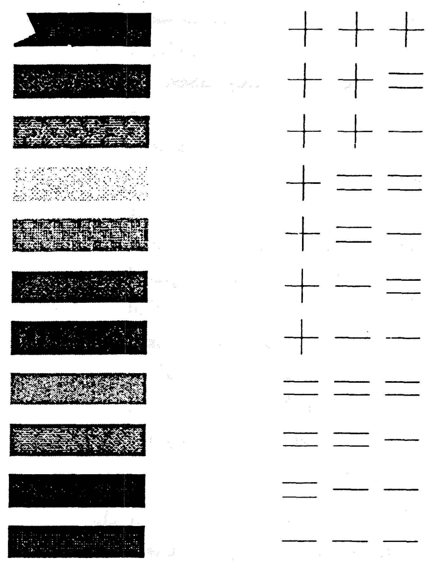
This section briefly describes the approach taken by Besl and Jain [2] and by Pieroni and Tripathy [11] that formed the starting point of our research. Besl and Jain used differential geometry to obtain a description for smooth surfaces. The *mean curvature* (H) and the *Gaussian curvature* (K), which possess desirable invariance properties, are used to retrieve surface characteristics. Their algorithm used the signs of H and K to classify each pixel's neighborhood as one of eight surface types. The surface types include: peak, pit, ridge, valley, flat, minimal, saddle valley, and saddle ridge. Besl and Jain compute H and K for each pixel of the surface by using the maximum principal curvature and the minimal principal curvature. The process of classifying pixels is referred to as surface characterization and provides an initial coarse segmentation of the surface. This information is used to further refine the segmentation of the surface based on a variable-order bivariate surface fitting algorithm [5].

Pieroni and Tripathy used a multiresolution approach for the segmentation of surfaces. They applied the properties of the mean (H) and the Gaussian (K) curvatures to triangularized surfaces, in which each triangle was characterized by one of the eight surface types listed above. They used triangle neighbors, instead of a neighborhood of pixels, for the computation of H and K. The reasons for using triangles as basic elements are simplicity and ease of calculation of derivatives and curvatures. They triangularize the xy-plane into *isosceles triangles* that are then projected onto the surface of the object to yield a polyhedral representation of the surface. To characterize each triangle as one of the eight surface types, they introduce the concept of pseudo-curvature, which is the folding of the triangles with respect to their neighbors. Their technique is a multiresolution technique because they use triangles of different sizes. Specifically, a triangles of level i (finer resolution) generates a child triangle at level i+1 (coarser resolution) by doubling the lengths of its sides and rotating by $\pi/2$. Pieroni and Tripathy used a total of four levels to calculate the surface characteristics of a triangle at the first level. Consequently, the characterization of a triangle at level 1 is obtained by analyzing its own classification, and the ones of its children at levels 2, 3, 4, and their neighbors. This model works well with relatively noisy surfaces, but may produce fuzzy edge characteristics and introduce orientation bias due to the chosen triangulation method.

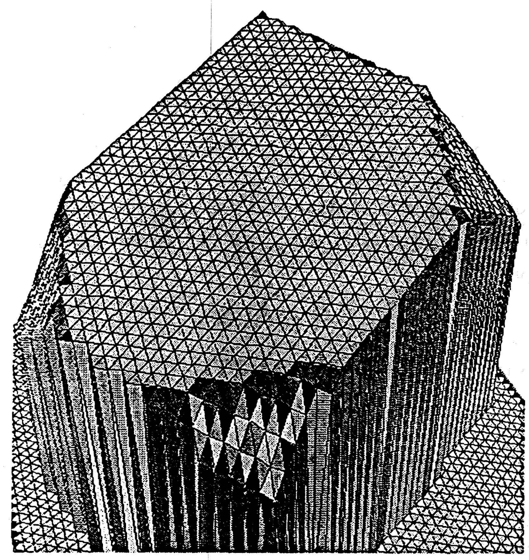
The algorithms used by Besl and Jain and by Pieroni and Tripathy both used H and K for the computation of the surface characteristics which require the computation of differentials up to the second order. In our research, we take a direct and simple geometric approach by using a measure of the type of angle $(>,=,<)$ between a triangle and each of its edge-neighbors. Also, the fact that our operator $\pi(\mathrm{A})+\pi(\mathrm{D})-2\pi(\mathrm{M})\{>,=,<\}0$ is very simple to compute makes our algorithm more efficient. Another improvement over the method used by Pieroni and Tripathy is that our triangulation technique uses equilateral triangles which solves the problem of orientation bias. The implementation of a multiresolution technique using our algorithm is simpler and more efficient since we use equilateral triangles in the triangulation Γ_d.

