Comparing Curved-Surface Range Image Segmenters

Mark W. Powell, Kevin W. Bowyer

Department of Computer Science
and Engineering
University of South Florida
Tampa, FL 33620

Xiaoyi Jiang, Horst Bunke

Department of Computer Science
University of Bern
CH-3012 Bern Switzerland

Abstract

This work focuses on creating a framework for objectively evaluating the performance of range image segmentation algorithms. The algorithms are evaluated in terms of correct segmentation, over- and under-segmentation, missed and noise regions. A set of images with ground truth was created for this work. The images were captured using a structured light scanner. Images used in the evaluation contain planar, spherical, cylindrical, toroidal and conical surface patches. The different surface patches in each image were manually identified to establish ground truth for performance evaluation. Two segmentation algorithms from the literature are compared.

1 Introduction

The aim of this paper is to describe a framework for objectively comparing the performance of range image segmentation algorithms. The framework for the comparison is designed to be generally applicable to segmentation algorithms that take as input a range image of a scene containing planar and curved surfaces. This work is an extension of previous work that provides a general framework for the objective evaluation of segmentation algorithms but considered only planar surface segmentation algorithms [1]. The motivation for this work remains the same as the previous work: to provide an objective performance evaluation of segmentation algorithms, the need for which has been expressed for many years in the literature (see [6], for example).

The problem of choosing an appropriate set of curved surface types was a difficult one, as various segmenters use different surface patch types. We chose a set of primitives that are common among objects encountered in an industrial environment. To encompass 95 percent of conventional, unsculptured parts the domain must include all solids definable by the natural quadric (planar, spherical, conical, cylindrical) and toroidal halfspaces [2].

In performing this evaluation we have built upon the framework of the first comparison, which provides a method to measure not just correctness but also error measurement. We have created a data set that is greater in size (by 50%) and complexity than that of [1] in order to effectively challenge algorithms which solve the segmentation problem in the domain of both planar and curved surfaces. The results of this comparison, the data set that was used and the tools that were employed to produce it are all available on the World Wide Web at http://marathon.csee.usf.edu/range/seg-comp/SegComp.html.

The next section defines the segmentation problem and describes our comparison framework. In the third section we discuss the details of the image set, ground truth, training and testing of the algorithms. In section four we explain the segmentation algorithms we are comparing and the training process. This is followed by experimental results and discussion.

2 Problem Definition

2.1 Segmentation

Our definition of segmentation for planar surface range images is given in detail in the previous work and is summarized here [1]: For any image R consisting of $r_1 \ldots r_n$ subregions:

1. $r_1 \cup r_2 \cup \ldots \cup r_n = R$ (Every pixel belongs to a region)
2. $\forall i = 1 \ldots n, r_i$ is four-connected.
3. $\forall r_i, r_j \in R \vdash i \neq j, r_i \cap r_j = \emptyset$ (All regions are disjoint)
4. $\forall r_i \in R, P(r_i) = true$ (All pixels in a region belong to the same surface)
5. $\forall r_i, r_j \in R \vdash i \neq j$ and r_i, r_j are four-connected and adjacent, $P(r_i \cup r_j) = false$ (If two regions are four-connected and adjacent then they represent different surfaces)

6. There are noise regions in the image where no valid measurement was possible which all have the same label (violating rule 2) and for which rules 4 and 5 do not apply.

2.2 Classification of Machine Segmented Regions

When we compare the results of a machine segmentation to ground truth it is based on a threshold: the percentage of overlap between a machine-segmented region and a ground truth region. For example, if we choose a threshold of 0.8, then 80% of the pixels in one machine-segmented region must lie in or on the boundary of one ground truth region, and 80% of the pixels from that ground truth region must also lie within the machine-segmented region under consideration in order to result in a correct detection. We also define over-segmentation, under-segmentation, missed and noise regions in terms of the threshold. See [1] for details.

3 Test Data

3.1 Structured Light Scanner

There are many types of range sensors in use today, including radar (e.g., a laser range finder), triangulation (e.g., a structured light scanner), moiré, Fresnel diffraction, lens focus, etc. [3]. Among these, laser range finders and structured light scanners are currently the most commonly used types of range sensors. For our image set, we used a K^{2}T model GRF-2 structured light scanner. The imaging system registers range data to as fine as 0.1 mm precision and produces data as Cartesian triples representing 3-d points as well as an intensity image of the scene under uniform lighting from the projector [4]. We then quantize this data to a 640 × 480 8-bit raster image for space considerations. For our image set, we manually optimized the selection of minimum and maximum distance thresholds used during rasterization to ensure the least possible quantization noise.

