Segmentation of Range Images Using Local Approximation of Scan Lines

Inas Khalifa*, Medhat Moussa†, and Mohamed Kamel*
* Systems Design Engineering Dept., University of Waterloo, Waterloo, ON N2L 3G1
{inas,mkamel}@pami.uwaterloo.ca
† Dept. of Physics and Computing, Wilfrid Laurier University, Waterloo, ON N2L 3C5
mmoussa@wlu.ca

Abstract

Range image segmentation is an essential building block in many computer vision systems. This paper presents a method for automatic segmentation of range images using scan lines. Local approximation of each scan line is used to reliably detect position discontinuities as well as noise pixels. A model for the detection of orientation discontinuities that is capable of capturing weak discontinuities without producing false edges at high curvature regions is also presented. The different image regions are then identified using an adaptive grouping process and approximated by appropriate surfaces. Experimental results on real range images demonstrate the robustness and efficiency of the method.

Keywords– Range images, segmentation, edge detection, edge models, scan lines.

1 Introduction

Range images have been widely used in computer vision systems in recent years. Their use has been prompted by the availability of fast, accurate, reliable, and economical range sensors. A range image is an array of pixels in which each pixel value encodes the depth or the distance of a point on a visible scene surface to the sensor. The pixel values directly approximate the three dimensional shape of the objects in a scene. Thus, range images are more suited to 3D scene analysis tasks than intensity images. Image segmentation is considered to be the highest domain-independent processing stage in a computer vision system. It constitutes an essential building block in many vision applications such as reverse engineering, object recognition, and automated assembly and inspection.

Range image segmentation can be defined as the process of partitioning an image into subsets of connected image regions, such that each subset corresponds to a meaningful surface portion in the scene and is represented by an appropriate geometric primitives. There are three types of edges in range images. Jump and fold edges signify position and orientation discontinuities, respectively. Smooth edges signify discontinuities in surface curvature. Segmentation techniques can be broadly classified as either *edge-based* or *region-based* depending on whether they emphasize the detection of surface discontinuities or the detection of smooth surface regions, respectively [1].

Edge-based segmentation techniques usually suffer from the inevitable fragmentation of edges and the need for efficient postprocessing, e.g. edge linking and gap filling, in order to obtain a complete segmentation. However, they possess *relatively* simple control structures compared to region-based techniques, and they tend to locate regions boundaries more precisely. Examples of such techniques have been reported in [2–5]. On the other hand, region-based techniques always produce closed regions. However, they typically suffer from the possibility of over or under-segmentation, distortion of region boundaries, the need for a good criterion for merging and splitting adjacent regions, and/or the sensitivity to the choice of the initial seed regions, e.g., [6], and [7]. Also, they possess complex control structures, are typically iterative, and are generally much slower than edge-based techniques.

Since the problems of detecting surface discontinuities and homogeneous surface regions are in fact complementary, combining both techniques can potentially produce better results. Lejeune and Ferrie [8] have proposed to use a feature map as input along with a surface segmentation calculated using curvature scale space. The final set of surface regions that are most consistent with input features is then determined using a relaxation labelling algorithm. Fitzgibbon *et al* [9] developed a region-based technique that is preceded by edge detection, in the form of thresholding of the difference in depth values, in order to estimate surface curvature reliably. While this method has demonstrated very good segmentation results according to a recent comparison [10], it overlooks the fact that thresholding is not *suffi-*

cient (consider, for example, pixels on a surface with a high slope). A more robust curvature estimation method using tensor voting has been proposed in [11].

Segmentation techniques that utilize scan lines exhibit high-speed performance and allow parallel processing. Moreover, recent implementations have presented promising results. In [12] and [13], region-based techniques that use scan lines are proposed, however, they are limited to polyhedral objects. Jiang and Bunke [5] presented an edge detection method based on scan lines. They approximate each scan line by a quadratic function using its midpoint and two end points. The pixel that exhibits maximum fit error is labelled an edge candidate and the scan line is split at that location. This process is repeated until the fit error at all pixels is less than a certain threshold. Edge candidates are then examined to remove false edges. Unfortunately, this technique has two main disadvantages: (a) poor edge localization, since the splitting algorithm is not precise for noisy images and it provides no intuitive geometric relationship between the true edge location and the location where the lines are split, and (b) over-segmentation due to using a quadratic function, since curves or surfaces of order 4 at least are required to accurately represent natural quadrics such as spheres and cylinders [14].