6 Experimental results

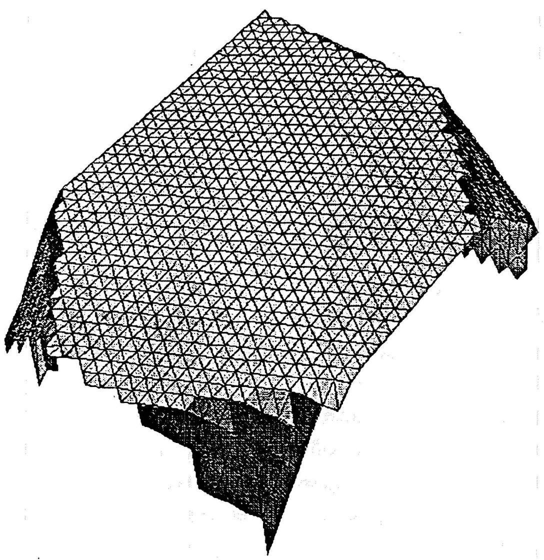
To illustrate our recognition system, we show how to recognize a box with three cut off corners. For details on parameters, etc. see [12]. The box is rotated to an arbitrary position and the range data are synthetically generated. The first step is to triangulate the depth-map for the box, where Γ_d has d set to 0.3. Once triangulation is performed, each triangle of the surface is characterized by one of the eleven surface types listed in Section 2.2. The results of triangulation and characterization are shown in Figure 6.b. The triangles of Figure 6.b are painted with eleven different colors (see Figure 6.a), where each color represents one of the eleven surface types. Note that the edges of the box are characterized by different types from the planar surfaces. Since the interpolation introduced some error, the cut off corners are not completely characterized as a planar surface. Figure 6.c shows the result after region growing and region merging is performed. Note that the triangles of the cut off corners are now grouped into three regions. Figure 6.d shows the edges of the regions that form the box. Finally, geometric features are computed and matching is performed.



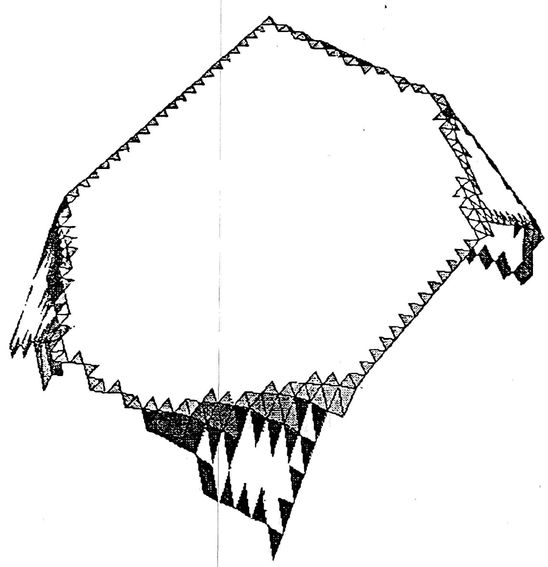
(a) Colors for the eleven surface types.



(b) Triangulation and characterization.



(c) Region merging.



(d) Region edges.

Figure 6: Box with three corners cut off.

The recognition system was tested comprehensively by generating all ten boxes rotated about the x and y-axis from $-30°$ to $30°$ in steps of $3°$, and about the z-axis from $0°$ to $180°$ in steps of $10°$. This resulted in a total of 83790 runs, where the program failed to recognize the box correctly in fewer than 0.5% (357 runs) of the cases. In almost 50% (166 runs) of these incorrect matches, the program provided as final result two choices, one of which was correct. Figure 7 summarizes these results. Each run of the program took less than 3 sec from generating the range data to recognizing the object, when executed in the Unix environment on a Sun SPARCstation5. Thus, the program is capable of real-time recognition.