3.2 Image Complexity

The 60 images used for the data set are taken from scenes containing planar, cylindrical, spherical, conical and toroidal object surfaces. The scenes range from simple to complex in terms of number and type of surface patches and the edges they make with one another. For all of the images except for five, surface patches which correspond to the support plane (the table), the background plane, or both are present in the image. The five images where no background is in the scene consist of only one surface patch over the entire image. An effort was made to include some scenes containing models of objects which occur in real environments, such as a doorknob, a drain pipe, etc.

Let us consider the range of dimensions of the segmentation problem. The following is a reasonably complete list of these variables and how they are represented by this data set:

- Surface patch size: from a few hundred pixels to the entire image (307200 pixels)
- Number of surface patches: 1 to 120
- Types of surface patches: planar, cylindrical, spherical, toroidal and conical
- Crease edges: continuous/discontinuous, varying edge length and angle between surface normals of the adjacent surfaces sharing an edge
- Jump edges: varying depth discontinuity and edge length
- Surface curvature: from planar surfaces with no curvature to small spherical objects with a high degree of curvature
- Compactness: varying amounts of compactness due to inter-object occlusion

3.3 Ground Truth

Ground truth is specified for each image in the data set, and consists of a segmentation into regions, a correspondence of each region to the surface in the scene that it represents, and geometric properties of the surfaces represented by these regions. Ground truth is specified on a per-pixel basis and at the region level. Some regions of the image have no corresponding surface and we denote these as either noise, shadow, cross-edge or false shadow. A noisy reading at a pixel will be either noise or false shadow, depending on whether the depth value is within the minimum and maximum distance threshold or not, respectively. An area in the scene that was visible to the camera yet never received any projected light is labeled as shadow. Where a pixel overlaps the edge between two adjacent surfaces, we label it cross-edge if it is possible for a human to detect it.

3.4 Train and Test Set Philosophy

Each segmenter used in this experiment uses a different set of parameters which adapts it to suit a variety of images. We have set aside a portion of our data set for the sole purpose of optimizing these parameters for best performance. We denote this subset of the images as the training set. Each segmenter ran for many combinations of input parameters to find their optimal values when the output is compared with the ground truth segmentation. Once this set of parameter values is determined, it is used for producing results from the remaining images which we call the test set.

4 Segmentation Methods

4.1 Variable Order Surface Fitting

Besl and Jain developed a segmentation algorithm which uses signs of surface curvature to obtain a coarse segmentation and iteratively refines it by fitting bivariate polynomials to the surfaces [5]. The algorithm begins by estimating the mean and Gaussian surface curvature at every pixel and uses the signs of the curvatures to classify each pixel as belonging to one of eight surface types. The resultant coarse segmentation is enhanced by an iterative region growing procedure. For every coarse region, a subregion of a size at or above a threshold is selected to be a seed region. Low order (up to degree four) bivariate polynomials are used to produce an estimated surface fit to the seed region. Next, all pixels in all regions of the image that are currently outside the seed region are tested for possible inclusion into the current region. Two tests are used to determine compatibility for inclusion. First, the difference between the depth value at the pixel in the image and at the same location on the fitted surface must fall within a threshold. Second, the estimated normal to the surface in the image is compared to the normal of the fitted surface, and the angular difference between the normals must fall below another threshold. The largest connected region which is composed of pixels in the seed region and pixels that pass the compatibility tests is chosen as the new seed region. Expansion continues until either there is almost zero change in region size since the last iteration, or when the surface fitting error becomes larger than a threshold. Finally, fit error is calculated, and if it falls below a threshold the region is accepted. If not, the region is rejected and the seed region that produced it is marked off so that it may not be used again.

4.1.1 Training

This segmenter has 38 parameters which can affect the results of the segmentation. Of these, we selected 10 which were of highest practical importance. A 10 dimensional search space is daunting in terms of required time to find an optimal set of values, so we consulted with Besl for advice on how best to rank the parameters in order of magnitude of impact on performance. He advised us that the RMS surface fit error threshold is of chief importance to performance considerations since it varies with the amount of noise present in the image to be segmented. The additional parameters were ranked in order of significance to performance after studying the implementation of the algorithm. The order of importance is supported by the non-decreasing number of correct detections and the decreasing change in total correct detections with each parameter optimization. Table 4.1.1 is a list of all the parameters we trained and the values we chose for them. First, we varied the value of the RMS surface fit error threshold, which is sensitive to the degree of noise in the image and ranked highest in importance relative to performance. We selected the optimum value for this parameter and then used this value when selecting the next most important parameter value, and so on for the remaining ones. For each run, we sampled the parameter space starting with the default value until a local maximum in the number of correct detections was found. Due to the long running time of the implementation of this algorithm on our images (an average of six hours per image on a Sun SPARCstation 10), we trained it first on five of the twenty designated training images. Finally, we ran the segmenter on the rest of the training set images where we set the two most important parameters to three values below and three values above their optimized values. We found that there was no improvement in the performance of the segmentation. This indicates that our training process brought us to a reasonable point of optimality in performance for our images.