This paper presents an efficient method for automatic segmentation of range images. A novel edge-based segmentation method based on local approximation of scan lines that is capable of overcoming both shortcomings of [5] is introduced. The method calculates the best local approximating function to the scan lines at each image pixel. These functions are first used to identify noise pixels. They are also employed, in conjunction with the jump edge model used by Jiang and Bunke [5], to identify jump edges. Then, a fold edge model capable of capturing weak folds without producing false edges at high curvature regions is introduced. In order to close the contour gaps that may be present in the resulting set of edge, an adaptive pixel grouping process [15] is used. Finally, a complete segmentation is achieved by correcting instances of over and under-segmentation.

The rest of this paper is organized as follows. Section 2 introduces the different edge models and describes the edge-based segmentation. Section 3 describes the adaptive pixel grouping process and how a complete segmentation is achieved. Experimental results are presented in section 4. Summary and conclusion are given in section 5.

2 Edge Detection Using Local Approximation of Scan Lines

Consider a dense range image regularly sampled in both the x and y directions, i.e., orthographic. Each row or column in the image represents a vertical section of the scanned object surface. Thus the pixels in each scan plane constitute a digital piece-wise smooth curve, i.e., a *scan line*. Local approximation of scan lines is accomplished as follows. At each image pixel, $\mathbf{x}_i$, in a scan line, the best approximating quadric in a neighborhood of N pixels, N being odd, is calculated by considering the N possible sets of pixels containing $\mathbf{x}_i$, i.e.,

$$\{\mathbf{x}_j : j = i + k - (N-1), \ldots, i + k\} \qquad (1)$$

where $k = 0, 1, \ldots, (N-1)$ and i is the index of $\mathbf{x}$ in the scan line. There is a tradeoff in the choice of the value of N. It should be set small enough to insure fast processing, and allow correct representation of small regions but not too small that noise pixels cannot be identified. In the current implementation, the value of N is set through training as will be shown in section 4.

This technique overcomes the disadvantages of [5] mentioned above, as follows. It allows better approximation at and near jump and fold edges, and thus, edges can be localized precisely. Also, the use of quadrics here will not result in an over-segmentation, since only small local regions are considered. If higher order or piece-wise functions, e.g. splines, were used, they would follow the data points very closely. This would be reasonable if the measurements were reliable, or noise-free, which is not the case in most of the available range sensors.

2.1 Detection of Noise Pixels

Range images are typically corrupted by different types of noise. Spike noise results in isolated image pixels with range values remarkably different than their surrounding. There are also some instances where a sensor is unable to make a range measurement (usually the range value is set to zero). Moreover, the range measurements are sometimes valid, but there is insufficient information to discern separation of surfaces [10], such cases may occur due to surfaces almost parallel to the scanning direction giving rise to one or two pixel wide regions. Finally, there is quantization and measurement noise which add a small tractable error randomly at each image pixel. This type of noise can be estimated by finding the best-fitting plane to a scanned surface known to be planar. The standard deviation, σ of the error of fit indicates the severity of the noise. In the following, the first three types will be referred to collectively as *noise pixels*.

Noise pixels are identified using the technique of local approximation of scan lines, equation (1), as follows. Typically, one or more of the N sets will produce a maximum fitting error comparable to σ. However, for noise pixels, the fitting error of *all* sets will exceed σ and hence they can be identified.

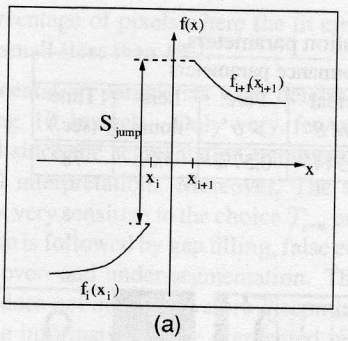

Figure 1: Illustration of the jump edge detection model.