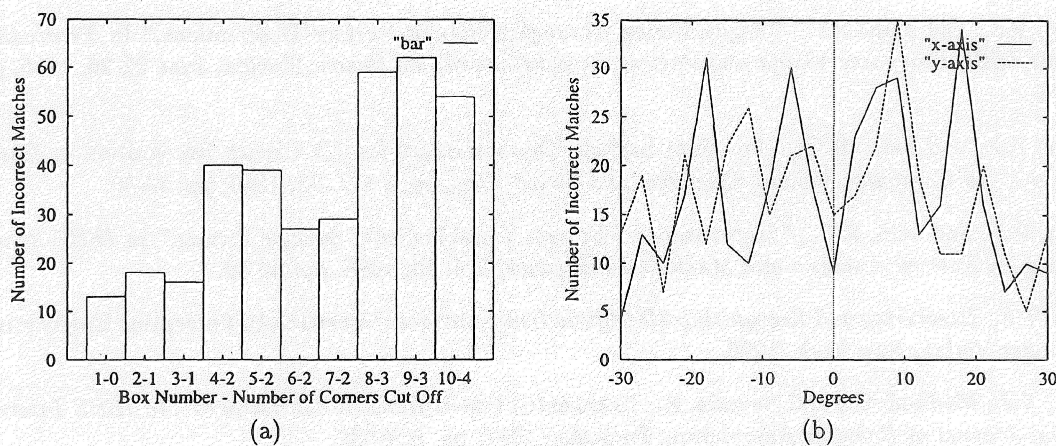


Figure 7: (a) Incorrect matches for each box (b) Incorrect matches for specific x and y-axis rotations (for all z-axis rotations).

7 Conclusions

This work proposed a new algorithm for the characterization of 3-D surfaces using range data. The aim of our research was to characterize a 3-D surface from range data by using a simple and efficient algorithm that provides good results. We used a direct geometric approach that exploited the geometric properties formed by neighboring triangles. A measure of the angles formed between a triangle and each of its three edge-neighbors was used for the classification of the surface types.

Our new surface characterization algorithm applied to triangularized surfaces provides promising results because it effectively distinguished edges from planar surfaces. Furthermore, results show that our algorithm provides excellent computation time and can be easily implemented and used in a recognition system. The performance of our recognition system was good since it only failed in less than 0.5% of the test cases. Many of the incorrect matches (see Figure 7) presented in the results were due to the merging and matching algorithms. More powerful region growing, region merging, and matching techniques may be needed for recognizing objects that are geometrically more complicated and have higher degree surfaces. Additional research can be done to test the algorithm presented in this paper by using multiple objects composed of curved surfaces, introduce other types of noise to the range data, and implement a multiresolution technique [11] as described in Section 2.2.

References

[1] Ballard, D.H., and Brown C.M., *Computer Vision*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[2] Besl, P.J., *Surfaces in Range Image Understanding*, In Perception Engineering, Springer-Verlag, New York, 1988.

[3] Besl, P.J., and Jain, R.C., "Segmentation Through Symbolic Surface Descriptions," In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, Miami Beach, Florida, June 22-26, 1986, pp. 77-85.

[4] Besl, P.J., and Jain, R.C., "Invariant Surface Characteristics for 3D Object Recognition in Range Images," In *Computer Vision, Graphics, and Image Processing*, Vol. 33, 1986, pp. 33-80.

[5] Besl, P.J., and Jain, R.C., "Segmentation Through Variable-Order Surface Fitting," In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 33, 1986, pp. 33-80.

[6] Fan, T.J., *Describing and Recognizing 3D Objects Using Surface Properties*, In Perception Engineering, Springer-Verlag, New York, 1990.

[7] Fan, T.J., Medioni, G., and Nevatia, R., "Segmented Descriptions of 3D Surfaces", In *IEEE International Journal of Robotics Automation*, December 1987, pp. 527-538.

[8] Faugeras, O.D., Hebert, M., and Pauchon, E., "Segmentation of Range Data Into Planar and Quadratic Patches", In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Washington, D.C., June 19-23, 1983, pp. 8-13.

[9] Pieroni, G.G. and Leiss, E.L., "High-Speed Reconstruction of 3D Images from Range Data,", Keck W.M. Research Computation Laboratory, Vol. 8, April 1992, pp. 253-267.

[10] Lin, C., and Perry, M.J., "Shape Description Using Surface Triangulation," In *IEEE Proceedings Workshop on Computer Vision: Representation and Control*, August 1982, pp. 38-43.

[11] Pieroni, G.G., and Tripathy, S.P., "A Multiresolution Approach for Segmenting Surfaces," In *Issues of Machine Vision*, Springer-Verlag, New York, 1989, pp. 221-247.

[12] Rios, S., "High-Speed 3D Object Recognition Using Range Data," Master Thesis, Department of Computer Science, University of Houston, Texas, USA, 1997.

[13] Shirai, Y., *Three-Dimensional Computer Vision*, In Symbolic Computation, Springer-Verlag, Berlin Heidelberg, 1987.