Robust and Fast Edge Detection and Description in Range Images

X.Y. Jiang, H. Bunke *
Department of Computer Science, University of Bern, Switzerland

Abstract

A novel algorithm is presented for robust and fast edge detection and description in range images. In contrast to many other known methods, our algorithm provides a rich description of edge structures including detailed edge types and local surface characteristics. The proposed algorithm has been tested on a large number of real range images acquired by three scanners with quite different noise characteristics, and demonstrated good results. The computation time for one image lies between 1.5 and 10 seconds on a Sun Sparcstation 5, dependent on the image resolution. The highly parallel nature and the simple control structure of our algorithm makes a further significant speedup on a parallel architecture possible.

1 Introduction

Today range scanners are able to acquire high-resolution and high-quality range images in (quasi-) real-time [2]. However, range image analysis and interpretation in a robust and fast way is still an open research topic. In particular, there is considerable room for improvement in range image segmentation with respect to both quality and speed [6, 7]. In the past we have developed two very fast region-based segmentation algorithms based on a scan line grouping technique: a method for segmenting a range image into planar surfaces [8] and its extension to curved surfaces [9]. In a recent experimental comparison of segmentation algorithms using two large range image sets acquired by two different scanners, this scan line grouping technique has demonstrated good results at high speed [6, 7]. It is our intention to further improve the robustness of our segmentation algorithm by incorporating edge information, and this has motivated us to develop a robust and fast edge detection method.

The central task of edge detection is to reliably detect and locate edge points. But a rich description of edge points including edge strengths, detailed edge types (see [11]), and local surface characteristics is highly desirable, too. A number of edge detection algorithms for range images exist in the literature, see [10] for a review. However, they suffer from several drawbacks, for instance,

- no quantitative characterization of edge strengths [11],
- no straightforward geometric interpretation of the quantitative characterization of edge strengths [1, 5],
- no support of the classification in detailed edge types [1, 3, 5].

Based on the same scan line grouping technique as in [8, 9] we propose an edge detection algorithm that enables us to quantitatively investigate both detailed edge types and local surface characteristics. This way we are able to achieve a robust and fast edge detection and description method for range images.

2 Scan line grouping technique

For description clarity we assume a dense range image $z(x, y)$ regularly sampled in both coordinate directions. Our algorithm can easily be extended to other scanner geometries, see [10] for details. Let $z = f(x, y)$ be the 3-D surface of an object to be segmented. Then, an image row $z(x, y_0)$ with a constant y_0 is simply a two-dimensional curve $z = f(x, y_0)$ in the $x - z$ plane. A planar 3-D surface $z = ax + by + c$, for example, results in a straight line $z = ax + \hat{b}$, $\hat{b} = by_0 + c$, on the image row. Generally, curve segments $z = f(x, y_0)$ corresponding to different surfaces have different function parameters. So, we can partition an image row into a set of curve segments. In the ideal case all partitioning points lie on the boundary between two surfaces and are therefore edge points.

In this work we assume the quadratic surface model $z = \sum_{i+j \leq 2} a_{ij} x^i y^j$ that leads to curve segments $z = ax^2 + bx + c$ in image rows with constant $y = y_0$. The classical splitting algorithm described by Duda and Hart [4] has been used to partition an image row into curve segments under a preselected threshold ϵ. A quadratic approximation function is first determined for a whole image row based on the midpoint and the two endpoints. Then, whenever the largest error $e_{\max}$ between the approximation function and the image row is greater than ϵ, the image row is split into two parts at the location where $e_{\max}$ occurs. The splitting algorithm proceeds recursively until the approximation error $e_{\max}$ doesn't exceed the threshold ϵ.

*Address: Neubrückstrasse 10, CH-3012 Bern, Switzerland. E-mail: jiang,bunke@iam.unibe.ch

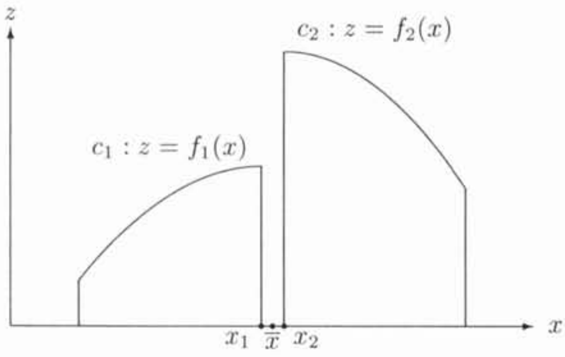

Figure 1: Definition of discontinuity strength at the boundary of two curve segments.

