Robust Edge Detection in Range Images Based on Scan Line Approximation

Xiaoyi Jiang, Horst Bunke
Institut für Informatik und angewandte Mathemathik
Universität Bern, Switzerland

Abstract

In this paper we present a novel edge detection algorithm for range images based on a scan line approximation technique. Compared to the known methods in the literature, our algorithm has a number of advantages. It provides edge strength measures that have a straightforward geometric interpretation and supports a classification of edge points into several subtypes. We give a definition of optimal edge detectors and compare our algorithm to this theoretical model. By simulations we show that our algorithm has a near-optimal performance. We have carried out extensive tests using real range images acquired by three range scanners with quite different characteristics. The good results that were achieved demonstrate the robustness of our edge detection algorithm.

CR Categories and Subject Descriptors: I.4.6 [Segmentation]: edge and feature detection, pixel classification; I.4.8 [Scene Analysis]: range data.
General Terms: Algorithms.
Additional Key Words: edge detection, scan line approximation, optimality analysis.

1 Introduction

Many machine vision tasks extensively use range imagery to obtain reliable descriptions of 3-D scenes. Due to the large amount of data, direct interpretation of range images is extremely costly in terms of both storage and computation time. Thus, a segmentation step is usually carried out to group the range data into high-level features suitable for the subsequent image analysis and interpretation. In the range image domain, vision tasks such as object recognition [3, 11], model construction [15], configuration analysis [25], motion analysis [21, 30], automated visual inspection [26], and robotic grasping operations [1, 2] have been build in most cases upon scene representations of surface patches. This has led to the general agreement of defining the range image segmentation task as one of dividing range images into closed regions with application domain specific surface properties [16].

Range image segmentation algorithms can be broadly classified into two categories: edge-based and region-based segmentation. Region-based approaches group pixels into connected regions based on homogeneity measures, while boundaries between regions are located by edge detection methods. Both techniques have their strengths and drawbacks. Edge detection is mostly criticized for its tendency to produce non-connected boundaries. Extensive postprocessing may be needed to provide the final segmentation. Despite of the guarantee of closed regions, region-based techniques suffer from a number of problems. Usually, they have complex control structures. Also, the region boundaries are generally distorted. In addition, commonly used region-based techniques such as region-growing and clustering have several critical design issues to be dealt with. The performance of most region-growing approaches crucially depends on the selection of initial regions. In clustering-based methods it is difficult to adaptively determine the actual number of clusters in range images. Often, an oversegmentation is achieved and a subsequent merge step is needed to provide the final segmentation. As a matter of fact, a recent experimental comparison[1][16] reveals that even the seemingly simple task of segmenting range images into planar surface patches cannot be regarded as solved. There is still considerable room for improvement with respect to both segmentation quality and computation time.

Edge detection methods, on the other hand, possess simple control structures and regular operators such as image convolution, making them suitable for implementation on special-purpose image processors and parallel computers. Due to the nature of edge-based approaches, the region boundaries tend to be located precisely. The usefulness of edge detection is actually twofold. Edge detection has the potential of a complete segmentation. For this purpose heuristic criteria [2, 20] have been shown to be effective to close gaps in edge maps of range images. Alternatively, we can make use of edge detection to support region-based segmentation. An edge map may provide an initial segmentation that is further refined by region-based techniques. Edge information can also be incorporated into a region-based algorithm for a more reliable guidance of region extraction. For instance, edge information may be useful for seed region extraction. This point will be further discussed in Section 6.

In this paper we propose a simple and robust edge detection algorithm for range images. Our work was partially motivated by the desire to overcome the following drawbacks inherent to most of the algorithms known from the literature:

- no straightforward geometric interpretation of edge strength,

[1] This comparison is based on a number of objective performance metrics and two large range image sets acquired by a time-of-flight laser range finder and a structured light scanner, respectively, that have manually specified ground truth.

source	real images shown (real images evaluated)	scanners
Al-Hujazi & Sood [2]	4 (several)	1
Berkman & Caelli [4]	0 (0)	0
Ghosal & Mehrotra [12]	3 (several)	1
Ghosal & Mehrotra [13]	2 (2)	1
Günsel *et al.* [14]	0 (0)	0
Kaveti *et al.* [20]	6 (MSU)	1
Krishnapuram & Gupta [22]	7 (?)	2
Mintz [24]	2 (2)	1
Wani & Batchelor [34]	3 (3)	1

Table 1: Summary of recent journal-published edge detection algorithms for range images. "Real images shown" is counted from figures in the paper, while "real images evaluated" is drawn from the text. MSU means the popular image set acquired at State University of Michigan.

- no support of classification of edge points into detailed edge types,
- no comparison to an optimal (theoretical) edge detector,
- limited tests on real range images.

In order to develop a robust edge detector for range images, all these issues must be discussed. Usually, edge detection methods assign an edge strength value to each pixel. Then, an edge map is constructed by a thresholding operation. It is important that the edge strength has a straightforward geometric interpretation so that we can easily choose the threshold. Although the central task of edge detection is to reliably detect and locate edge points, a rich description of edge points including detailed edge types is highly desirable, too. Thus, the ability of an edge detection method to support the classification into various edge types is of importance. The optimality of edge detectors has been rarely considered so far in the literature. As an essential means of performance characterization, however, this issue should be definitely investigated. Finally, not only theoretical performance characterization but also experimental evaluation is essential to demonstrate the robustness of an edge detection algorithm. Today, a large number of range scanners with different characteristics (working principle, sensor geometry, noise, etc.) are available [5, 18]. Therefore, the ease of adaptation to range images acquired by different range sensors should belong to the efforts to characterize the performance of an edge detection method. For example, some algorithms are limited to range images sampled on a regular grid. Most algorithms known from the literature have been tested only on a small number of images taken by a single range camera, as illustrated in Table 1. Note that experiments using synthetic images are not included. In our opinion, work that stops short of using real images inspires little confidence in its relevance. From Table 1 we can see that only one paper, namely [22], has reported experiments using two different range cameras.

The rest of this paper is organized as follows. We begin with the formulation of the edge detection problem. Then, we give a brief review of known algorithms in the literature.

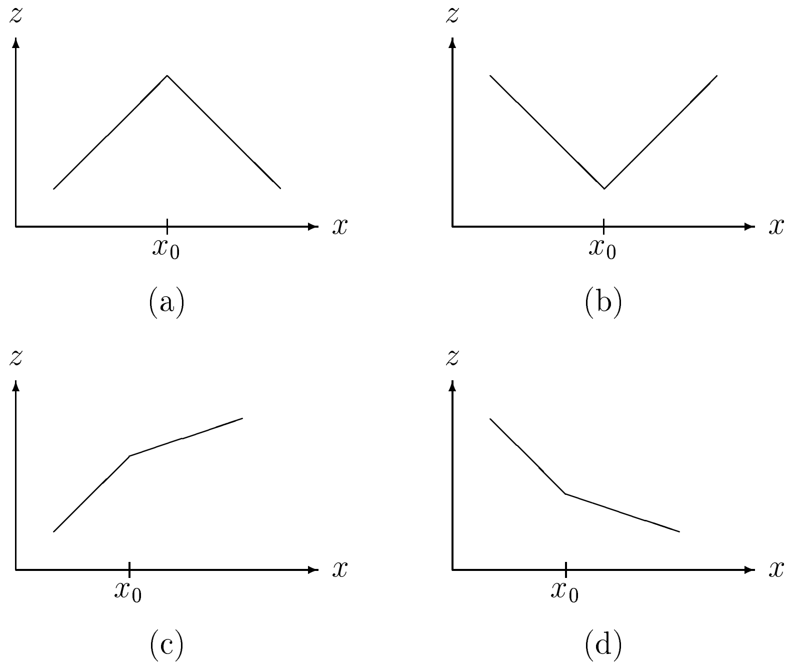


Figure 1: Subtypes of crease edges: (a) convex roof edge, (b) concave roof edge, (c) convex non-roof edge, (d) concave non-roof edge.

Section 4 is devoted to our new algorithm, followed by an optimality analysis in Section 5. In Section 6 we show experimental results, applying our algorithm to images that come from three different range scanners. Finally, a discussion concludes the paper.

2 Edge detection problem in range images

In range images we can distinguish between three basic types of edges. *Jump edges* are usually defined as discontinuities in depth values. Such edges occur when an object is occluded by another object or itself. *Crease edges* are formed where two surfaces meet. Such edges are characterized by discontinuities in surface normals. Finally, *smooth edges* are those with continuous surface normals but discontinuous curvatures. Precise computation of curvatures is extremely difficult in range images due to noise, making the detection of smooth edges almost impossible. Because of this problem and the fact that smooth edges relatively seldom occur in range images, their detection has been widely ignored in the literature. In this paper we concentrate our attention to jump and crease edges, too.

