Chapter 4

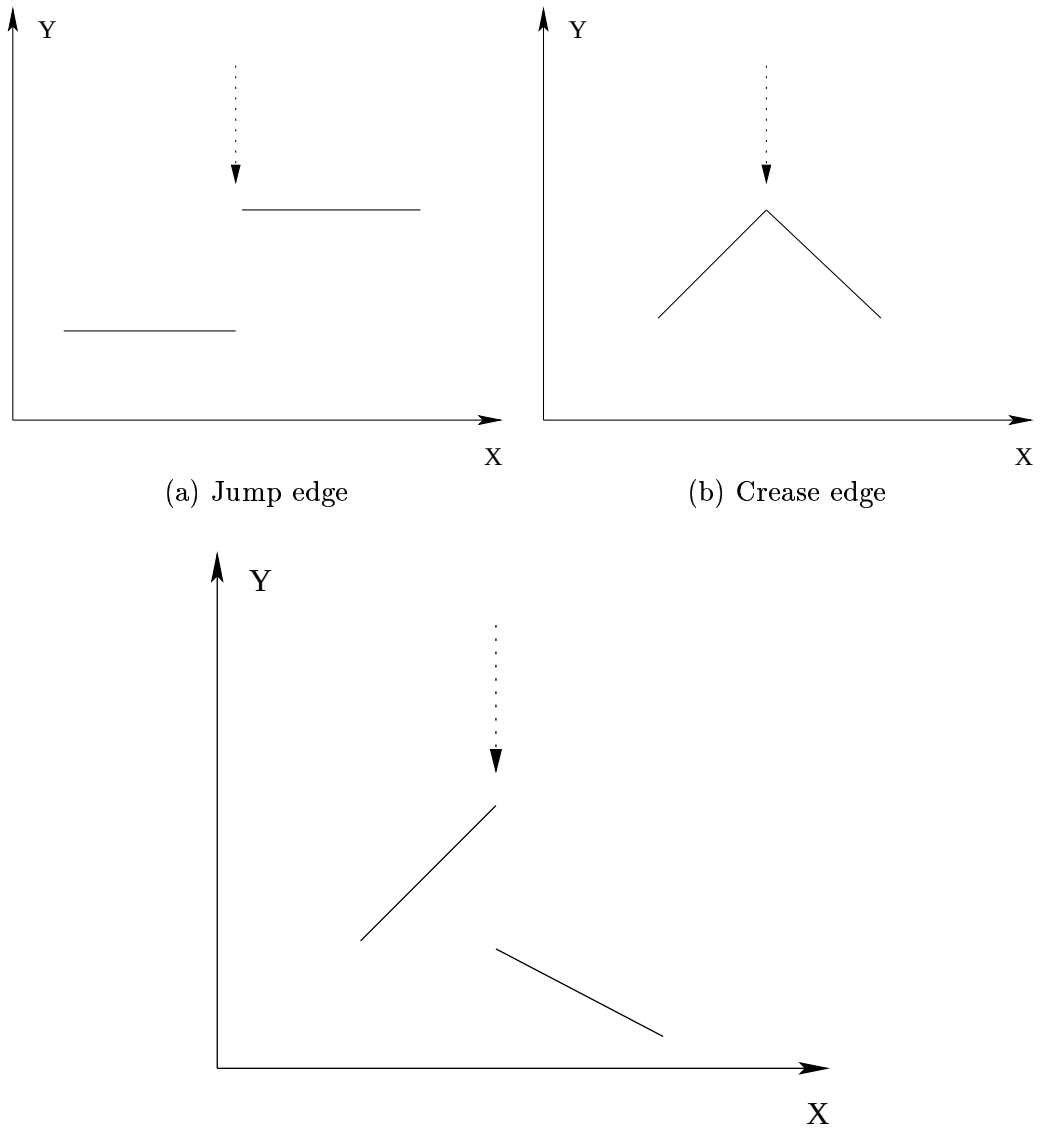
Boundary Detection in Range Images

4.1 Introduction

The great importance of boundary or edge detection as an early stage for more complicated image processing or computer vision tasks is known since many years [53], [132]. Image segmentation is perhaps the most blatant example where consideration of the outcome of edge detection results in significant improvement of its robustness: Usually, image segmentation techniques fail to provide accurate information about the shape and the position of the objects in the image. On the contrary, edge detection can accurately determine object boundaries, which are closely related to the shape and the position of the objects. Thereby, boundary detection complements image segmentation. Since our approach for localizing piled objects is based on range image segmentation, edge detection in range images is one of the most essential components of our overall strategy. In the context of our system edge detection is used as a pre- processing stage, the output of which is seamlessly incorporated in the subsequent segmentation process. The way in which this is performed, will be shown in chapter 5