3 Edge detection and description

We only consider the end points of a curve segment as potential edge points. All other pixels are on a smooth surface and are thus excluded from further investigation. For each edge candidate x_1 (see Figure 1) a *discontinuity strength* is defined in the following way. Let x_2 be the end point of the curve segment adjacent to x_1 and $z = f_{1,2}(x)$ be the function of the two curve segments c_1 and c_2. The usual definition of jump edge strength is the difference of depth values of adjacent pixels, i.e., $|f_1(x_1) - f_2(x_2)|$ in our case. This definition, however, is not adequate since at a constant sampling density for the whole scene two adjacent pixels on a highly sloped surface may have quite different depth values. Instead, we consider the midpoint $\overline{x} = (x_1 + x_2)/2$. Its expected depth value on c_1 and c_2 are $f_{1,2}(\overline{x})$, or alternatively $z_1 + f'_1(x_1)(\overline{x} - x_1)$ and $z_2 - f'_2(x_2)(x_2 - \overline{x})$, respectively, where $z_{1,2}$ is the z-value of $x_{1,2}$. Then, a suitable discontinuity strength for jump edges is given by

$$|f_1(\overline{x}) - f_2(\overline{x})|,$$

or

$$|(z_1 + f'_1(x_1)(\overline{x} - x_1)) - (z_2 - f'_2(x_2)(x_2 - \overline{x}))|.$$

The expected normal vectors at $\overline{x}$ on c_1 and c_2 are $(-f'_{1,2}(\overline{x}), 1)$, respectively. In this case the angle difference between the two normal vectors

$$\cos^{-1} \frac{(-f'_1(\overline{x}), 1) \cdot (-f'_2(\overline{x}), 1)}{||(-f'_1(\overline{x}), 1)|| \cdot ||(-f'_2(\overline{x}), 1)||}$$

provides a good definition of discontinuity strength for crease edges. Alternatively, we may also express the crease edge strength by

$$\cos^{-1} \frac{(-f'_1(x_1), 1) \cdot (-f'_2(x_2), 1)}{||(-f'_1(x_1), 1)|| \cdot ||(-f'_2(x_2), 1)||}$$

using the normal vectors at $x_{1,2}$.

The image row partition is controlled by the threshold ϵ. It should be set small enough so that we will not miss any edge point. On the other hand, a too conservative strategy may result in an overpartitioning of image rows and report too many nonedge points as edge point candidates. However, this doesn't cause any serious problem since then both discontinuity strength values defined above will have very small values compared to true edge points.

Dependent on the configuration of surfaces in the scene, the maximal value of the discontinuity measurements may not be observed in the horizontal direction. To capture the information available in the scene to a larger extent we carry out the operations described above also in the vertical (image columns) and the two diagonal (45° and 135°) directions. A pixel thus has at most four different discontinuity measurements of each type. We combine them by simply taking the maximum.

In our experiments we found out that for noisy range images the position of edge points as determined by the simple splitting algorithm is not very precise. To solve this problem we have developed an edge position adaptation method with subpixel accuracy. We use the functions $z = f_{1,2}(x)$ of two adjacent curve segments and compute the intersection point. Then, the intersection point is backprojected into the image plane to get its pixel coordinate. This pixel is considered as the improved edge position if it is within a distance of a preset number of pixels from the original boundary position of the two curve segments. Instead of the functions $z = f_{1,2}(x)$ provided by the splitting algorithm, we can also make use of another function $z = \overline{f}_{1,2}(x)$ computed by the least square method for each curve segment. As a matter of fact, this latter approach gives us the best edge position adaptation results.

After edge detection it becomes trivial to build a rich description of edge maps. Detailed edge types as suggested in [11] (positive and negative crease edges, positive and negative roof edges, a.s.o), for example, can be easily determined using the functions of the two segments adjacent to an edge point (see Figure 1). For further discussions and examples, see [10]. Furthermore, the local surface characteristics are described by the neighboring curve segments in four different directions.

4 Accuracy considerations

In the ideal case the strength of a crease edge point can be quantitatively characterized in the following way. Two surfaces meet at a boundary containing the edge point under consideration. If the two surfaces are assumed to be locally planar and modeled by the surface functions $z = a_1x + b_1y + c_1$ and $z = a_2x + b_2y + c_2$, respectively, then the ideal edge strength is defined by the angle between the normals of the two planes:

$$\cos^{-1} \frac{(-a_1, -b_1, 1) \cdot (-a_2, -b_2, 1)}{||(-a_1, -b_1, 1)|| \cdot ||(-a_2, -b_2, 1)||},$$

that is independent of the position and orientation of the scene relative to the range scanner. An edge

