MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

A Novel Approach for Detection of Edges in Range Images Using Splines

Satish Kaveti, Eam Khwang Teoh and Han Wang

School of Electrical and Electronic Engineering
Nanyang Technological University, Nanyang Avenue, Singapore

Abstract

In this paper we propose an approach for detection of edges of range images using a local statistics for detection of crease edges. The jump edges are first located using the ratio of the slopes as a measure. For the detection of crease edges, curvature is estimated at all the data points using the first and second order derivatives of the smoothing spline. The local maximas of curvature are considered as the candidates for the crease edges. This condition is not a very strong constraint, as a result of which a large number of pixels are marked as crease edges. A statistical test is done over the image, and the distribution of the curvature values within a window is used for the detection of crease edges.

1 Introduction

Range image segmentation can be based on edge-based or region based approach. Edge based approaches have an advantage of reduced computational complexity and it can be used in conjunction with the region growing to obtain a better segmentation of the range images [1]. Edges have been a valuable feature for recognition and localization of three-dimensional objects. Bolles and Horaud [2] used distinctive edge feature for localization of the object. Fan, Medioni and Nevatia [3] used edges for extraction of the features, and, Kriegman and Ponce [4] used image contours derived from intensity images for localization.

The edges can be broadly classified into step edges, crease edges and smooth edges. In this paper we shall be concentrating on the step and crease edges only. The presence of noise and the presence of edges at various scales makes the problem of detection of creases a difficult one. A single threshold for the entire image may either result is missing of the weak edges or mislabeling of the highly curved surfaces as crease. The edge detectors used for intensity images have been found suitable only for step edges. The Canny's edge detector [5] for gray level images is suitable for step edges only. Many approaches for detection of crease edges have been based on first smoothing the image and then estimating the curvature extrema at various scales.

This paper describes an approach for detection of jump and crease edges for range images. Smoothing splines are used for estimation of the first and second order derivatives, using which, the curvature values are estimated along two orthogonal directions. The image is divided into non-overlapping windows and the edge detection is applied on each of the windows separately. This permits a high degree of parallelization for the algorithm. A statistical test is done on each of the windows using the histogram of the curvature values. This local approach ensures that the weak edges are also detected, while at the same time the noisy / textured surfaces are not labeled as edges.

2 Preprocessing and detection of jump edges

The range image is obtained using a Technical Arts 100X Active Range Scanner from Michigan State University. The original image is available as three 2D arrays corresponding to the x, y, and z co-ordinates of the range pixel. The x-coordinate of the range pixel remains the same along the rows of the range image. However, the y-coordinate does not remain the same along the columns of the range image. As we intend to use interpolation of planar curves, the y-coordinates are computed on a rectangular grid using piecewise linear interpolation. To avoid oversmoothing across the jump edges the linear interpolation is inhibited at points lying between a two jump edges. After the linear interpolation, the points lie on a rectangular grid.

2.1 Detection of jump edges

For the detection of jump edges in range images, the gradient based operators like Sobel operator are not suitable. Such gradient based operators, would mark edges even for a uniform region having a high gradient. Reducing the gradient, would result in missing of the actual edges. In our approach, for detection of edges in the y-direction, we compute

$$\delta_{(i,j)_1} = |z_{i,j} - z_{i,j-1}|$$

$$\delta_{(i,j)_2} = |z_{i,j+1} - z_{i,j}| \cdot \frac{y_{i,j} - y_{i,j-1}}{y_{i,j+1} - y_{i,j}} \quad (1)$$

and a pixel at (i, j) is labeled as a jump edge if

$$\frac{\max\{\delta_{(i,j)_1}, \delta_{(i,j)_2}\}}{\min\{\delta_{(i,j)_1}, \delta_{(i,j)_2}\}} > 10. \quad (2)$$

The scaling in Eq. 1 is done to compensate for the non-uniform sampling of the data. This criterion has been found to be a very simple and effective for detection of jump edges. The position of the jump edges is detected along the rows and columns of the image.