A multitude of algorithms for detecting edges in intensity images have been proposed, based on local linear filtering ([53] p.175 - 181). The features of interest of these strategies are the *jump edges*, since only this type of edges is encountered in intensity images. A jump edge in an intensity image is considered as a discontinuity in the intensity values. In accordance to this definition, jump edges in range images are defined as discontinuities in depth values. Such edges occur when an object is occluded by other objects or itself. In addition to the jump edges, *crease edges* can be found in range images as well, in the position where two surfaces meet, and are characterized by discontinuities in surface normals. Besides, combinations of jump and crease edges may exist. These different types of edges, are illustrated in fig. 4.1. Fig. 4.1 (a), shows an one dimensional jump edge. Fig. 4.1 (b), shows a crease edge. Finally, Fig. 4.1 (c), shows a combination of the two edge types. In each figure a dotted arrow shows the position of the edge along the x axis of the sensor coordinate system. The reader is referred to [53] p. 473, for a more formal definition of the different kinds of edges and their properties.

The existence of more than one type of edges, renders the task of edge detection in range images more elaborate than edge detection in intensity images. The majority of algorithms proposed to deal with edge detection in range images, (e.g. [6], [22]), define the sought edges



(a) Jump edge (b) Crease edge

(c) Jump- crease edge

Figure 4.1: Edge types in range images

in a local manner, and thereby use local operations for detecting them. Their advantage is the computational efficiency. However, local determination of boundary points often leads to false alarms in the presence of noise in the image. Incorporating global shape information in the edge detection process, which is available when knowledge about the geometry of the objects appearing in the image exists, is expected to increase the robustness of an edge detection algorithm.

Our approach for edge detection in range images is inspired by [81], in the context of which edge detection via scan-line approximation with geometric parametric models is performed. The main drawback of this edge detector, namely the scan line over-segmentation problem is addressed by the introduction of a simple merging step. In addition, we incorporate a method for detection of the noisy data points created by the effect of laser beam splitting between surfaces of different ranges. Finally, a procedure for fine localization of the edge points is

introduced. Experimental results on a variety of target object configurations demonstrate that our edge detection framework exhibits increased robustness and accuracy with regard to [81]. These characteristics in combination with the computational efficiency of our edge detection approach, allows for its usage as a component of our overall system for automatic unloading of piled box-like objects.

In the sections that follow our approach for boundary detection is discussed in detail. Section 4.2, gives an overview of the scan line approximation method and highlights its advantages and problems. Section 4.3 introduces our improved version for scan line approximation. In section 4.4, experimental results demonstrate the credibility of our approach. Finally, an overview of the advantages of our approach, along with a discussion about the way in which its output will be subsequently used by our system, conclude this chapter.

4.2 The Scan-line approximation technique

One of the most popular approaches able to incorporate information about the target objects' shape in the edge detection process, is the one reported in [81]. The method approximates the rows and columns of the image, namely the scan lines, with one dimensional curves. Assuming that the objects in the image can be well modeled by implicit quadratic surfaces, approximation of the scan lines via quadratic polynomials is performed.

The framework involves usage of the scan line splitting technique of [45]: If s,e, and m the respective positions of the start, end and mid point in the scan line, the corresponding depth values in these positions y_s, y_e and y_m are used to define the parameters of the approximating quadratic model via interpolation. Subsequently, the point with the maximum distance from the approximating model is retrieved. Assume max its position in the scan line. If the distance of this point to the model is lower than a user-defined threshold ϵ, the points between s and e are considered to be satisfactory approximated by the model. Else, if the number of points contained in each segment with bounding indices s,max and $max+1$,e is bigger than the user defined threshold ν, the splitting process is recursively applied on the two segments.