Crease edges can be further divided into *roof* and *non-roof edges* [22]. Roof edges correspond to local extrema and have either higher or lower depth values on both sides, while non-roof edges are characterized by discontinuities in surface normals with lower values on one and higher values on the other side. Since convexity provides another edge classification scheme, we have totally four possible subtypes of crease edges (see Figure 1). Note that the distinction between roof and non-roof edges is dependent on the viewpoint and thus of little practical usefulness. The convexity, on the other hand, is a view-invariant feature and represents an intrinsic property of objects. In this work we will only make the distinction

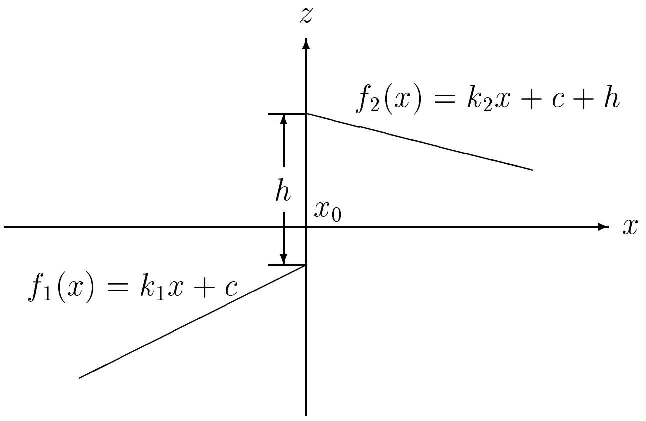


Figure 2: 1-D edge model.

between convex and concave, but not between roof and non-roof edges.

Careful examination of range images reveals that the usual definition of jump edges as discontinuities in depth values is not always adequate. At a constant sampling density for the whole scene, two adjacent pixels on a highly sloped surface may have quite different depth values. This is particularly true for some of the range images used in our experiments. Therefore, a simple thresholding of discontinuities does not work well for jump edge detection. For this reason we introduce the following definition of edges. Consider the 1-D edge model shown in Figure 2. If we model each side of a pixel locally by a straight line, then the pixel is considered as an edge pixel if the two lines $f_1(x)$ and $f_2(x)$ are different. In this case we have a jump edge if there exists a difference between the function values $f_1(x)$ and $f_2(x)$ at position x_0 (i.e., $h = |f_1(x_0) - f_2(x_0)| \neq 0$), and a crease edge otherwise. The edge strength can be defined as h for a jump edge, and the angle between the normals of $f_1(x)$ and $f_2(x)$ at x_0 for a crease edge, respectively. This definition properly excludes pixels on a highly sloped surface from being recognized as jump edges. The 1-D edge model introduced above can be easily extended to a 2-D one if we model the local environment of an edge point (x_0, y_0) by two planes $f_1(x, y) = a_1x + b_1y + c_1$ and $f_2(x, y) = a_2x + b_2y + c_2$. Then, the edge strength is simply

$$|f_1(x_0, y_0) - f_2(x_0, y_0)|$$

for a jump edge and

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

for a crease edge. Notice that although both the 1-D and 2-D edge model use planar surface patches to model the two sides of an edge point, they are useful for curved surfaces as well. The reason is that a small local environment can always be reasonably well approximated by a planar surface patch.

Generally, the detection of crease edges is much more difficult than that of jump edges. The crease edge strength in (1) is independent of the position and orientation of the scene relative to the range scanner. Also, it is invariant to changes of the coordinate system. Therefore, the term (1) can be regarded as an ideal quantitative characterization of the strength of a crease edge. Accordingly, we consider an edge detector that supplies this edge strength as *optimal* [2]. For any edge detection method, the deviation from this ideal edge strength provides thus an effective way of performance evaluation.

3 Related work

A variety of methods are available for detecting edges in range images. Most directly, step edge detection operators developed for the intensity image domain can be used to detect jump edges. For crease edges we can apply the same edge detector independently to the three components of the normals of the imaged surfaces and then combine the results to locate edges, say by taking the maximum of the three channels. The works [7, 19] belong to this category of edge detection methods and are based on a morphological edge detector and the Canny operator, respectively.

Another approach to edge detection is residual analysis. Al-Hujazi and Sood [2] considered the absolute difference (residue) between the input image and its smoothed version, which possesses maxima at the locations of jump and crease edges. Edge detection is done by locating such maxima. Using the edge model in Figure 2, it has been shown that the residue for crease edges is proportional to $|k_1 - k_2|$. This leads to a non-uniform edge strength, i.e., two intersecting planes will produce different edge strengths dependent on their orientation relative to the scanner. It is also possible that two pairs of intersecting planes with different angles of intersection will result in the same edge strength. This non-uniformity is certainly an undesired property.

Mathematical morphology is attractive due to the fact that it involves simple logical operations and can be implemented in parallel, thus making real-time application possible. An application of mathematical morphology to edge detection in range images has been described in [7]. Cheng and Don [8] proposed another morphological approach to detect convex roof edges only. Such edges are found by looking for the leaf nodes of the skeletal tree of a range image constructed by morphological skeleton operations. Krishnapuram and Gupta [22] have developed two other morphological methods. Essentially, they are a morphological implementation of residual analysis techniques and the first-derivative operator, respectively. The results of morphological operations are used to classify pixels into non-edges and edges of several types by rules. However, these two methods don't provide a quantitative characterization of edge strengths.

Multiscale boundary detection has proven to be effective for dealing with discontinuities occurring at a variety of spatial scales. Günsel *et al.* [14] followed this approach by considering the boundary detection process as a fusion of n different sensory processing modules, each corresponding to a specific scale. The output of each module was modeled to be dependent on all other outputs by being part of a joint a *posteriori* probability distribution. Then, the boundary detection was done by maximizing this probability function using the Bayesian approach.

In [12, 13] operators based on orthogonal Zernike moments were used to recover the parameters of a general 2-D edge model at each pixel. Similar to [2], this method provides an edge strength that is proportional to $|k_1 - k_2|$ and thus suffers from the same non-uniformity problem.

[2] Any edge strength definition that is a monotonic function of (1) corresponds to an optimal edge detector, too.

Berkman and Caelli [4] explored the application of covariance techniques to surface representation of 3-D objects. It was shown that the covariance approach provides shape descriptors invariant to rigid motions via the eigenvalues of covariance matrices of different orders, without explicitly using surface parameterizations or derivatives as in the case of curvatures. These eigenvalues were directly geometrically interpreted and thresholded to detect jump and crease edges.

In [24] Mintz made use of robust estimators to transform local image windows into binary (inlier/outlier) windows. In case of discontinuities, the binary window resembles an inlier/outlier step edge and its location corresponds to the location of the discontinuity in the original image. For more robustness the final decision of edge/non-edge was made by a consensus of the votes for a pixel resulting from different image windows.

4 Edge detection based on scan line approximation

The edge detection method proposed in this paper is based on the approximation of a scan line by a set of polynomial functions. First, we apply the approximation separately to the rows, columns and the two diagonal directions in the image. In the second phase, the results of the four processes are combined.

4.1 Scan line approximation technique

For description clarity we first assume a dense range image $z(x, y)$ regularly sampled in both coordinate directions. Let $f(x, y, z) = 0$ be the 3-D surface of an object to be segmented. Then, an image row with a constant y_0 is simply a two-dimensional curve $f(x, y_0, z) = 0$ in the $x - z$ plane. A planar 3-D surface $ax + by + cz + d = 0$, for example, results in a straight line $ax + cz + e = 0,\ e = by_0 + d$, on the image row. Generally, curve segments $f(x, y_0, z) = 0$ 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 boundaries between two surfaces and are therefore edge points. The same idea applies to image columns and the two diagonals as well.

To implement the scan line approximation technique, we have to choose an appropriate surface function $f(x, y, z) = 0$. One possibility is the implicit quadratic surface function:

$$f(x, y, z) \ = \sum_{i+j+k \leq 2} a_{ijk} x^i y^j z^k \ = \ 0 \tag{2}$$

that covers common surface types such as spheres, cylinders and cones. In this case the curve segments in the scan lines have the form:

$$f(x, z) \ = \sum_{i+j \leq 2} a_{ij} x^i y^j \ = \ 0.$$

For the partitioning of scan lines into curve segments, however, this representation suffers from a high handling complexity. So, we turn to bivariate polynomials:

$$z \ = \ f(x, y) \ = \sum_{i+j \leq k} a_{ij} x^i y^j.$$

To make the scan line approximation as simple as possible, we have chosen $k = 2$. In [6] it has been shown that the implicit surface function (2) can only be approximated well by a

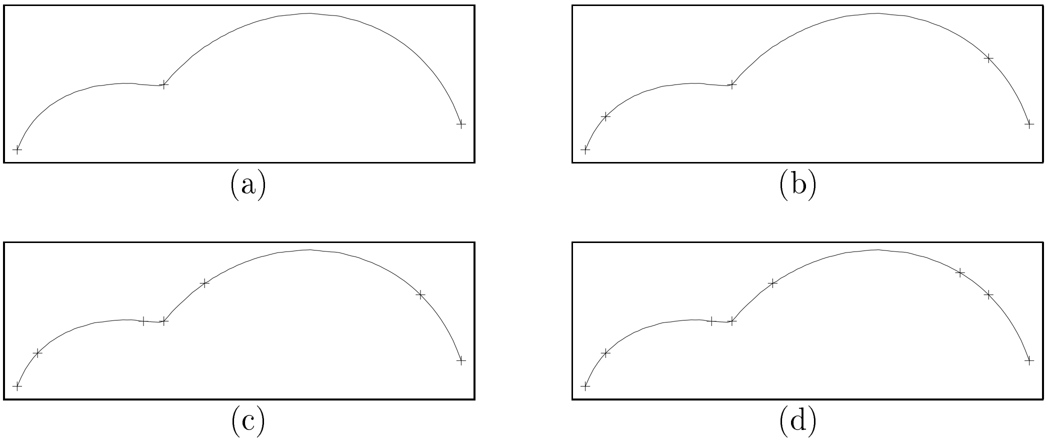


Figure 3: Splitting process of a scan line.

bivariate polynomial using $k = 4$, implying that our use of quadratic bivariate polynomials may produce some oversegmentation of scan lines. I.e., more partitioning points than the true edge points will be generated since a continuous part of a scan line may be divided into more than one curve segment. Fortunately, the edge candidates resulting from such an oversegmentation usually have a very small edge strength so that they can be easily filtered out by a thresholding operation (see the discussion in Section 4.2).