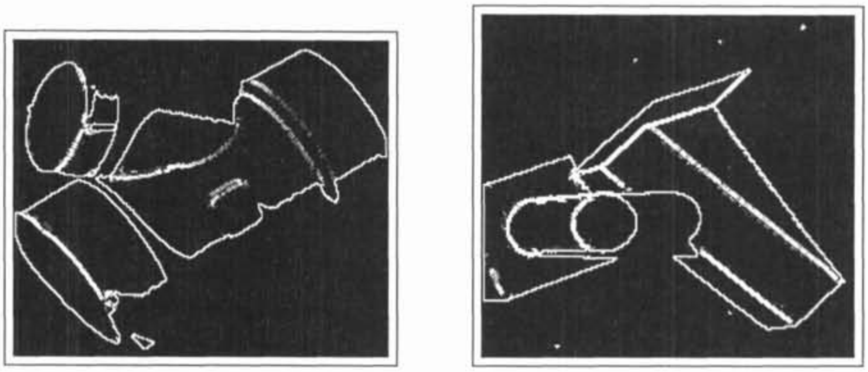

Figure 2: Edge detection for two Michigan images.

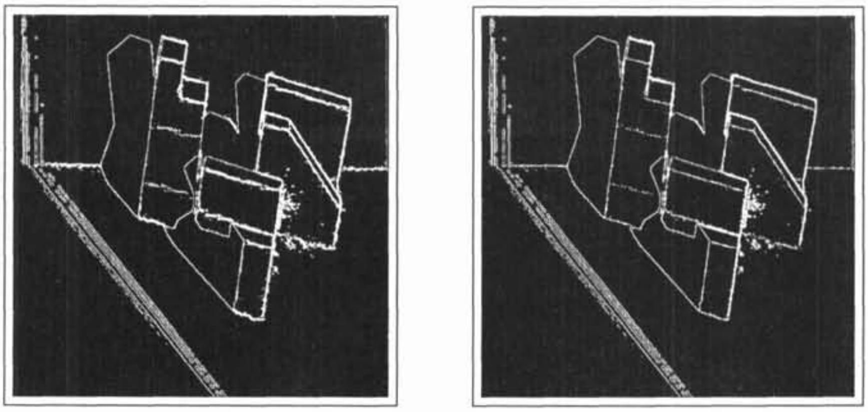

Figure 3: Edge detection for an ABW image (left). The edge position adaptation method significantly improves the accuracy of edge localization (right).

detector that provides this edge strength is regarded as *optimal*.

On the other hand, our edge detection algorithm considers only directional sections of a scene. Even though totally four directional sections are taken into account, the computed crease edge strength is still different from the optimal values defined above. We are interested in the amount of this deviation and thus the question to which extent our edge detector is an optimal one.

For this purpose we have carried out simulation tests. Since the terms $a_{1,2}$ and $b_{1,2}$ represent the slope with respect to the x- and y-axis, respectively, we consider for each term 29 different values which correspond to the value of the angle to the coordinate axis, ranging from $-70°$ to $70°$ at a step size of $5°$, and determine for each combination (a_1, b_1, a_2, b_2) the difference between the edge strength computed by our edge detection algorithm and the optimal value. For all combinations this difference value has an average of $3.8°$ and a standard deviation of $4.2°$. Importantly, the largest difference values are observed if one of the two surfaces is highly sloped with respect to at least one coordinate axis. For more details of the simulation tests, see [10]. These simulation results demonstrate the accuracy of our method compared to the optimal edge detector.

5 Experimental results and discussions

The proposed algorithm has been implemented in C on a Sun Sparcstation 5 and tested on a large number of range images (about 280) taken by three range scanners with quite different noise characteristics. The first image source is the popular image set from the PRIP Lab of Michigan State University and another 38 images from the same Lab [12], both acquired by a Technical Arts Scanner. We have also used 80 range images containing only polyhedral objects. These images were acquired by a structured light ABW scanner [14] and a Perceptron time-of-flight scanner [13]. They constitute the test data in a recent experimental comparison of range image segmentation algorithms [6, 7]. Note that the images from Michigan are approximately regularly sampled in both coordinate directions, while this property is not given in neither ABW nor Perceptron images. For all test images of each scanner, the same threshold ϵ was used. Figure 2 shows the results for two images from the Michigan set where the grey levels are proportional to the discontinuity strength. The two images contain mainly curved surfaces (left) and planar surfaces (right), respectively, and illustrate that our algorithm is able to handle both types of surfaces in a unified framework. The edge strength map for