Candidate edge points are the end points of neighboring segments produced by the splitting process. Fig. 4.2 illustrates: Consider the neighboring segments s_1, s_2. Assume e the end index of s_1 and $s = e+1$ the start index of s_2. Assume as well that $y = f_1(x)$, $y = f_2(x)$, the parabolas approximating s_1, and s_2 respectively. If $\overline{x} = \frac{s+e}{2}$, then the jump and crease edge strengths of candidate edge points at e and s are given by (4.1),(4.2), respectively. The candidate points whose strength is bigger than a user defined threshold are the output edge points.

$$\text{JES} = |f_1(\overline{x}) - f_2(\overline{x})| \tag{4.1}$$

$$\text{CES} = \cos^{-1} \frac{(-f_1'(\overline{x}), 1)(-f_2'(\overline{x}), 1)^T}{\|(-f_1'(\overline{x}), 1)\| \|(-f_2'(\overline{x}), 1)\|} \tag{4.2}$$

The approach exhibits many advantages: Computational efficiency is the outcome of the fast

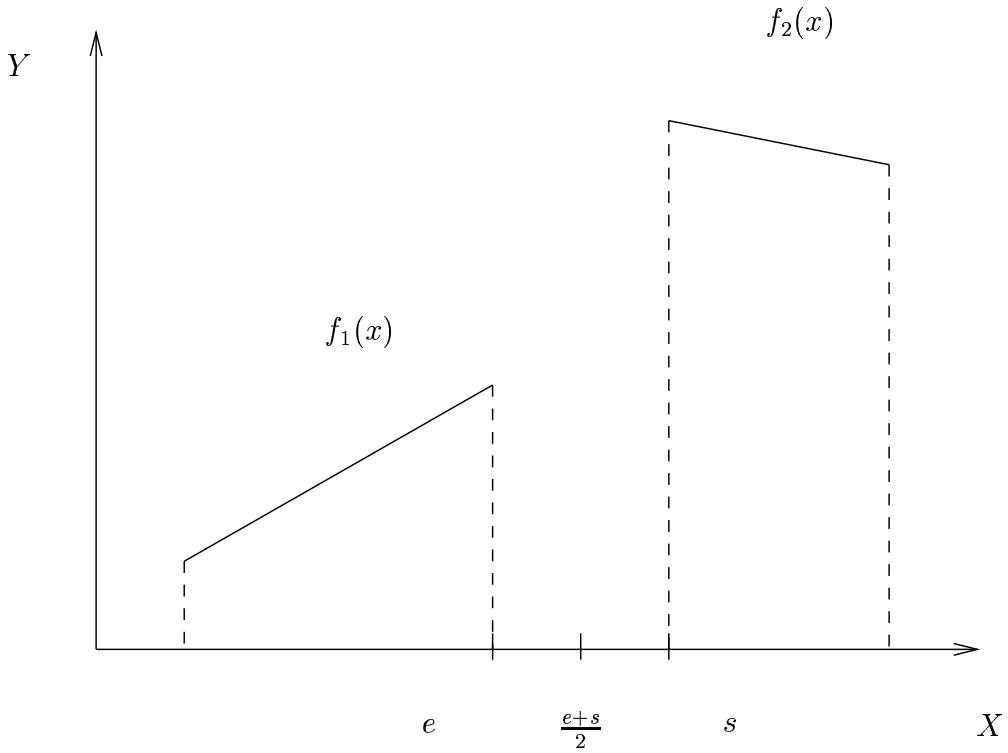
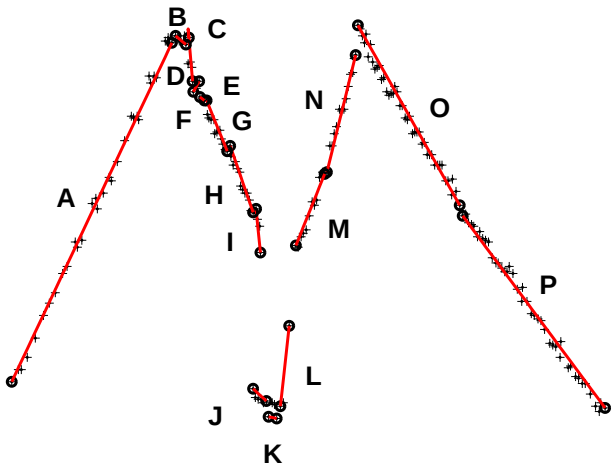