For quadratic bivariate polynomials we need to partition scan lines into curve segments of the form:

$$z \ = ax^2 + bx + c.$$

We have used the classical splitting algorithm described by Duda and Hart [9]. A quadratic approximation function is first determined for a whole scan line based on the midpoint and the two endpoints. Then, whenever the largest error $e_{\max}$ between the approximation function and the scan line is greater than a preselected threshold ϵ, the scan line 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.

It is well known that the simple splitting method of Duda and Hart produces sometimes spurious segments. For the problem of piecewise linear approximation of curves, Pavlidis and Horowitz [27] tried to solve this problem by introducing a merge step. As stated in [10], however, this algorithm is computationally rather expensive. For complex curves, it produces sometimes worse results than the simple splitting method. For the partitioning into quadratic curve segments, the usefulness of a merge operation is even more questionable. Therefore, we use the splitting algorithm without any merging and potentially generate an oversegmentation. Again, this kind of oversegmentation will not result in any serious problem because of small edge strength (see Section 4.2).

To illustrate the scan line approximation technique, Figure 3 shows a scan line from a real range image containing two cylindrical surfaces and the results of different partitioning levels. The whole scan line is first divided into two parts (Figure 3(a)). Then, a further division of each part leads to totally four curve segments (Figure 3(b)). Two more splitting operations on the left segment of the right cylinder, and one splitting operation on the right segment of the left cylinder lead to a total of seven curve segments, which is the final result (Figure 3(c) and (d)). Eventually, there are six edge candidates, corresponding to the interior partitioning points. Besides the expected one on the boundary between the two surfaces,

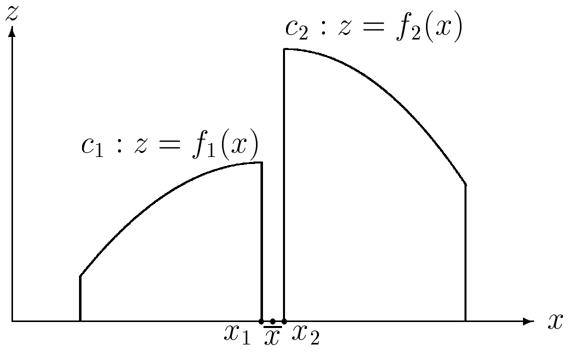


Figure 4: Definition of discontinuity strength at the boundary of two curve segments.

the splitting algorithm also generates five other edge candidates on smooth curves due to the reasons discussed above. In Section 4.2 it will be shown that all these spurious edge candidates produce negligible edge strength values so that they can be easily excluded from further consideration by a thresholding operation.

In summary, the scan line approximation technique considers each scan line as a two-dimensional curve and splits it into quadratic curve segments. For description ease, we have assumed a regularly sampled range image in the discussion above. But it is easy to see that the only condition for applying this technique is that all image rows, columns and diagonals result from the intersection of a 3-D plane with the objects' surfaces and each scan line thus corresponds to a curve in a plane. Notice that this condition is satisfied by a much wider range of scanners than those of regular sampling in both coordinate directions, thus substantially extending the applicability of our edge detection algorithm. For instance, structured light scanners like that described in [31] typically don't have the property of regular sampling. However, the measured points of a scan line in 3D space lie in a plane spanned by the scan line in the image plane and the focal point of the camera. Therefore, our algorithm is certainly applicable to this type of range images. Actually, some of the tests described in Section 6 are based on this type of range images.

4.2 Edge detection and classification

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 4) 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. According to our discussion in Section 2, the difference of depth values of adjacent pixels, i.e., $|f_1(x_1) - f_2(x_2)|$, is not an adequate characterization of jump edge strengths. 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

$$|(z_1 + f_1'(x_1)(\overline{x} - x_1)) - (z_2 - f_2'(x_2)(x_2 - \overline{x}))|.$$

edge candidate	1	2	3	4	5	6
jump	0.006	0.005	0.018	0.008	0.001	0.005
crease	4.0°	1.6°	36.3°	4.0°	4.0°	2.8°

Table 2: Edge strength values for the scan line in Figure 3(d).

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 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 determination of the edge type becomes now very simple. An edge candidate is of the type convex if

$$-f'_1(\overline{x}) < -f'_2(\overline{x}), \quad \text{or} \quad f'_1(\overline{x}) > f'_2(\overline{x});$$

otherwise, it is a concave edge. Alternatively, the test

$$\text{convex: } f'_1(x_1) > f'_2(x_2), \quad \text{concave: } f'_1(x_1) < f'_2(x_2)$$

may be used, too.

The scan line partition is controlled by the threshold ϵ. It should be set small enough so that we will not miss any edge point. This way we potentially generate an oversegmentation. Actually, an oversegmentation is also caused by our use of quadratic bivariate polynomials as surface representations. However, this kind of oversegmentation doesn't result in any serious problem since it can be expected that both jump and crease discontinuity strength values defined above will have very small values compared to true edge points.

As an example to illustrate the edge strength measures, we consider the scan line and the partitioning shown in Figure 3(d). The edge strength values for the six edge candidates are listed in Table 2, where the edge candidates are numbered from left to right on the scan line, and for each edge type the second of the two possible edge strength measures defined above has been used. Except for the expected edge point on the boundary between the two cylindrical surfaces, all candidates have negligibly small crease edge strengths. As a reference for the significance of jump edge strengths we take the difference of z-values between two adjacent pixels in a range image. For the range image containing the scan line in Figure 3(d) this difference value has an average of 0.009 and standard deviation of 0.033. Obviously, all the six edge candidates have very small jump edge strength, excluding them from being considered as jump edge candidates.

Dependent on the configuration of surfaces in the scene, the maximal value of the discontinuity measures may not be observed in one particular direction. To capture the information

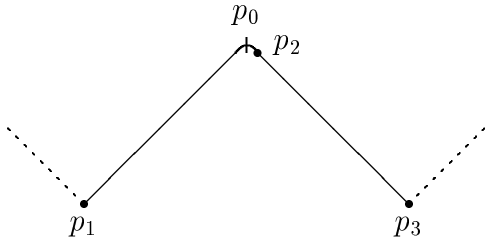


Figure 5: The edge position as given by the splitting algorithm is not always precise.

available in the scene to a larger extent we carry out the scan line approximation and the subsequent computation of edge strengths in four directions: image rows, image columns and the two diagonal (45° and 135°) directions. A pixel thus has four different discontinuity measures of each type. We combine the four measures by simply taking the maximum. The further edge type classification into convex or concave is determined by the direction corresponding to the maximal crease edge strengths.

In our experiments it turned out that for noisy range images the position of edge points as determined by the simple splitting algorithm is not very precise. The reason is that no further adjustment of the partitioning points is performed. So, partitions like that in Figure 5 may occur. Here part of a scan line, the curve p_1p_3, has been divided into two segments p_1p_2 and p_2p_3. But the real edge position p_0 has not been optimally recovered. To achieve better edge localization 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. Furthermore, recall that the functions $z = f_{1,2}(x)$ provided by the splitting algorithm are computed by the midpoint and the two endpoints of the curve segments. To capture the information of the curve segments to a larger extent, we can alternatively compute another function $z = \overline{f}_{1,2}(x)$ by means of the least square method using all pixels of each curve segment and forward these functions to the subpixel edge localization process. As a matter of fact, this latter approach gives us the best results. The results reported in Section 6 are based on this edge position adaptation method.

5 Optimality analysis

In Section 2 we have defined an optimal edge detector to be one that provides the angle between the two intersecting surfaces at an edge point as discontinuity measure. Our 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, i.e., real value. We are interested in the amount of this deviation and thus the question to which extent our edge detector is optimal. In the following it is shown by a quantitative performance analysis that our edge detection algorithm is near-optimal.