4.2 Edge Grouping

The UB segmenter [8] consists of two parts: edge detection and grouping of edge points into closed regions. The edge detection algorithm is described in [7]. It makes use of the fact that each scan line (row, column or diagonal) of a range image is a curve in 3-d space. Therefore, we partition each scan line into a set of curve segments by means of a splitting method. All the splitting points represent potential edges. The jump and crease edge strength of the edge candidates are evaluated by analytically computing the height difference and the angle between two adjacent curve segments, respectively. Each pixel can be assigned up to four edge strength values of each type (jump and crease) from the four scan lines passing through the pixel. These edge strength values are combined by taking the maximum to define the overall edge strength of each type. A pixel is considered as an edge point if at least one thresholding operation is successful.

The grouping process is based on a hypothesis generation and verification approach. From the edge map, regions can be found by a component labeling. Due to the inevitable gaps in the edge chains, however, this initial grouping usually results in an under-

Parameter name	Values assigned	Correct detections
RMS error limit in hundredths	420*, 1000, 5000, 6000, 6250, 6500, 6750, 7000, 7250, 7500, 7750, 8000	53
Error limit factor for growth	200, 270, 275, 280*, 285, 290, 300, 350	53
Pre-smooth window size	7*, 9, 11, 13, 15	81
Derivative window size	7*, 9, 11, 13, 15	81
HK-smooth window size	5*, 7, 9, 11, 13	81
Min. initial region size	30*, 60, 120, 200, 240, 300, 480	84
Max. number of iterations	30*, 40, 50, 60	84
Acceptability error factor	100, 150*, 300, 500	84
Max. blackboard region size	40, 50*, 70, 90	84
Regions test threshold	10, 20*, 40, 60	84

Table 1: Besl and Jain's segmenter parameters and values. Underlined values are the optimal values for our data set. An asterisk denotes the default value for each parameter. We trained with threshold values of 51%, 60%, 70%, 80% and 90%. The correct detections are out of 200 possible.

segmentation. To recognize the correctly segmented and under-segmented regions, we perform a region test for each region of the initial segmentation. If the region test is successful, the corresponding region is registered. Otherwise, the edge points within the region are dilated once, potentially closing the gaps. Then, hypothesis generation (component labeling) and verification (region test) are carried out for the region. This process is recursively done until the generated regions have been successfully verified or they are no longer considered because of too small a region size. The region test starts with a plane test. The principal component method is used to compute the plane function of a region. The region is regarded as a plane if both the RMS and average fit error are small enough. If this test fails, we compute a second surface approximation using a bivariate polynomial function of degree four. After the grouping process, a post-processing step is performed to merge the pixels not labeled so far.

The UB segmenter has a fundamental limitation. The edge detection method described above is able to detect jump and crease edges but not smooth edges (discontinuities only in curvature). This seems to be true for all edge detectors reported in the literature. Therefore, two surfaces meeting at a smooth boundary will not be separated and an under-segmentation occurs.

4.2.1 Training

The UB segmenter [8] has ten parameters. For the edge detection, the parameter T_g controls the scan line partitioning precision. Two thresholds T_j and T_c are needed to produce jump and crease edges. In the grouping process, two thresholds T_r^p and T_a^p represent the tolerable RMS and average fit error for the plane approximation, and another two thresholds T_r^c and T_a^c for the bivariate polynomial function approximation. The minimum region size is represented by T_s. Finally, the parameters T_f^p and T_f^c for the post-processing step express the tolerable fit error with regard to a planar and curved region, respectively. After some experiments, two parameters were fixed: $T_j = 1.0, T_s = 200$. The values for T_r^p, T_a^p, T_r^c, and T_a^c were determined from the training images in the following way. The four fit errors of all regions from the training images were computed and histogrammed. Then, the four fit error thresholds were chosen such that almost all regions of the training images will successfully pass the region test. Their values are $T_r^p = 0.11, T_a^p = 0.07, T_r^c = 0.11$, and $T_a^c = 0.09$. The thresholds T_f^p and T_f^c were set to be twice the average fit error threshold T_a^p and T_a^c, respectively. For the parameters T_g and T_c, we localized a good region in the parameter space after some tests using arbitrarily chosen values, namely $R : (T_g, T_c) \in [0.4 \ldots 0.5] \times [25.0 \ldots 35.0]$. The goodness of this region was verified by two methods. First, nine combinations of parameters $(T_g, T_c) \in [0.4, 0.45, 0.5] \times [25.0, 30.0, 35.0]$ were run on the training images and the segmentation results were compared with the ground truth through visual inspection and by using the comparison tool. Secondly, tests on 200 randomly selected parameter pairs within the region R were carried out and the segmentation results were evaluated by the comparison tool. It turned out that within R the segmenter was very stable. For all 200 test series the average values of the five performance measures (correct detection, over-segmentation, under-segmentation, missed, and noise) were similar. Finally, we selected the mean value of the region R as $T_g = 0.45, T_c = 30.0$.