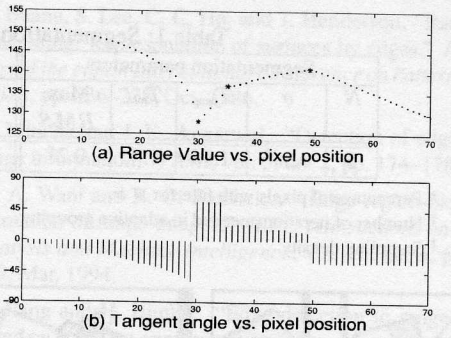

Figure 2: Illustration of the fold edge detection model.

2.2 Detection of Jump and Fold Edges

Now that a good local approximation of the surface at each pixel has been calculated, the precise definition of jump edge strength given by Jiang and Bunke [5], shown in figure 1, can be used as follows. Let $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ be two adjacent pixels, belonging to two curve segments $f_i(\mathbf{x}_i)$ and $f_{i+1}(\mathbf{x}_{i+1})$, respectively. The discontinuity strength of a jump edge is defined as:

$$S_{\text{jump}} = |f_i(\overline{\mathbf{x}}) - f_{i+1}(\overline{\mathbf{x}})|$$

where $\overline{\mathbf{x}} = \frac{\mathbf{x}_i + \mathbf{x}_{i+1}}{2}$. If S_{jump} is larger than a certain threshold, both $\mathbf{x}_1$ and $\mathbf{x}_2$ are labelled as jump edge pixels, i.e., each of them lies on the boundary of its corresponding region. This threshold can be fixed for a certain sensor using the maximum noise level and the sampling density as will be explained in section 4. The edge model defined above clearly alleviates the problems encountered with simple thresholding.

Fold edges are usually detected by thresholding the difference in surface normals or the maximum curvature. This definition is not always adequate. For example, in order to capture a fold edge at the junction of two planes with a small difference in surface normals, the threshold will be set too low that pixels on a highly curved surface will be labeled incorrectly as edges. Alternatively, if only pixels with very high curvature are considered to be fold edges, the pixels at the junction of the two planes will be missed. To overcome this problem, an appropriate model for fold edge detection is introduced. Consider the scan line shown in figure 2a. The surface normal components are calculated using local approximation as explained above. For simplicity, the angle between the curve tangent and the x axis at each pixel is depicted instead, figure 2b. Note that the angle changes relatively *smoothly* on each surface, however, at the point where two surfaces meet (shown as *), the angle change is large, i.e., a jump discontinuity occurs. Thus, in order to detect fold edges, the best local function that approximates the surface normal components is calculated at each pixel in a neighborhood of N pixels as before, say $g_i(\mathbf{x}_i)$. The strength of a fold edge is calculated as:

$$S_{\text{fold}} = |g_i(\overline{\mathbf{x}}) - g_{i+1}(\overline{\mathbf{x}})|$$

The fold strengths are then thresholded to detect fold edges. It is clear that using this definition, the threshold can be set low enough to capture weak fold edges without producing false edges on curved surfaces. In order to use the above model, the normals at, and in the vicinity, of fold edges should be calculated with small error which is guaranteed by the local approximation method.

3 Complete Segmentation and Surface Description

The output of the edge-based segmentation process is a binary edge map. Typically, as with most edge detection schemes, there will be gaps in region boundaries. In order to fill those gaps, the adaptive pixel grouping technique proposed in [15] is used. First, connected image regions are identified. Then, a region test is performed for each region. The region test is a surface fitting operation, where the test is successful if the RMS fit error is lower than a certain threshold, and the region size is larger than T_{size}. If a region passes the region test, it is registered and added to the final region map. Otherwise, the edges in that region are dilated once in hope of closing an existing gap. This process is repeated until all regions pass the test. After that, edge pixels, from edge detection and dilations, are assigned to existing regions if the fit error is less than σ. Adjacent regions are examined to detect instances of over-segmentation, again using the region test.

Instances of under-segmentation may also occur. An edge contour may not be detected due to low edge strength, or the presence of a smooth edge. Also, if the scanned object

Table 1: Segmentation and performance evaluation parameters.