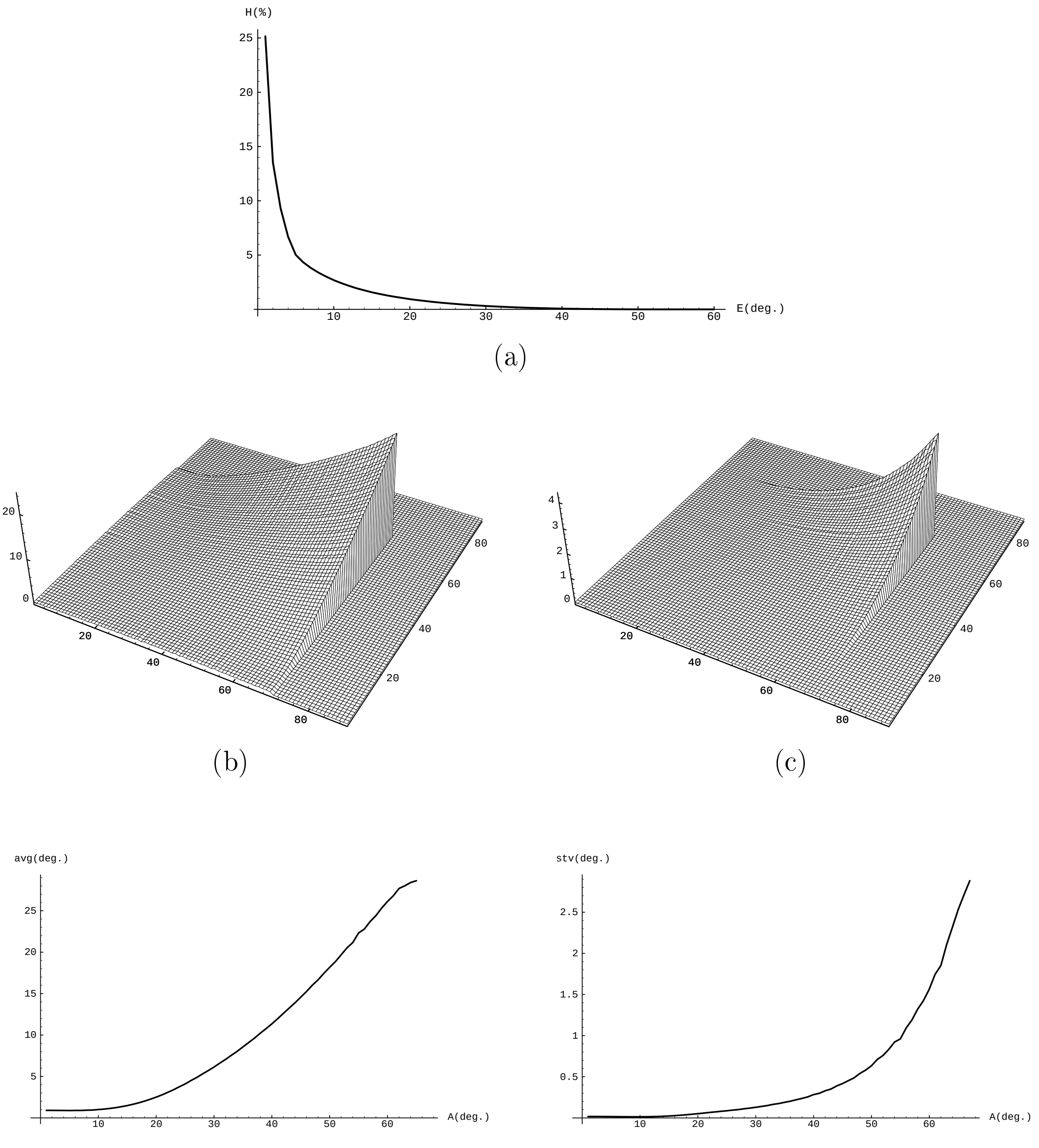
For the purpose of optimality analysis simulation tests were carried out. We investigated all possible configurations of two intersecting planes at an edge point. Both intersecting planes can take an arbitrary orientation. For practical reasons the plane orientation is limited such that its slant angle (with the z-axis) is smaller than 70^o. Other orientations are rarely observed in range images. We consider all combinations of two plane orientations and compute for each such combination the difference between the crease edge strength determined by our edge detection algorithm and the optimal value. We call this difference the edge strength estimation error (shortly estimation error).

We represent the orientation of the two intersecting planes by their unit surface normals (a_1, b_1, c_1) and (a_2, b_2, c_2), where each surface normal is given by a point on the unit sphere. For this representation we need a uniform tessellation of the unit sphere so that only a finite number of surface normals have to be dealt with. In our simulations we apply a tessellation method based on the well-known geodesic dome constructions [17]. Starting with a regular icosahedron, each of its edges is divided into f equal sections, where f is called the frequency of the geodesic division. This results in f^2 triangles for each face and totally $20f^2$ triangles. Then, this divided icosahedron is projected onto the unit sphere and the centers of all cells define an approximate uniform tessellation of the unit sphere. In our simulations a geodesic division of frequency 32 has been used.

Since we use unit surface normals, the optimal edge strength defined in (1) becomes now

$$\cos^{-1}[(a_1, b_1, c_1) \cdot (a_2, b_2, c_2)].$$

In our simulation the estimation error E of the crease edge strength determined by the algorithm is computed for all combinations of two orientations $((a_1, b_1, c_1), (a_2, b_2, c_2))$, each corresponding to a cell of the unit sphere tessellation whose slant angle is smaller than 70^o. The estimation error has an average of 6.4^o and a standard deviation of 7.7^o. Figure 5(a) shows the histogram for E. Apparently, the edge strength computed by the proposed algorithm is very close to the optimal value. Now we give a more detailed characterization of the estimation error E. First, we investigate the dependence of E on the slant angle of the two intersecting planes. Figure 5(b) and (c) plot the average and standard deviation (both in degrees) of E, respectively, as a function of a pair of slant angles (α_1, α_2), $0^o \leq \alpha_1, \alpha_2 \leq 70^o$. Obviously, E increases with the slant angles. The larger estimation errors occur in the case of large values for both slant angles that is rather rarely observed in range images. This means that in most cases our edge detection algorithm is able to provide a precise edge strength estimation. There is a second way to characterize the estimation error E, which is not entirely dependent on the slant angels of the two intersecting planes. For instance, two planes $z = \pm sx$ have very large slant angles for large values of slope s. But the optimal edge strength $2\tan^{-1} s$ will be captured by our algorithm through the analysis of image rows. Actually, the optimal edge strength is reached at the directional section that contains the surface normals of both intersecting planes. In the example above this directional section coincides perfectly with image rows. As the angle A of the optimal directional section with the z-axis becomes larger, the four directional sections used in our algorithm will move away from the optimal one and the estimation error E will increase. Therefore, we can also look at the dependence of the estimation error on the angle A. Figure 5(d) and (e) plot the average and standard deviation (both in degrees) of E, respectively, as a function of A, confirming the good edge strength estimation performance of the algorithm. Based on the simulation results described above we conclude that our edge detection algorithm is near-optimal.



(a)

(b)

(c)

(d)

(e)

Figure 6: Optimality analysis: (a) histogram of the estimation error; average (b) and standard deviation (c) of the estimation error as a function of (α_1, α_2); average (d) and standard deviation (e) of the estimation error as a function of the angle A.

6 Experimental results

The proposed edge detection algorithm has been implemented in C on a Sun Sparcstation 5. For tests we have used a large number (about 280) of real range images acquired by three range scanners with quite different characteristics:

- a Technical Arts scanner [32] operating on the triangulation principal using laser plane projection,
- an ABW structured light scanner [31], and
- a Perceptron time-of-flight laser scanner [28].

These three types of range scanners are among the most important active ranging methods [5, 18] and thus represent quite well the spectrum of the range scanners available today. The first image source ist the popular image set from the PRIP Lab of Michigan State University and another 38 registered range/intensity image pairs[3] from the same Lab [23]. The second image set acquired by the ABW scanner consists of about 50 images of both polyhedral and curved objects[4]. By contrast, all 40 images taken by the Perceptron scanner[5] contain only polyhedral objects[6]. The three image sets have quite different characteristics. The images from Michigan are approximately regularly sampled in both coordinate directions, while this property is given neither for ABW nor for Perceptron images. Generally, the Technical Arts scanner provides the best range images in terms of noise, followed by the ABW and Perceptron scanner. The three image sets have also different imaging volumes. The Perceptron's is room size, whereas both the Technical Arts and the ABW scanner have an image volume of table-top size.

Our edge detection algorithm has a single threshold, namely ϵ, for the control of the scan line approximation process. For all test images of each scanner, we have applied the same value for ϵ. The choice of this value is not critical. For a range of values, we have observed similar edge detection results.

To visualize the output of our algorithm we map the edge strength map $e(i,j)$ to the range [0..255] of greylevels by

$$\begin{cases} \lceil \frac{e(i,j)}{e_m} \cdot 255 \rceil, & e(i,j) \leq e_m \\ 255, & e > e_m \end{cases}$$

where e_m is a cut value specified for each edge type. In the case of crease edges, for instance, it has been chosen to be 64^o for all three image sets. To reduce the space needed for the visualization, the edge strength maps of both edge types are merged to get an overall edge strength map by simply taking the maximum. We use a threshold to construct a binary edge map. In the experimental results described below, this threshold is set to 80 for the Michigan

[3] Both are available from http://www.eecs.wsu.edu/IRL/RID/RID.html.

[4] Available from http://iamwww.unibe.ch/~fkiwww/ResearchAreas/RangeImages.html. The images of polyhedral objects only can also be downloaded from http://marathon.csee.usf.edu/range/seg-comp/SegComp.html.

[5] Available from http://marathon.csee.usf.edu/range/seg-comp/SegComp.html.

[6] Forty images of the ABW image set and the entire Perceptron image set constitute the test data in a recent experimental comparison of region-based range image segmentation algorithms [16].