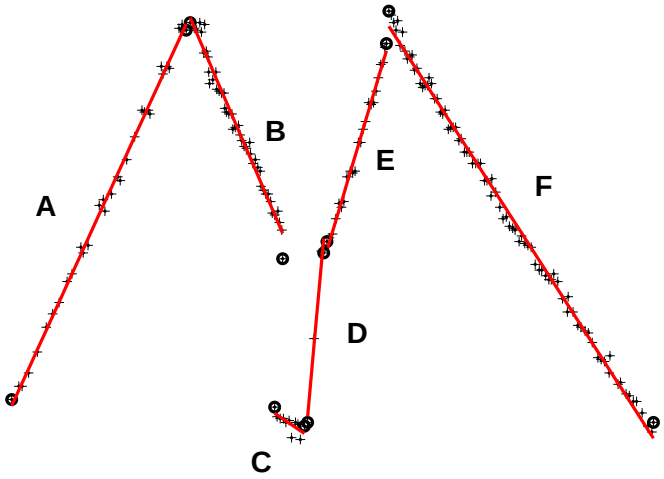
Figure 4.2: Edge detection

scan line segmentation via splitting, and of the fact that interpolation is used for the approximation. Robustness and accuracy in the localization of the edge points, is the outcome of incorporating global shape information in the process. Classification of a range point as edge point, does not any more depend on local information but on the parameters of neighboring approximating models, the estimation of which is influenced by a bigger number of range points. Finally, the approach is simple and easy to implement. The thresholds used, namely the minimum number of points per segment ν, the maximum allowable distance of a segment's point to the approximating model ϵ, and the crease and jump edge strength thresholds, have a straightforward interpretation, which renders their tuning relatively easy.

A drawback of the method however, is the scan line over-segmentation, illustrated in fig. 4.3 (a), where, for reasons of simplicity, linear instead of parabolic models have been employed for the approximation. In this image, data points are represented by crosses, and the segment end points are shown in bold. Ideally, all segments inclusively between each of **C**-**I**, **J**-**K**, **M**-**N** and **O**-**P** should be replaced with one segment. In [81], the problem is implicitly addressed by the fact that not all segment end-points are delivered as output of the edge detector, but only those with large edge strengths. The end point of segment **O** for example is determined not to be a valid edge point, since the angle between **O** and **P** is almost π. However, the problem cannot be entirely solved in this way: Boundaries of small noisy segments are falsely classified as edge points, since their crease edge strength value can be relatively large. This is the case for the boundaries of the segments **E**, **F**, for example. Experimentation with our sensor showed that such cases cause a significant degradation of the boundary detection output. Thereby, a solution to the over-segmentation problem is of significant importance



(a) Without merging



(b) Merging included

Figure 4.3: Edge detection via scan line approximation with linear models

to our application.

Another problem observed, has not entirely to do with the segmentation approach itself, but with the fact that time of flight laser sensors output noisy points in the areas of depth change, due to splitting of the laser beam [2]. Fig. 4.4 (a) illustrates: The noisy points are assigned to the segments **B** and **C** in the figure. Since these segments do not correspond to real objects, edge strength values calculated at their boundaries are unreliable, and should not be included in the output of the edge detection process. Detecting such noisy segments and discarding the respective edge information, is expected to improve the performance of the edge detection algorithm.

We employ the scan line approximation technique for edge detection. However, there are differences from the original approach of [81]. A merging step is incorporated to deal with the over-segmentation problem. In addition, a simple technique for detecting segments with

noisy points in the areas of depth change is devised. Finally, a method for more accurate localization of the edge points in the image is introduced.

4.3 Our approach for edge detection

A possible strategy for dealing with scan line over-segmentation is to increase the maximum distance threshold ϵ. Unfortunately, this may lead to the inverse effect, namely scan line under-segmentation, which is more difficult to handle. A solution to the problem is described in [135]. In this work, a method for scan line segmentation based as well on approximation with geometric models is presented. The main difference with [81], is that the segmentation is attempted without the usage of the threshold ϵ, so that segmentation becomes almost non-parametric. Scan line splitting continues until the next splitting step creates a segment with less than ν points. The resulting over-segmentation is handled by the introduction of a very simple merging step: For each segment of the final segmentation, the point with the maximum distance d from the approximating model is considered. A significance value **S** is then assigned to the segment, defined as the ratio between the length of the segment L divided by d, as in (4.3).

$$S = \frac{L}{d} \tag{4.3}$$

According to the merging procedure each segment is combined sequentially with the previous, the next and both the previous and next segments and for each combination the value **S** is computed. If one of the combinations results in a bigger **S** than the one of the candidate segment, the corresponding segments are merged. Note, that definition of the significance measure as in (4.3), implies $\nu = P + 1$, where P the number of parameters of the approximating model, so that infinite significance values are avoided.