3 Detection of crease edges

Similar to the jump edges, the crease edges are first found along the rows (y direction) and then along the columns (x direction). The location of the crease edge is marked as local maxima of the curvature. For planar curves the curvature is given by

$$\text{Curvature} = \frac{d\vec{T}}{ds} = \frac{\frac{d^2y}{dx^2}}{\left(1 + \left(\frac{dy}{dx}\right)^2\right)^{3/2}} \quad (3)$$

which requires an estimate for the first and second derivative. To estimate the first and second derivative, we fit the data with smoothing cubic splines. The main motivation for using the smoothing splines is that it minimizes

$$p \sum_{i=1}^{N} \left(\frac{y_i - f(x_i)}{\delta y_i} \right)^2 + (1-p) \int_{x_1}^{x_N} (f^{(m)}(t))^2 \, dt \quad (4)$$

where $m = 2$ in the case of cubic splines and p tries to provide a comprise between our desire for our approximant to be smooth and at the same time it should be able to approximate the data points closely. In our experiments, we have chosen $p = 0.9999$ which results in the interpolant to follow the data points very closely. This has a serious side effect that our estimate of the second derivative is not a smooth estimate, making the localization of the crease edges difficult. If we increase the smoothing, then there are two main problems, namely,

- the maxima of the curvature is away from the true location of the maxima
- small creases are smoothened out and thereby making them difficult to detect.

This behavior is similar to the scale-space filtering which has been investigated by a number of researchers in the past.

The spline interpolation of the data is inhibited at the jump edges to avoid over-smoothing. The position of the jump edges are used as break points for spline interpolation.

All the pixels for which the absolute value of the curvature is higher than its neighbors are marked as candidates for the crease edges. We shall call these points as the *local curvature peaks*. This condition of local maxima is a very weak condition, as a result of which the procedure generates a large number of *local curvature peaks*. For example, for the cup-block image shown in Fig. 1, the number of *local curvature peaks* were around 6000 which is much higher than the actual number of crease edge pixels. We use a statistical approach for removing extraneous curvature peaks generated by noise. At each of the *local curvature peaks* we also associate with it the value of the curvature at that pixel.

3.1 Statistical test for detection of crease edges

The extraneous crease edges are removed by inferring about the presence of the edge using the histogram of the curvature values of the *local curvature peaks*. In Fig. 2, we have plotted the histogram for the estimated curvature for a window of size 20×20 for three different cases, i.e.,

- crease not present
- crease imbedded with surrounding noise peaks
- isolated crease.

The regions of interest have been highlighted in Fig. 1. As we can see that in the case when the crease is present, the peak of the distribution is near the origin, which confirms our belief that the noisy peaks have *locally* smaller value of curvature as compared to its value for a neighboring actual crease. The number of intervals in the histogram is decided so that,

$$\text{Number of intervals} = \frac{N}{d}$$

where N is the number of local curvature peaks in the window and d is the desired average. It has been done to normalize the histogram. If the parameter curvature was a uniform random variable, then its histogram would have an height of d.

Depending upon the nature of the histogram, the crease edges are identified. The criterion used for identification of the edges for different cases are

- If the peak of the histogram lies close to the origin and the level is greater than a certain level, then this implies a presence of a crease in the window. The threshold is taken as the value at which the peak at the origin fall below d, and all the pixels having the value of curvature above this threshold are labeled as crease edges.
- If the histogram has a large gap in between, then it implies that the number of extraneous creases are small and the population is primarily dominated by edge pixels. All the pixels lying beyond this gap are labeled as crease edges.

- If neither of the above conditions are satisfied, then the window is assumed not to contain any edges.

We partition the entire image into non-overlapping windows of size $M \times M$ (in our example $M = 20$). In each of the windows the above test is applied for the detection of crease edges.

All the pixels having curvature greater than the the threshold derived from the statistical test are candidates for crease pixels. To avoid the problem of incorrectly labeling outlier data as crease pixel, the following criterion is used for labeling :

1. the pixel should have curvature greater than the threshold and its two neighboring local curvature peaks should also have curvature more than the threshold.

2. the sign of the curvature of the center pixel should be same as that of its neighbors.

If the above two criterion are satisfied then the center pixel and the neighboring pixels are labeled as crease edges. The result of this operation is shown in Fig. 3. As we can, there are still few stray streaks of crease points. These stray crease are removed on the on the basis of the length of the crease edges. All the crease edges having length less than 10 pixels are removed. The result of this operation is shown in Fig. 4. This is the only threshold in the algorithm which is non-adaptive and is not inferred from the image.

3.2 Edge growing

The edge pixel chains at this stage, are either smaller than their actual size or disconnected. This stage involves growing of the edges using the following two criterion.

1. the new pixel should have the same sign of the curvature

2. the new pixel should not have more than two eligible neighbors. In case, a pixel having more than two neighbors is encountered, the edge growing operation is stopped. The basis for this criterion is that if a pixel is a crease edge, than normally it does not have stray pixels as its eligible neighbors.