5 Experimental Results

The best performance for correct detections (at a moderate comparison threshold value of 80%) was an average of 68.2% for the test set images. Table 5 shows the average scores in correct detection, over- and under-segmentation, missed and noise regions for the University of Bern (UB) and Besl and Jain (BJ) segmenters for the test set images. In order to find the "best" segmenter, we would need to find one having the highest score in correct detections and also the lowest score in all the error measures. Although the UB segmenter was highest in correct detections and lowest in over-segmentation, missed and noise regions, it was higher than the BJ algorithm in under-segmentation.

Classification	Threshold Value	Average Percent BJ	Average Percent UB
Correct	0.51	15.33	68.65
	0.60	16.06	68.65
	0.70	16.65	68.65
	0.80	16.12	68.2
	0.90	16.14	64.02
Over-segmentation	0.51	62.94	2.29
	0.60	61.90	2.29
	0.70	58.64	2.29
	0.80	54.48	2.2
	0.90	43.29	0.85
Under-segmentation	0.51	5.47	12.57
	0.60	4.31	12.57
	0.70	4.27	12.57
	0.80	3.64	12.34
	0.90	1.67	10.05
Missed	0.51	10.13	1.97
	0.60	12.80	2.56
	0.70	15.86	2.68
	0.80	21.78	4.52
	0.90	37.23	14.9
Noise	0.51	18.79	1.45
	0.60	20.75	1.45
	0.70	24.3	1.45
	0.80	30.82	2.59
	0.90	44.88	13.98

Table 2: Results from the BJ and UB segmenters on the test set images.

5.1 Results from Variable Order Surface Fitting (BJ)

The results for the BJ segmenter were not as good as we expected. The segmentations that it generated exhibit a large amount of over-segmentation and noise. We believe this has to do in part with differences in the type of range data we presented to it versus the type of data it was originally tested with. In its published results [5] this segmenter was given range images taken with laser range finders (primarily the ERIM scanner) which measure depth at pixel level. The data from the GRF-2 structured light scanner is measured at the stripe level followed by interpolation to estimate values at unknown pixels. If the interpolation procedure has a margin of error such that it does not produce data points for a smooth surface, then this introduces noise into the range data. Figure 5.1 shows a plot of the quantized range values taken from an image of a planar surface. Instead of a straight line, we see a jagged line that is roughly linear. Upon examination of the segmentation results produced by the BJ segmenter after training, it is apparent that this kind of noise is not handled gracefully by the algorithm. Figure 5.1 shows an example of an image segmentation from the BJ algorithm which is over-segmented due to the presence of interpolation error, indicated by the presence of the horizontal line-shaped regions. There are three actual regions in this image: planar, conical and planar, from left to right. The leftmost planar patch is over-segmented by four regions, the rightmost planar patch is over-segmented by 14 regions, and the center conical patch is over-segmented by five regions.

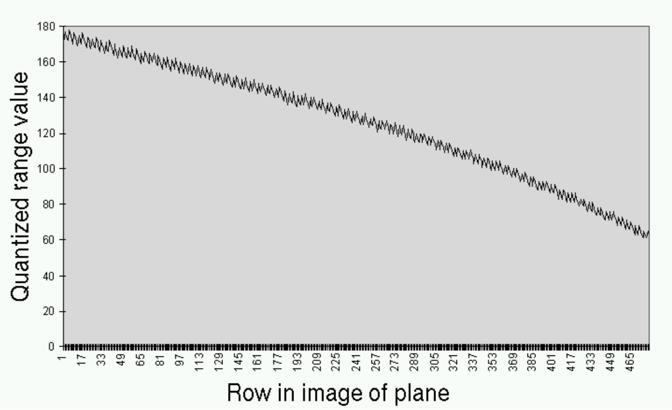

Figure 1: Quantized depth values from the center column of a range image of a plane. The deviations from a straight line indicate the presence of interpolation noise.