The advantage of the merging approach is that its realization does not require introduction of additional thresholds. Experimentation with the method showed that it produces superior results when long, well approximated segments outnumber the small, probably noisy segments. In [135], this is not the case: the number of small segments is relatively large, because splitting is performed exhaustively. The splitting threshold ϵ guarantees the existence of longer segments in our case. Application of the merging step in the scan line of fig. 4.3 (a), is depicted in fig. 4.3 (b). Note that fitting is used, instead of interpolation for model approximation, in order to generate the segmentation of the figure. Using fitting results into more accurately approximated segments, and thus in more reliable significance values. Thereby, usage of fitting improves the effectiveness of the merging process.

We employ a simple but effective heuristic approach to detect and discard segments corresponding to spurious data points caused by the effect of laser beam splitting between surfaces of different ranges. These points are aligned and can be accurately modeled by linear models, which are almost parallel to the range axis **Y** of the sensor coordinate system. This is illustrated in fig. 4.4 (a). The noisy segments **B**, **C** in the figure, can be easily identified: The angle formed between each segment and the depth axis **Y** of the sensor coordinate system is computed. If the angle is smaller than a predefined threshold, the segment is considered noisy and discarded. Application of this filtering to the data of fig. 4.4 (a), is depicted in

fig. 4.4 (b). Note in the figure, that the meaningless crease edges generated between the segments **A**-**B** and **C**-**D** in fig. 4.4 (a) are replaced by a jump edge expressing the depth difference between the segments **A**-**D**, as desired. We have experimented with more sophisticated techniques for discarding these noisy points (e.g. [2]), which did not prove to deliver better results for our data sets. In addition, combining edge detection and noise filtering in one process saves computational costs.

Note, that in fig. 4.4 (a), some points expected to belong to segment **D** (namely the points between the dashed arrows in the figure), are assigned to the noisy segment **C** by the splitting process. The reason for this is that the splitting process is not allowed to generate segments with less than ν points. This problem of minor point misclassification during splitting is observed in [81], and a heuristic solution is proposed for candidate crease edge points only. For improving the position of candidate jump edge points after noisy segment deletion, points from the deleted noisy segment **C** which have a distance lower than ϵ from the approximating model of the neighboring segment **D** are added to the segment **D**. The new starting point of **D** is marked as candidate edge point. As seen in fig. 4.4 (b), the candidate edge point is accurately recovered in this way.

As discussed in chapter 3, we employ superquadrics for modeling our target objects. Since the two dimensional analog of a superquadric is a two dimensional superellipse, superellipses should be normally used for approximating the scan lines in our image. The problem is that the superelliptic model does not linearly depend on it parameters. Thus, fitting superellipses to data points requires iterative optimization techniques, which have high computational costs. This is the reason why we used linear segments for scan line approximation. This decision reduces the accuracy in edge point localization when the target objects are not planar. As we will see in the following section, the introduced inaccuracy is not high, as is the case for the computational cost savings achieved. In chapter 5, we will as well see that the introduced inaccuracy in edge detection does not prevent the subsequent segmentation framework from robustly recover the objects in the image.

Our overall approach for edge detection acts on all the rows and columns (scan lines) of the input range image. Linear models are employed for modeling our almost planar target objects, and model fitting is utilized for the approximation. For every scan line, splitting is performed. A merging step follows. Subsequently, noisy segments are identified and discarded. The boundary points of segments neighboring noisy segments are accurately localized. Finally, edge strength values are calculated and the points with high values are returned as edge points. In the paragraph the follows, experimental results give the reader an overview of the performance of our approach.

4.4 Experiments

We conducted experiments with about 20 range images corresponding to box-like object configurations. In figures 4.5, 4.6, 4.7, representative results are illustrated: Fig. 4.5 shows edge detection results for an object configuration consisting of card board objects. Fig. 4.6 shows results for a configuration consisting of box-like objects wrapped in transparent foil. Finally, fig. 4.7 shows results for a configuration consisting of bags (sacks) full of material.