Segmentation parameters					Average performance parameters					
N	σ	T_{jump}	T_{fold}	Max. RMS	Image size	# of regions	Actual RMS	Perc. $> \sigma^*$	Iterations†	Time (sec.)
5	$0.5\delta^\ddagger$	1.5δ	15°	0.2δ	256×256	29.25	0.1544δ	0.57%	3.01	78

* Percentage of pixels with fit error $> \sigma$.
† Number of iterations needed in adaptive grouping.
‡ Sampling density.

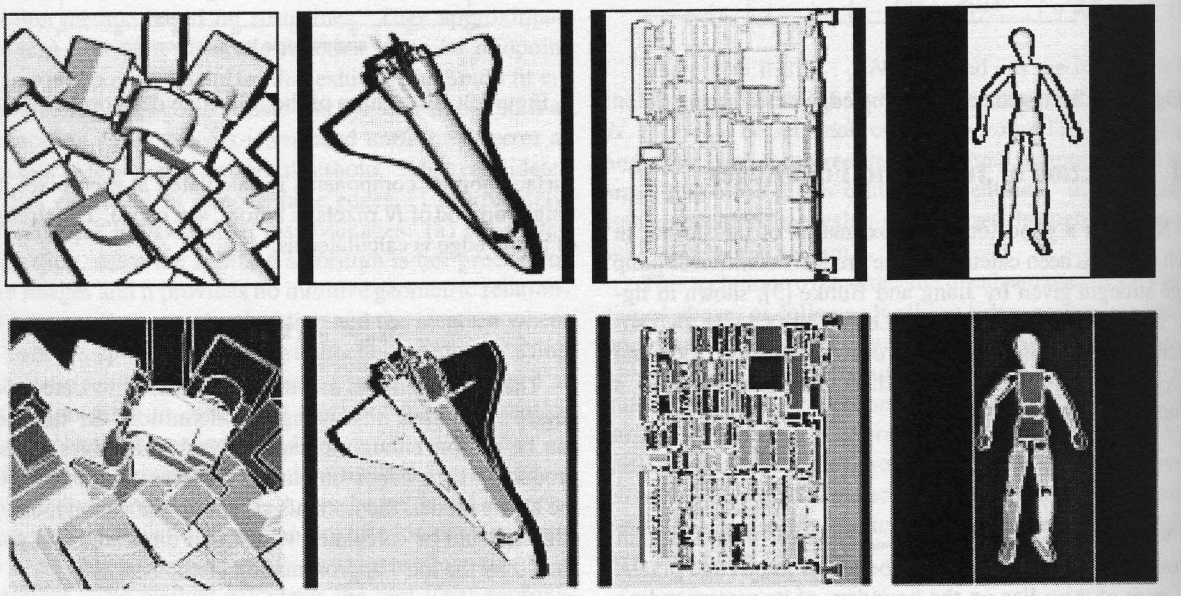

Figure 3: Segmentation results of selected NRC images

exhibits complex free-form surface structure (e.g., a human face), closed surface region may fail the region test due to the limited surface types allowed. In this implementation, this problem is treated by adding new regions to accommodate such cases as follows. A region growing process is started at an unlabelled pixel and continued as long as the new region passes the region test. The new region is then registered. The above steps are repeated until all unlabelled pixels are accounted for. This procedure, though it may not always be able to recover the missed edges, it produces a segmentation where image pixels are approximated by surfaces with a tolerable deviation.

In the current implementation, surface types allowed by the region test include planes, spheres and biquartic patches. Algorithms for fitting conventional quadrics such as cylinders, cones, and tori [16] have also been implemented. However, since these techniques involve solving nonlinear least-squares problems and thus increase the processing times remarkably, the choice of including those surface types in the region test can be made according to the application. Biquartic patches are used because they are capable of accurate representation of such natural qadrics [14]. Tensor product B-splines are considered for future addition.

4 Experimental Results

The segmentation method presented in this paper has been tested on 100 real range images obtained from the web site of the VIT group at the Canadian National Research Council (NRC). Figure 3 shows examples of the segmentation results obtained using this method. The first row shows the original images shaded such that the grey level at each pixel is inversely proportional to the angle between the surface normal and the positive z direction. The second row shows the final segmentation. Table 1 summarizes the segmentation parameters used by the method as well as the average performance evaluation parameters. The results demonstrate the efficiency of the method in terms of:

- Fast processing.
- Accurate surface representation, since the actual RMS fit error is much smaller than σ.