5.2 Results from Edge Grouping (UB)

Although the UB algorithm had the highest number of correct detections and the fastest runtime (23 seconds on average on a Sun SPARCstation 5), it also had the highest amount of under-segmentation. Whenever two adjacent regions differ only in curvature and not

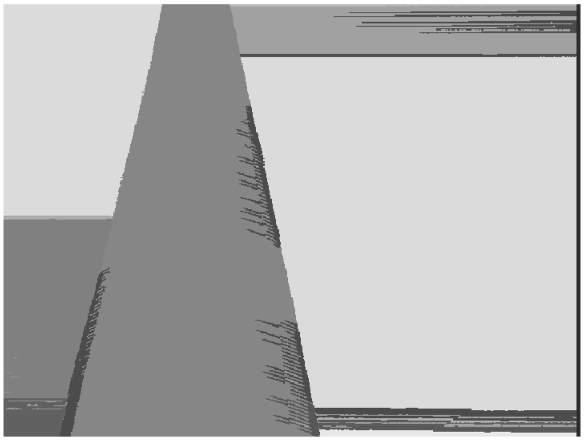

Figure 2: Segmentation of a cone (foreground) and two planar patches (background) from the BJ algorithm.

orientation, under-segmentation results. Figure 5.2 shows an example of a range image where a cylindrical region and planar region are adjacent and result in an under-segmentation by the UB algorithm.

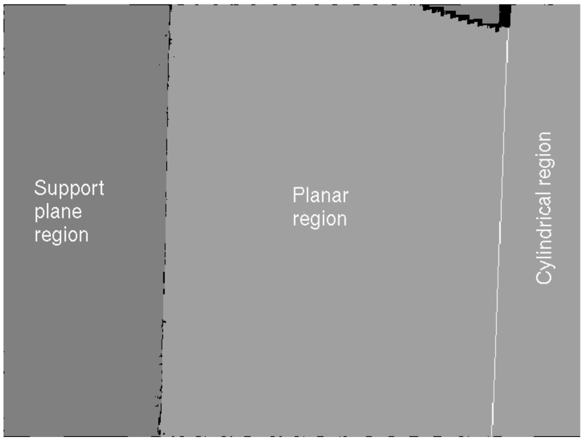

Figure 3: A segmentation of a scene where a plane and cylinder share a crease edge. The white line indicates the separation between the cylinder and plane that was not detected.

6 Discussion

In this comparison work we have taken a substantive first step toward extending a practical framework [1] towards quantitatively measuring the performance of curved surface range image segmentation algorithms. The results indicate that the range segmentation problem is not solved. We have identified four major sources of error (over- and under-segmentation, missed and noise regions) and have quantified their contributions towards failure to make correct detections of regions.

The results of the BJ segmenter show a large error due to over-segmentation. This error is due to the interpolation noise that exists in the images taken with the GRF-2 SLS. We are currently recapturing the entire image set using new, custom software which interpolates correctly and reduces this noise substantially. We will then reevaluate these algorithms and others and report our new results.

Based on our results, we can recommend the UB segmenter over the BJ segmenter based on its highest average percentage of correct detections, relatively low error measures (except for the higher average percent under-segmentation) and speed of computation.

References

[1] A. W. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. W. Bowyer, D. Eggert, A. Fitzgibbon and R. 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.

[2] A.A.G. Requicha and H.B. Voelcker. Solid modeling: a historical summary and contemporary assessment. *IEEE Computer Graphics and Applications,* Vol. 2, No. 1, pp. 9-24, March 1982.

[3] Paul J. Besl. Active, Optical Range Imaging Sensors. *Machine Vision and Applications,* Vol. 1, pp. 127-152, 1988.

[4] K2T, Inc. GRF-2 Instruction Manual, 1995.

[5] Paul J. Besl and Ramesh C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 10, No. 2, pp. 167-192, March 1988.

[6] R. C. Jain and T. O. Binford. Ignorance, Myopia, and Naiveté in Computer Vision Systems. *CVGIP: Image Understanding,* Vol. 53, No. 1, pp. 112-117, 1991.

[7] X. Jiang, H. Bunke. Robust and fast edge detection and description in range images. *Proc. of IAPR Workshop on Machine Vision and Applications,* Tokyo, pp. 538-541, 1996.

[8] X. Jiang and H. Bunke. Range image segmentation: Adaptive grouping of edges into regions. Accepted to appear at the *Asian Conference on Computer Vision*, Hong Kong, 1998.