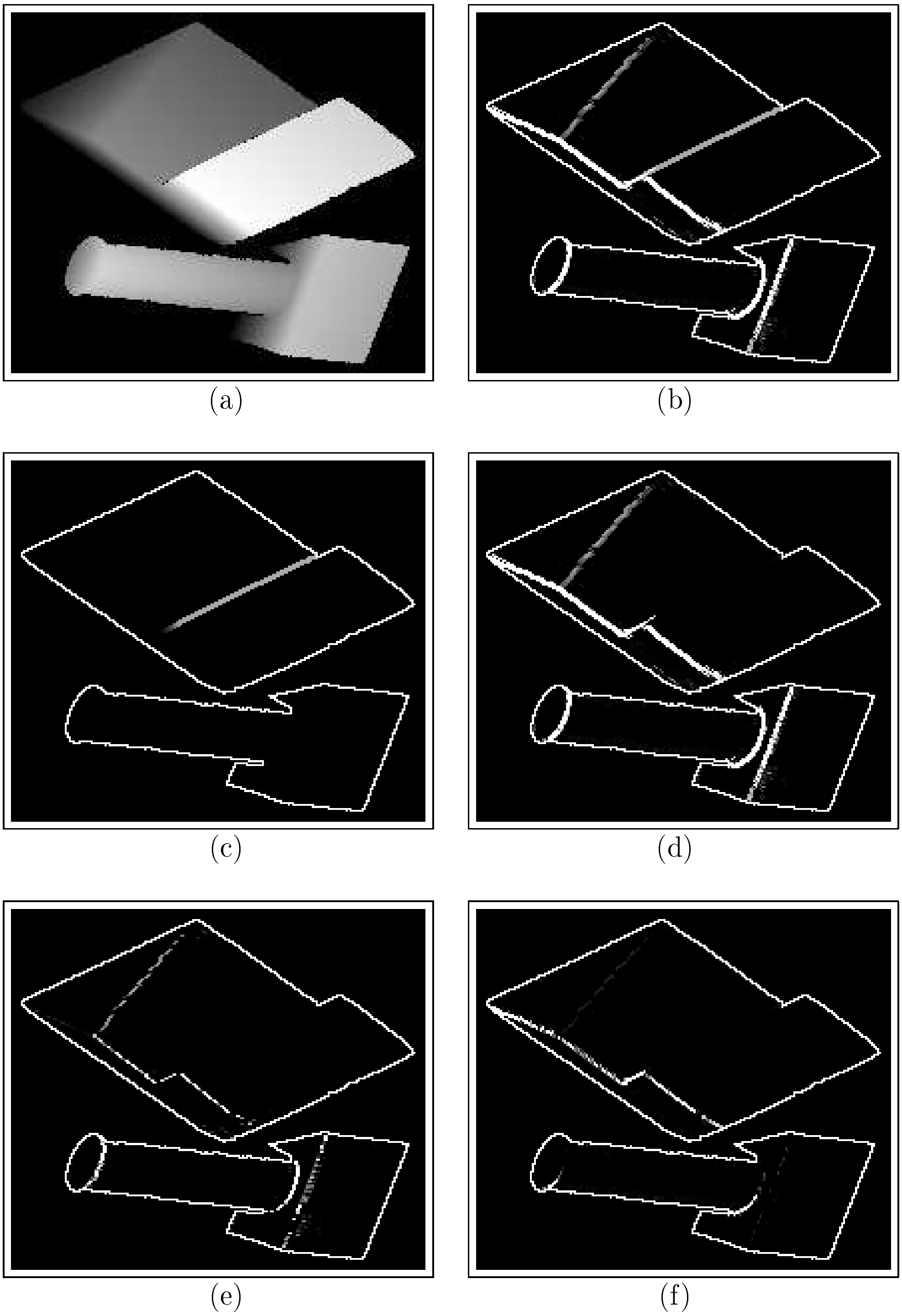
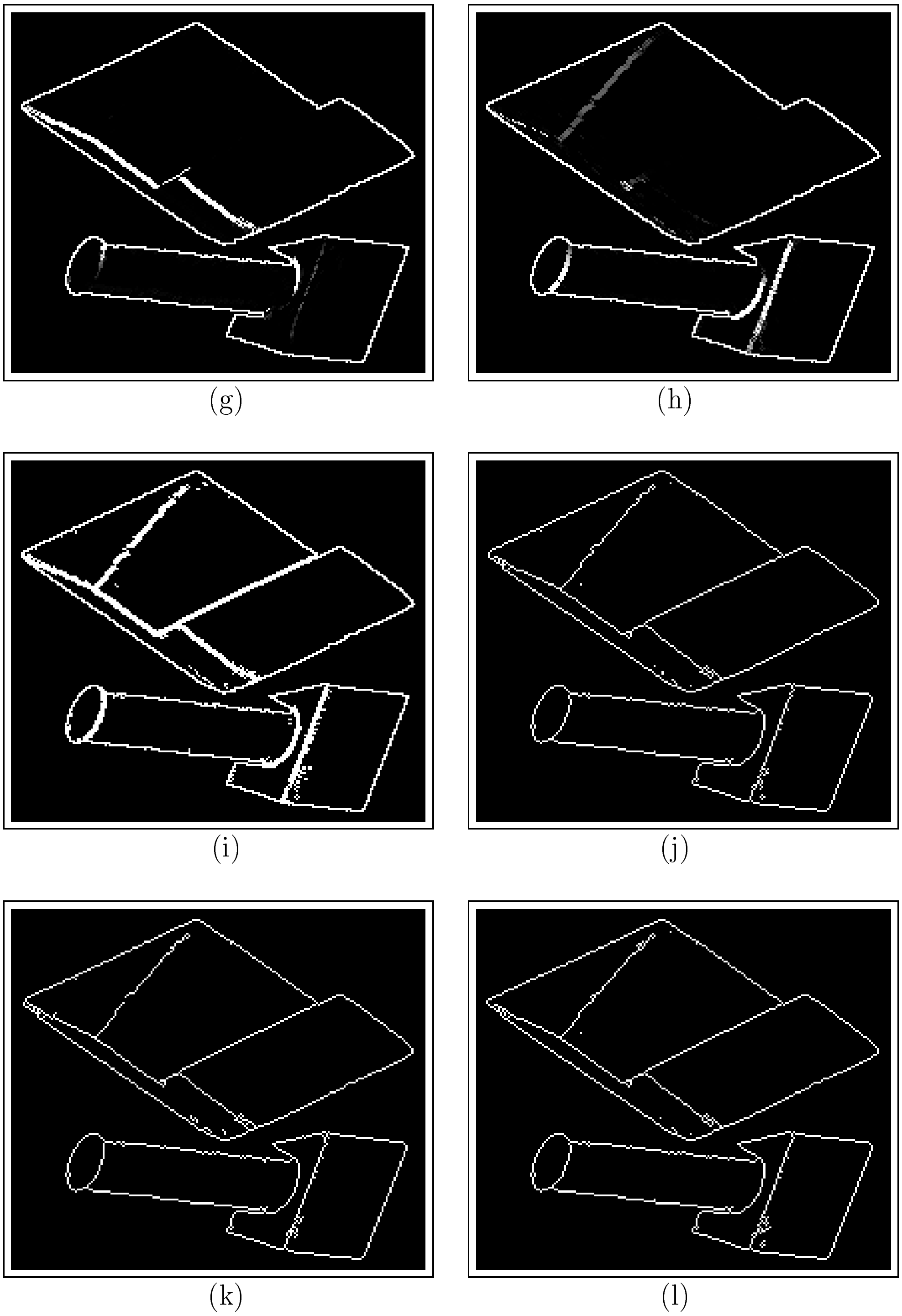


Figure 7: A range image from the Michigan image set and its edge detection results (see text).

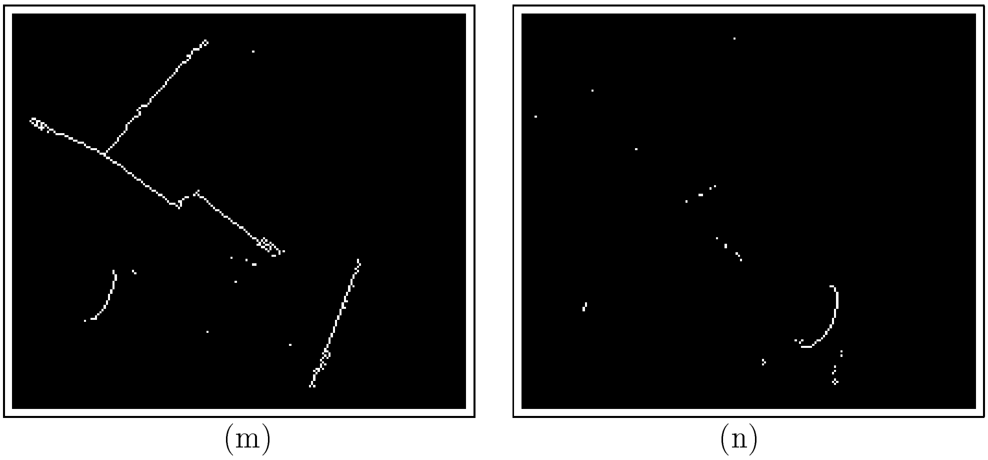


(g) (h)

(i) (j)

(k) (l)

Figure 7: (Cont.) A range image from the Michigan image set and its edge detection results (see text).



(m) (n)

Figure 7: (Cont.) A range image from the Michigan image set and its edge detection results (see text).