- The percentage of pixels where the fit error exceeds σ is very small (less than 1%).

The segmentation parameters were determined through training using 10 images. Only very few training runs were needed since the segmentation parameters have a direct physical interpretation. Moreover, The segmentation process is not very sensitive to the choice T_{jump} and T_{fold}, since edge detection is followed by gap filling, false edge removal, and treating over- and under-segmentation. The segmentation method does not detect curvature discontinuties. However, the edge information in the segmented images can be used to find better estimates of surface curvature and can thus facilitate the process smooth edge detection.

The surface representations allowed in the region test can, in principle, include a very wide variety of surface families. Intuitively, such surface types should be chosen such that they can represent a wide variety of objects. The limiting factor is that these surfaces should be approximated within a reasonable cost, i.e., computational complexity, which varies from one application to another. For example, including NURBS patches may allow the representation of very complex sculptured surfaces. However, fitting NURBS patches to regions with general-shaped boundaries is computationally expensive. The surface types used in the region test to generate the above results are planes, spheres, and biquartic patches. T_{size} was set to 25.

5 Summary and Conclusion

This paper has presented a method for automatic segmentation of range images. The method is based on local approximation of scan lines. Adequate models have been introduced to identify the different types of edges. Adaptive pixel grouping has been used to fill boundary gaps and find a geometric representation for the different surfaces. The method was tested on a large number of images. The reported results demonstrated the efficiency of the method in terms of fast processing and accurate surface representation. They also show the robustness of the method since it is capable of identifying noise pixels as well as treating over- and under-segmentation.

Acknowledgements

This work has been partially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] M. Suk and S. M. Bhandarkar, *Three-Dimensional Object Recognition from Range Images*, Springer-Verlag, 1992.

[2] B. Bhanu, S. Lee, C. C. Ho, and T. Henderson, "Range data processing: Representation of surfaces by edges," *Proceedings of the eighth International Conference on Pattern Recognition*, pp. 236–238, Oct. 1986.

[3] A. Mitiche and J. K. Aggarwal, "Detection of edges using range information," *PAMI*, vol. 5, no. 2, pp. 174–178, 1983.

[4] M. A. Wani and B. G. Batchelor, "Edge-region-based segmentation of range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 314–319, Mar. 1994.

[5] X. Jiang and H. Bunke, "Edge detection in range images based on scan line approximation," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 183–199, Feb. 1999.

[6] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 608–620, 1987.

[7] C. Dorai and A. K. Jain, "COSMOS–A representation scheme for 3d free-form objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1115–1130, Oct. 1997.

[8] A. Lejeune and F. P. Ferrie, "Finding the parts of objects in range images," *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 230–247, Sept. 1996.

[9] A. W. Fitzgibbon, D. W. Eggert, and R. B. Fisher, "High-level CAD model acquisition from range images," *Computer-Aided Design*, vol. 29, no. 4, pp. 321–330, Apr. 1997.

[10] A. Hoover, G. Jean-Baptise, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 673–689, July 1996.

[11] C.-K. Tang and G. Medioni, "Robust estimation of curvature information from noisy 3D data for shape description," *Proceedings of the seventh IEEE International Conference on Computer Vision, ICCV'99*, pp. 426–433, Sept. 1999.

[12] E. Natonek, "Fast range image segmentation for servicing robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 406–411, May. 1998.

[13] X. Jiang and H. Bunke, "Fast segmentation of range images into planar regions by scan line grouping," *Machine Vision and Applications*, vol. 7, no. 2, pp. 115–122, 1994.

[14] P. J. Besl, *Surfaces in range image understanding*, Springer-Verlag, 1988.

[15] X. Jiang and H. Bunke, "Range image segmentation: Adaptive grouping of edges into regions," in *Computer Vision - ACCV'98, Proceedings of the third Asian Conference on Computer Vision*, R. Chin and T.-C. Pong, Eds., pp. 726–733. Springer-Verlag, 1998.

[16] G. Lukács, R. Martin, and D. Marshall, "Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation," in *Computer Vision - ECCV'98, Proceedings of the fifth European Conference on Computer Vision*, H. Burkardt and B. Neumann, Eds., pp. 671–686. Springer-Verlag, 1998.