4 Results and Discussion

Using the above approach, and the detected crease edges are depicted in Fig. 5. The number of pixels marked as crease edges are 635 which quite better then the starting figure of > 6000 for *local curvature peaks*. Even the small wedges present in the block was detected satisfactorily. Due to the its statistical nature, the algorithm is not highly sensitive to noise.

The results for edges detection for different objects are shown in Fig. 6, Fig. 7 and Fig. 8.

The results of the above algorithm have been very encouraging. The main advantage of the algorithm is that it is highly parallelizable and can detect weak edges, while not labeling regions have very high curvature as edges. The algorithm was found to fail on edges very close to the boundary and at places where a convex and concave crease are very close to each other. The behavior is similar to that for the step edges, where instead of getting the maxima of the curvature, we have a zero crossing.. This can be solved by using the actual value of the curvature, instead of the absolute value for labeling of the *local curvature peaks*.

References

[1] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry : a hybrid approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 643–649, 1989.

[2] R. C. Bolles and P. Horaud, "3DPO : a three dimensional part orientation system," *International Journal of Robotics Res.*, vol. 5, no. 3, pp. 3–26, 1986.

[3] T. J. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-d objects using surface descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 11, pp. 1140–1157, 1989.

[4] D. J. Kriegman and J. Ponce, "On recognizing and positioning 3-d objects from image contours," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1127–1137, 1990.

[5] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

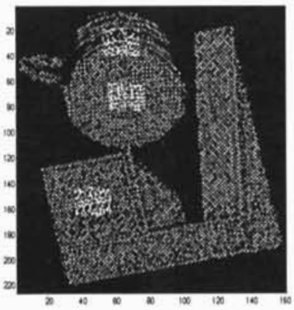

Figure 1: Local curvature peaks

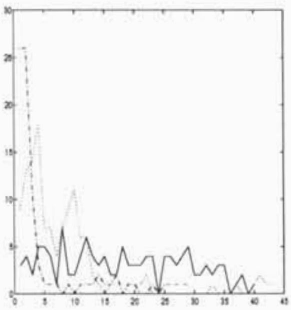

Figure 2: Histogram of the curvature values of the local curvature peaks ; the solid line corresponds to no edges; the dotted-dash line corresponds to edges with noise and the dotted line corresponds to isolated edges

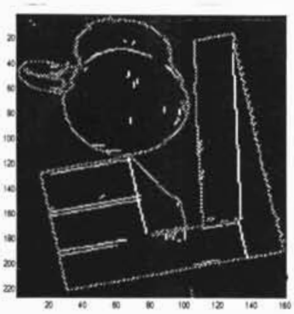

Figure 3: Crease edges at the end the labeling using the threshold and neighborhood criterion

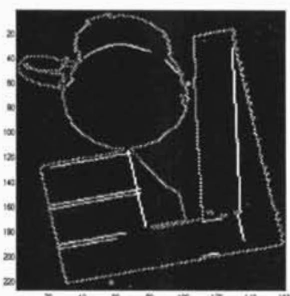

Figure 4: Crease edges after removing the stray pixels

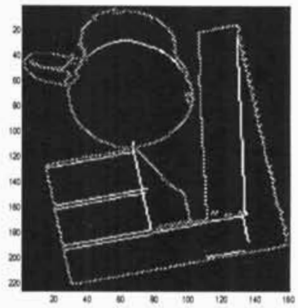

Figure 5: Crease edges for the block and cup image

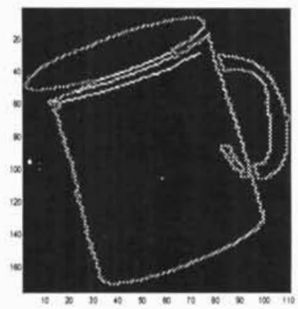

Figure 6: Crease edges for cup

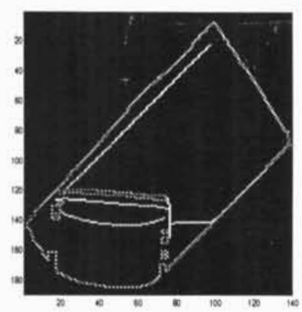

Figure 7: Crease edges for a hump and block

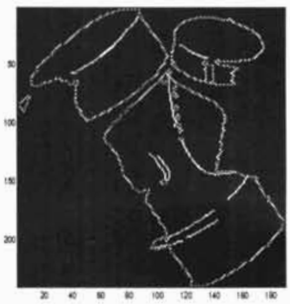

Figure 8: Crease edges for wye-joint