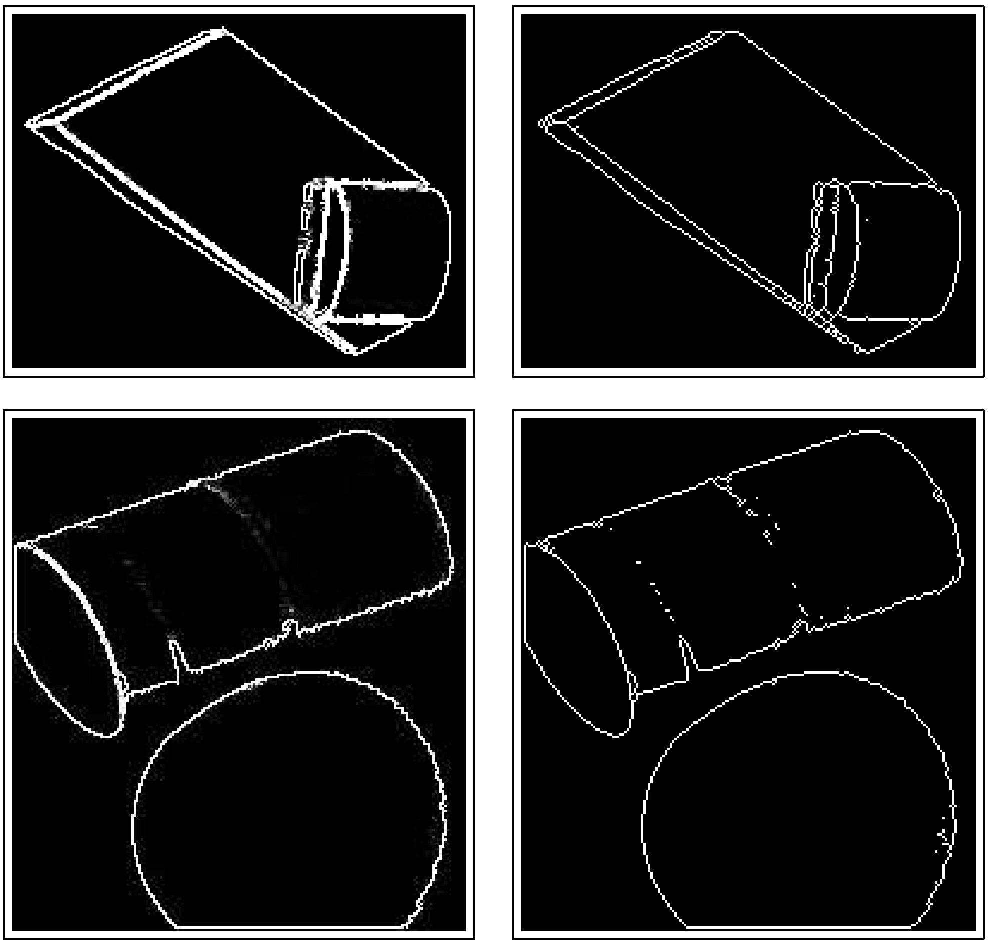


Figure 8: Results on six other Michigan images.

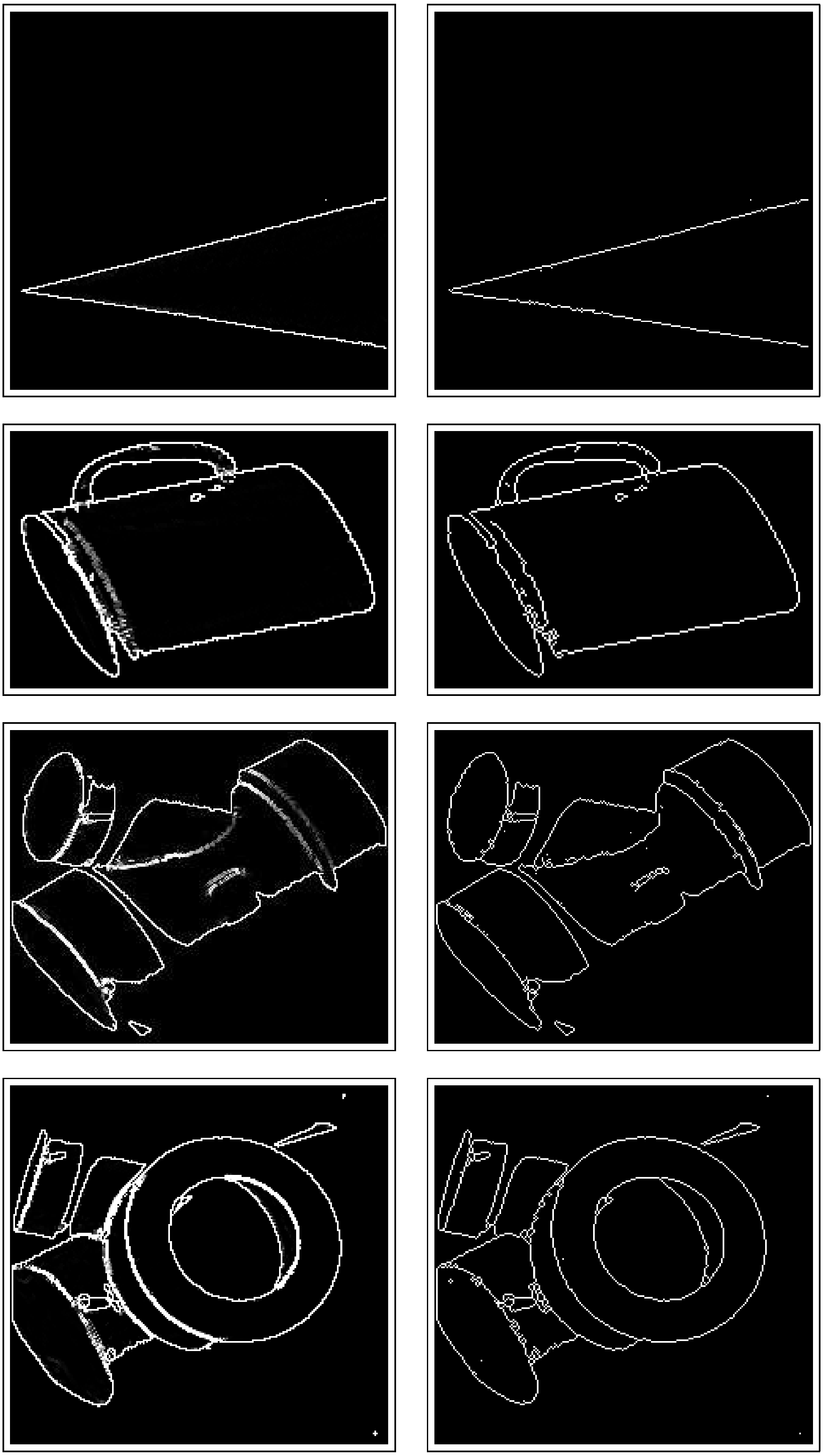


Figure 8: (Cont.) Results on six other Michigan images.