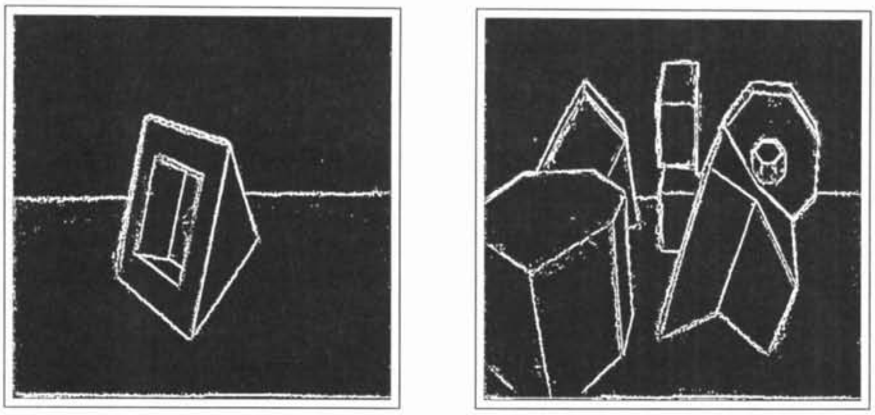

Figure 4: Edge detection for two Perceptron images.

an ABW image is presented in Figure 3 (left). The images acquired by this scanner are more noisy than the Michigan images. This makes the precise localization of edge points difficult. Our edge position adaptation method works very well on these images, as illustrated in the right part of Figure 3. In Fig. 4 the results for two Perceptron images are shown.

The computation time for the Michigan images of a typical resolution of 200×200 pixels is about 1.5 second. All ABW and Perceptron images have a resolution of 512×512 pixels and require thus more computation (about 10 seconds). The speed of our edge detection and description method surely lies in the scan line grouping technique. Compared to the number of pixels, a very small number of curve segments result from the scan line partitioning. We only consider the end points of the curve segments as edge candidates. In our method the discontinuity measurements have straightforward geometric interpretation and low computational expense. Moreover, our algorithm is of highly parallel nature. The processing of each image row, column and diagonal can be done independently and a further significant speedup can be achieved on a parallel architecture.

Acknowledgments

We thank the Pattern Recognition and Image Processing Lab of Michigan State University of making their database of range images for public usage and K. Bowyer, Univ. of South Florida, Tampa, for providing us the Perceptron images.

References

[1] E. Al-Hujazi, A. Sood, Range image segmentation with applications to robot bin-picking using vacuum gripper, IEEE. Trans. on SMC, 20(6), 1313–1325, 1990.

[2] C.M. Bastuscheck, Techniques for real-time generation of range images, Proc. of CVPR, 262–268, 1989.

[3] J. Berkman, T. Caelli, Computation of surface geometry and segmentation using covariance techniques, IEEE Trans. on PAMI, 16(11), 1114–1116, 1994.

[4] R.O. Duda, P.E. Hart, Pattern classification and scene analysis, Wiley, New York, 1972.

[5] S. Ghosal, R. Mehrotra, Detection of composite edges, IEEE Trans. on Image Processing, 3(1), 14–22, 1994.

[6] A. Hoover *et al.*, Range image segmentation: The user's dilemma, Proc. of Int. Symposium on Computer Vision, Coral Gables, Florida, 323-328, 1995.

[7] A. Hoover *et al.*, An experimental comparison of range image segmentation algorithms, IEEE Trans. on PAMI, 18(7), 673–689, 1996.

[8] X.Y. Jiang, H. Bunke, Fast segmentation of range images into planar regions by scan line grouping, Machine Vision and Applications, 7(2), 115–122, 1994.

[9] X.Y. Jiang, U. Meier, H. Bunke, Fast range image segmentation using high-level segmentation primitives, Proc. of 3rd IEEE Workshop on Applications of Computer Vision, Sarasota, Florida, 1996.

[10] X.Y. Jiang, H. Bunke, Robust and fast edge detection and description in range images, Technical Report, Dept. of Computer Science, Univ. of Bern, 1996.

[11] R. Krishnapuram, S. Gupta, Morphological methods for detection and classification for edges in range images, Journal of Mathematical Imaging and Vision, 2, 351–375, 1992.

[12] G.C. Lee, G.C. Stockman, Obtaining registered range and intensity images using the Technical Arts Scanner, Technical Report CPS-91-08, Dept. of Computer Science, Michigan State University, East Lansing, 1991.

[13] Perceptron Inc., LASAR Hardware Manual, 23855 Research Drive, Farmington Hills, Michigan, 1993.

[14] T.G. Stahs, F.M. Wahl, Fast and robust range data acquisition in a low-cost environment, Proc. of SPIE#1395: Close-Range Photogrammetry Meets Machine Vision, 496–503, 1990.

[15] M.A. Wani, B.G. Batchelor, Edge-region-based segmentation of range images, IEEE Trans. on PAMI, 16(3), 314–319, 1994.