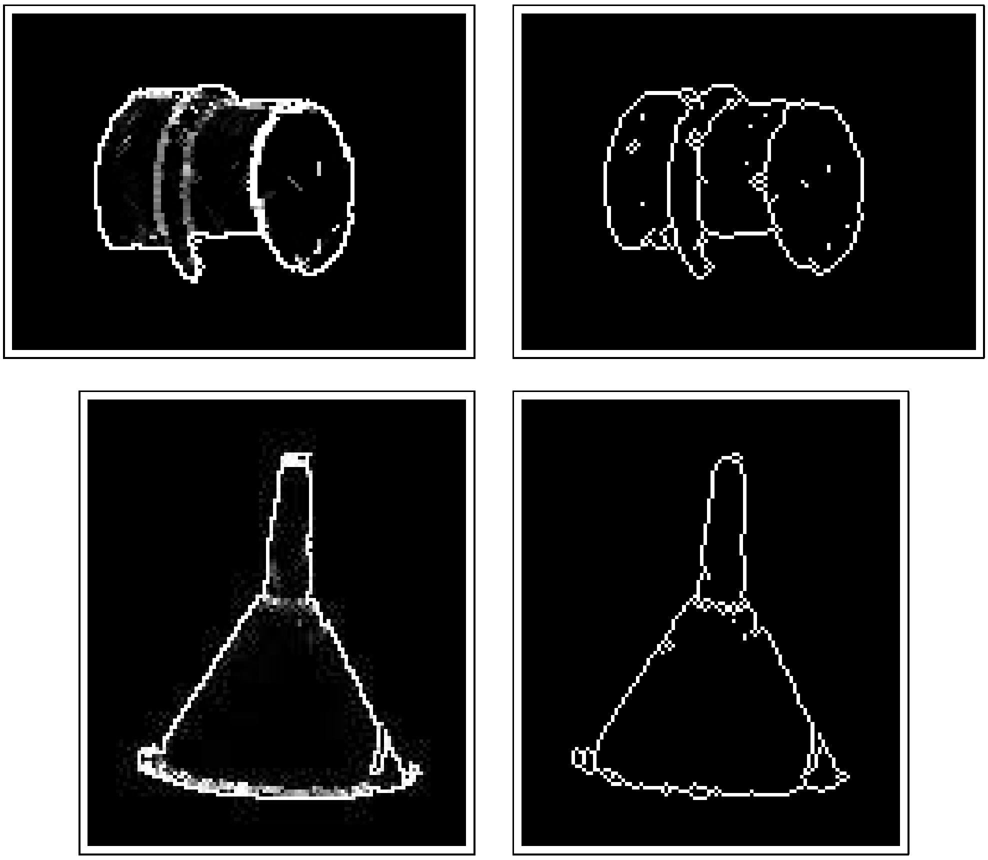
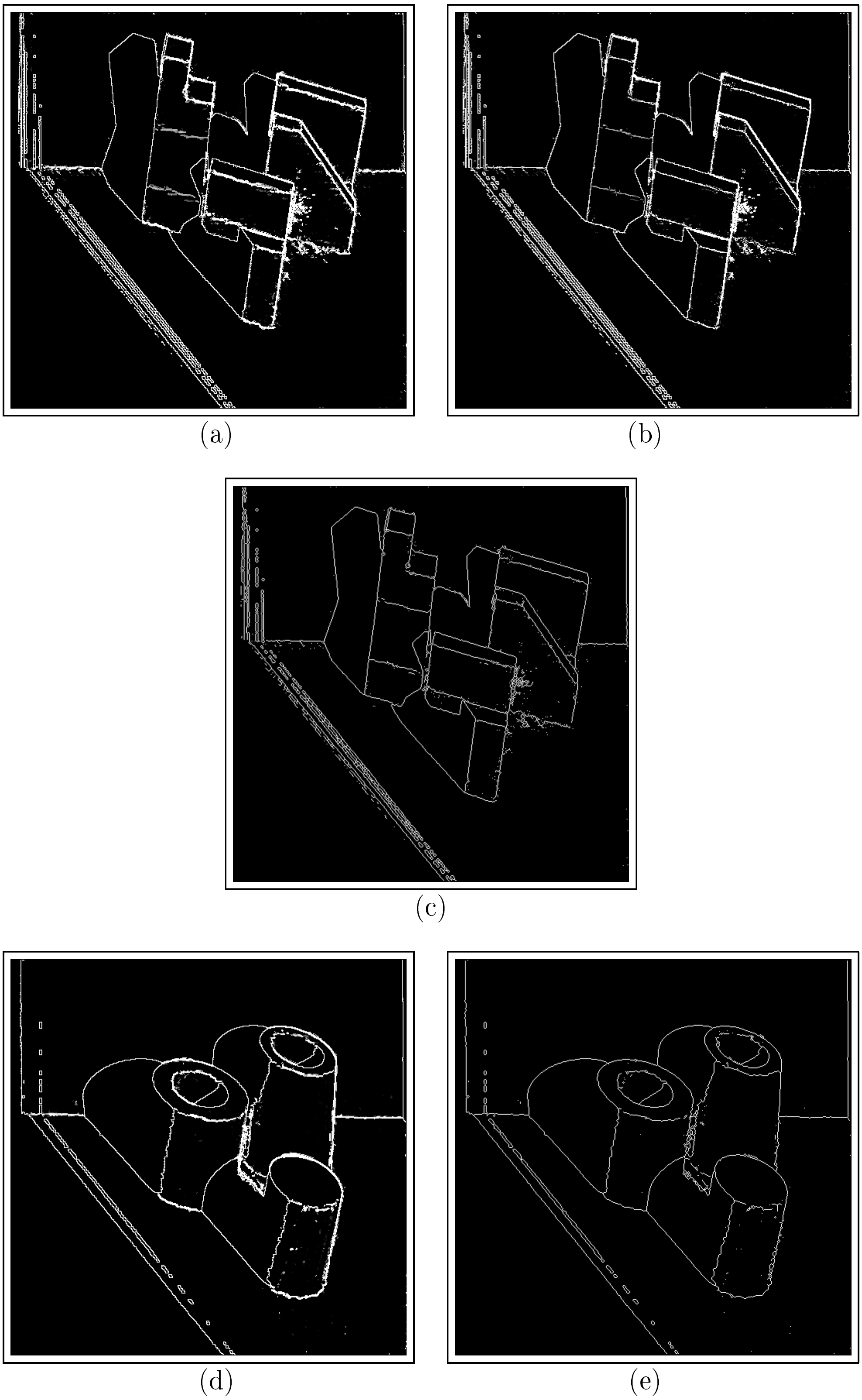


Figure 9: Results on two images described in [23].

images and 120 for the other two image sets, corresponding to 20^o and 30^o, respectively, in the case of crease edges. Note that more elaborate thresholding methods are known in the literature. The approach in [33], for instance, takes both edge strength and edge curve length into account and is very effective to delete isolated spurious edge points.

We use the range image in Figure 7(a) acquired by the Technical Arts scanner to illustrate the behavior of our edge detection algorithm. The edge strength maps for jump and crease edges are shown in Figure 7(c) and (d), respectively, and the resulting overall edge strength map in Figure 7(b). Notice that the two parallel surfaces of the top object caused high jump edge strengths but no crease edges. Figure 7(e)-(h) represent the crease edge strength maps created by the analysis of image rows, column and the two diagonals, respectively. Obviously, each process responds strongly to edges of a particular direction, and a combination of all four processes is necessary to capture edge information of arbitrary surface configurations. The binary edge map and its thinned version are in Figure 7(i) and (j), respectively. For this scene our algorithm has successfully extracted edge points in both planar and curved regions. There are only a few short gaps that can be easily closed to form a complete segmentation. For the entire Michigan image set the threshold ϵ has been set to 0.02. Other choices around this value lead to similar edge detection results. For the scene in Figure 7(a), the use of $\epsilon = 0.015$ and 0.025 produces the binary edge maps in Figure 7(k) and (l). There is no essential difference to the edge map in Figure 7(j). The edge points in Figure 7(j) are classified as convex or concave as shown in Figure 7(m) and (n). This example demonstrates the ability of our algorithm to reliably characterize edge types. The performance of the algorithm is further illustrated by six other scenes shown in Figure 8, where for each scene



(a)

(b)

(c)

(d)

(e)

Figure 10: Results of two ABW images (see text).

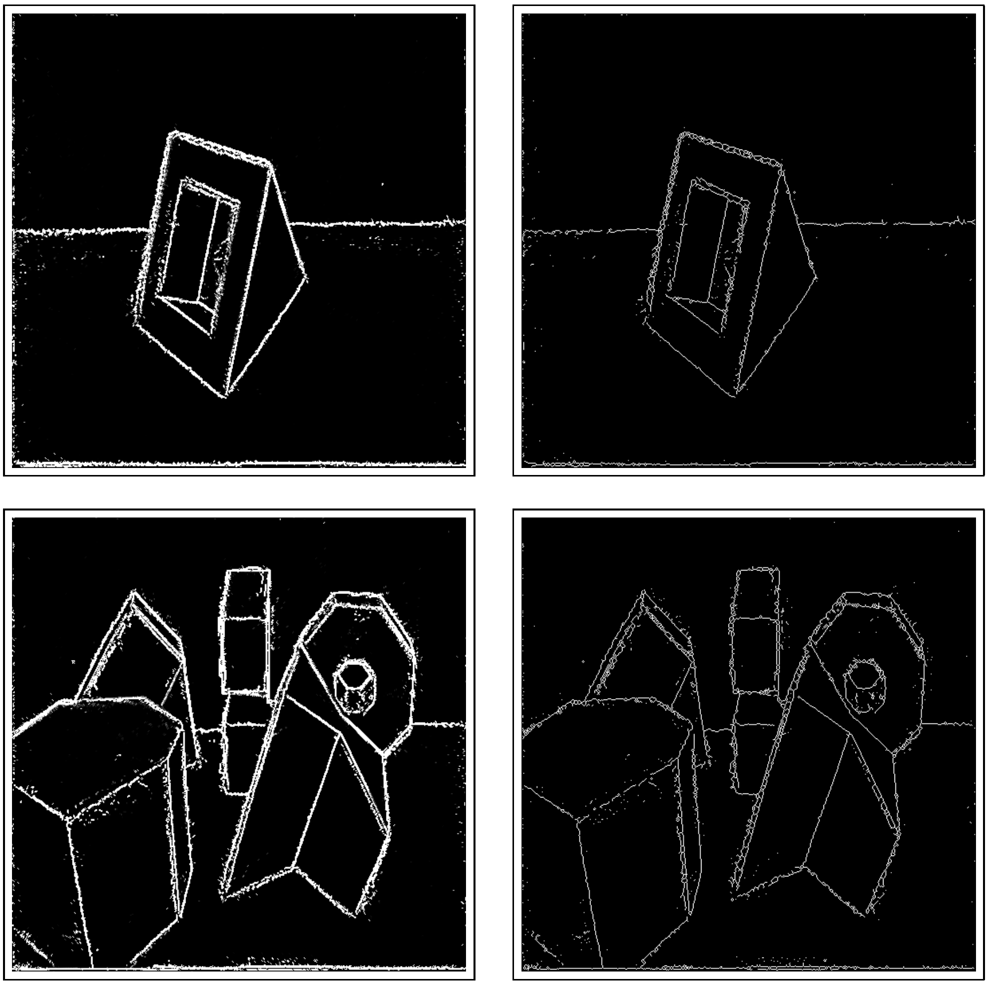


Figure 11: Results of two Perceptron images.

we show the overall edge strength map and the thinned binary edge map. The first scene in Figure 8 is another example where both planar and curved surfaces occur. The next two scenes contain surfaces such as cylinders, spheres and cones that are common in range images. Objects of more complicated shape can be seen in the remaining three scenes. The last scene contains two overlapping objects. For the image set described in [23], the results of two scenes are shown in 9, containing an adaptor and a funnel, respectively.

The other two image sets are more noisy than the Michigan images. In particular, we have observed that for ABW images the basic version of our algorithm cannot provide a precise edge localization. This problem is illustrated in Figure 10(a). After applying the edge position adaptation method we get the edge strength map and the final thinned edge map in Figure 10(b) and (c), respectively. This example confirms the usefulness of the edge position adaptation method. Also shown in Figure 10 are the results of another scene with curved objects. The performance of our edge detection algorithm on the Perceptron images is exemplified by the two scenes in Figure 11. In contrast to the Michigan images, many spurious edge points were reported due to the high noisy level in these images. In this case more elaborate thresholding methods like that described in [33] may be effective to delete the spurious edge points while retaining true edge points.

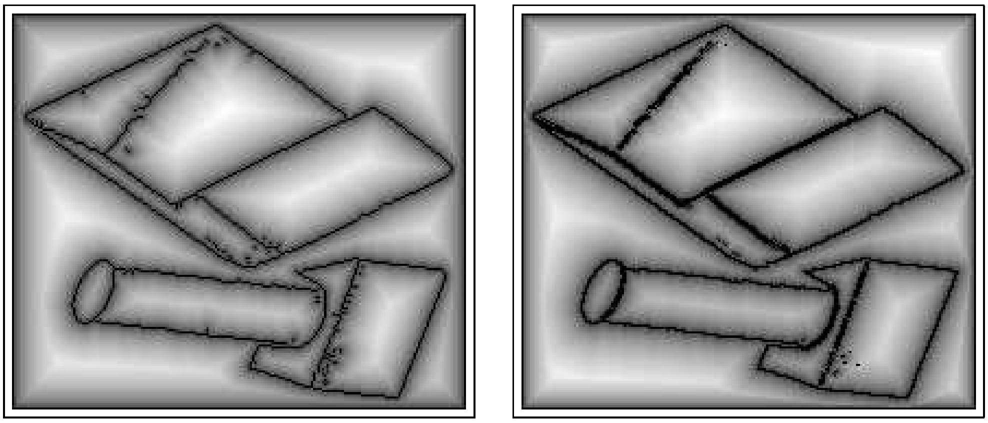


Figure 12: (a) Distance transform of the edge map in Figure 7(i). (b) Salience distance transform of the edge strength map in Figure 7(b).

The computation time for the Michigan images of a typical resolution of 200 × 200 pixels is about 1.5 seconds. All ABW and Perceptron images have a resolution of 512 × 512 pixels and require about 10 seconds.

7 Discussions and conclusion

In this paper we have presented a new method for edge detection in range images based on a scan line approximation technique. As mentioned in the introduction, the motivation of our work was to overcome some common drawbacks of the methods known from the literature. Our algorithm provides edge strength that has a straightforward geometric interpretation. This can potentially simplify the threshold determination for generating a binary edge map. Edge points found by our algorithm can be classified into several subtypes. In particular, the two subtypes convex and concave represent intrinsic properties of objects and are thus useful for the interpretation step following the image segmentation. The behavior of an optimal edge detector was defined. By means of simulation the near-optimal character of the edge detection algorithm was shown. Our method has been extensively tested on a large number of real range images acquired by three range scanners with quite different characteristics and demonstrated good results. We believe that all these aspects are important in designing a robust edge detection algorithm. So far no sufficient attention has been paid to them. In particular, optimality analysis and extensive experimental evaluation have been widely missing in the literature.

The edge detection results are useful in two ways. Edge detection is potentially able to achieve a complete segmentation. Our tests, especially those on the images acquired by the Technical Arts scanner, have demonstrated this potential. Alternatively, we can make use of edge detection to support region-based segmentation. In region-growing techniques, for instance, the quality of seed regions is of importance. Ideally, they should be far away from region boundaries and in the middle of regions. The distance transform of an edge map gives this information directly and thus provides an excellent support to finding reliable seed regions. After the distance transform has been applied, pixels are marked by their distances to the nearest edge points. Figure 12(a) shows the distance transform of the edge map in Figure 7(j) where the distance values are encoded by greylevels. If this kind of distance transform is taken into account during the seed region extraction, it is more likely to find reliable seed regions. Instead of using a binary edge map, we can also apply the so-called salience distance transform [29] to an edge strength map. In the salience distance transform all edge candidates are involved in the computation of a distance map. But their contribution is weighted by a salience value (the edge strength in our case). This way we can eliminate the need for the binarization threshold. The salience distance map of the edge strength map in Figure 7(b) is shown in Figure 12(b). Considering the usefulness of edge detection, robust edge detection methods like the one proposed in this paper will be certainly valuable to the field of range image analysis and interpretation.

In the present paper the results of our edge detection algorithm have been judged by human observation. Recently, a methodology [19] for experimentally evaluating edge detection methods for range images based on image sets with manually specified ground truth and a set of objective performance metrics was proposed. For the future we plan to apply this methodology to achieve an objective performance characterization of our algorithm and a comparison with other known methods.

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] F. Ade *et al.*, Grasping unknown objects, in H. Bunke *et al.* (Eds.), Modelling and Planning for Sensor Based Intelligent Robot Systems, 445–459, World Scientific, 1995.

[2] 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.

[3] F. Arman, J.K. Aggarwal, Model-based object recognition in dense-range images – A review, ACM Computing Surveys, 25(1), 5–43, 1993.

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

[5] P.J. Besl, Active, optical range imaging sensors, Machine Vision and Applications, Vol. 1, 127–152, 1988.

[6] P.J. Besl, Surface in range image understanding, Springer-Verlag, 1988.

[7] P. Boulanger, F. Blais, P. Cohen, Detection of depth and orientation discontinuities in range images using morphology, Proc. of 10th Int. Conf. on Pattern Recognition, Vol. B, 729–732, 1990.

[8] J.-C. Cheng, H.-S.Don, Roof edge detection: A morphological skeleton approach, in C. Archibald *et al.* (Eds.), Advances in Machine Vision: Strategies and Applications, 171–191, World Scientific, 1992.

[9] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1972.

[10] J.G. Dunham, Optimum uniform piecewise linear approximation of planar curves, IEEE Trans. on PAMI, 8(1), 67–75, 1986.

[11] P.J. Flynn, A.K. Jain, Three-dimensional object recognition, in T.Y. Young (Ed.), Handbook of Pattern Recognition and Image Processing: Computer Vision, 497–541, Academic Press, 1994.

[12] S. Ghosal, R. Mehrotra, Segmentation of range images: An orthogonal moment-based integrated approach, IEEE Trans. on Robotics and Automation, 9(4), 385–399, 1993.

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

[14] B. Günsel, A.K. Jain, E. Panayirci, Reconstruction and boundary detection of range and intensity images using multiscale MRF representations, Computer Vision and Image Understanding, 63(2), 353–366, 1996.

[15] K.K. Gupta, X.M. Zhu, Extracting polyhedral models from a range image: A hybrid approach, Technical report CSS-IS-TR 91-10, Centre for System Science, Simon Fraser University, 1991.

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

[17] B.K.P. Horn, Robot Vision, The MIT Press, 1987.

[18] R.A. Jarvis, Range sensing for computer vision, in A.K. Jain, P.J. Flynn (Eds.), Three-dimensional object recognition systems, 17–56, Elsevier Science Publishers, 1993.

[19] X.Y. Jiang *et al.*, A methodology for evaluating edge detection techniques for range images, Proc. of 2nd Asian Conf. on Computer Vision, Singapore, Vol.II, 415–419, 1995.

[20] S. Kaveti, E.K. Teoh, H. Wang, Second-order implicit polynomials for segmentation of range images, Pattern Recognition, 29(6), 937–949, 1996.

[21] N. Kehtarnavaz, S. Mohan, A framework of estimation of motion parameters from range images, Computer Vision, Graphics, and Image Processing, 45, 88-105, 1989.

[22] 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.

[23] 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.

[24] D. Mintz, Robust consensus based edge detection, CVGIP: Image Understanding, 59(2), 137–153, 1994.

[25] P.G. Mulgaonkar *et al.*, Understanding object configurations using range images, IEEE Trans. on PAMI, 14(2), 303–307, 1992.

[26] T.S. Newman, A.K. Jain, A system for 3D CAD-based inspection using range images, Pattern Recognitiuon, 28(10), 1555–1574, 1995.

[27] T. Pavlidis, S.L. Horowitz, Segmentation of plane curves, IEEE Trans. on Computers, C23, 860–870, 1974.

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

[29] P.L. Rosin, G.A.W. West, Salience distance transforms, Graphical Models and Image Processing, 57(6), 483–521, 1995.

[30] B. Sabata, J.K. Aggarwal, Surface correspondnece and motion computation from a pair of range images, Computer Vision and Image Understanding, 63(2), 232–250, 1996.

[31] 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.

[32] Tech. Arts Corp., 100x 3-dimensional scanner: User's manual and application programming guide, Redmond, WA.

[33] S. Venkatesh, P.L. Rosin, Dynamic threshold determination by local and global edge evaluation, Graphical Models and Image Processing, 57(2), 146–160, 1995.

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