For all figures, the first image shows the intensity image of the configuration. In the second row, edge detection results obtained via application of [81] to the corresponding range images are depicted. The detected edges are superimposed to the range images. The third row depicts the results of our approach. For all experiments $\epsilon = 15$ and $\nu = 3$ was used. By inspecting the figures, one can see that the accuracy on edge point localization drops when dealing with non planar objects. This is the outcome of using linear segments for approximating the superelliptic objects in the images. In the results obtained by our approach, the localization accuracy does not deteriorate sharply when dealing with non rigid objects, as is the case when [81] is applied.

$$FOM = \frac{M}{N} \times \sum_{i=0}^{N} d_i^2 \tag{4.4}$$

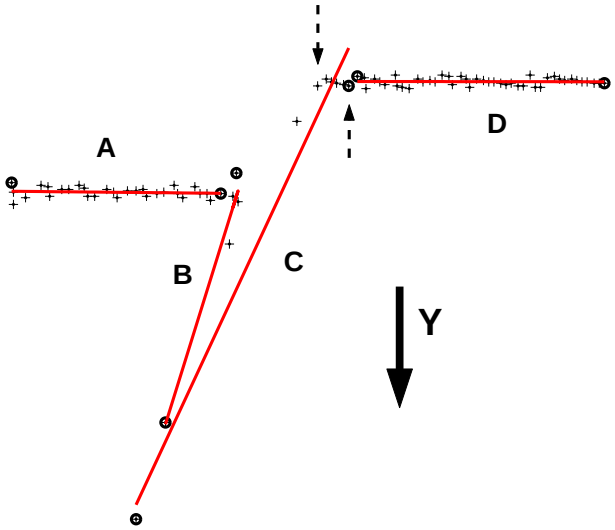
In order to acquire a more clear overview of the performance of our approach, we performed a quantitative comparison of our strategy to [81]. Assuming N the number of points in a scan line, M the number of segments and d_i the distance of the point at position i from the model approximating the point, we used the quantity defined in (4.4) to perform the comparison. The quantity FOM has been introduced in [137] and used among others in [135] for comparing segmentation results. Lower values of FOM correspond to better segmentations.

	Jiang/Bunke [81]	Our Approach
Fig. 4.5	1345	755
Fig. 4.6	1774	1074
Fig. 4.7	2375	1198

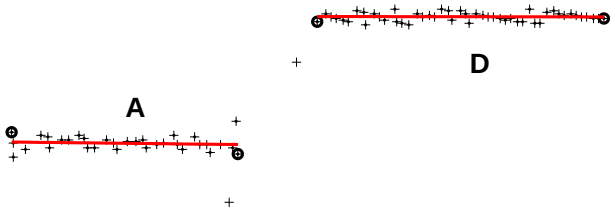
Table 4.1: Edge detection comparison results

The comparison results are illustrated in table 4.1. The first and second columns of the table correspond to the average FOM per scan line obtained by [81] and our approach, respectively. The rows of the table correspond to the range images where the methods have been applied. The first row concerns the card board boxes configuration (fig. 4.5), the second row the configuration of box like objects (fig. 4.6), and the third row the configurations of sacks (bags) of fig. 4.7. Note, that as highlighted in [133], the FOM measure favors approximations with larger number of segments. Since the number of segments produced by [81] is larger than those obtained by our approach, due to the fact that the latter includes a merging step, the FOM favors the result obtained by application of [81]. Despite this, for all cases the average FOM per scan line obtained by our approach is lower.

Regarding computational efficiency, execution of our method took about 2 seconds in a Pentium IV processor of 2.8 GHz, which implies that our technique is suitable for real time use.



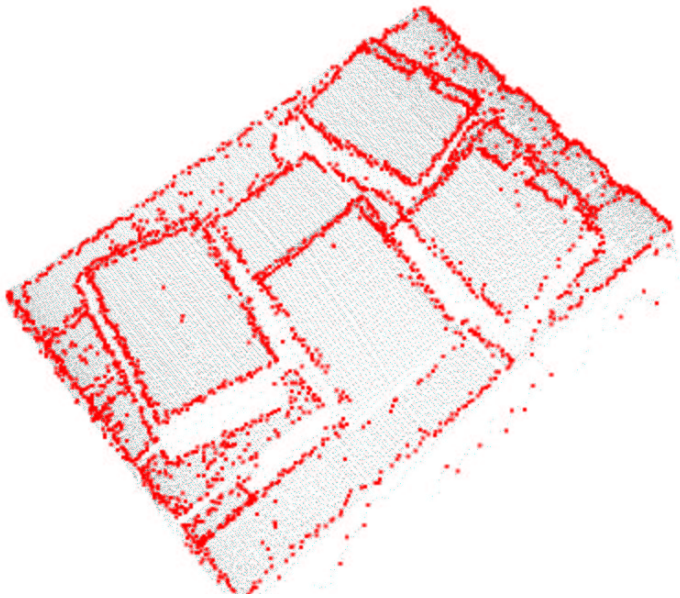
(a) Approximation output



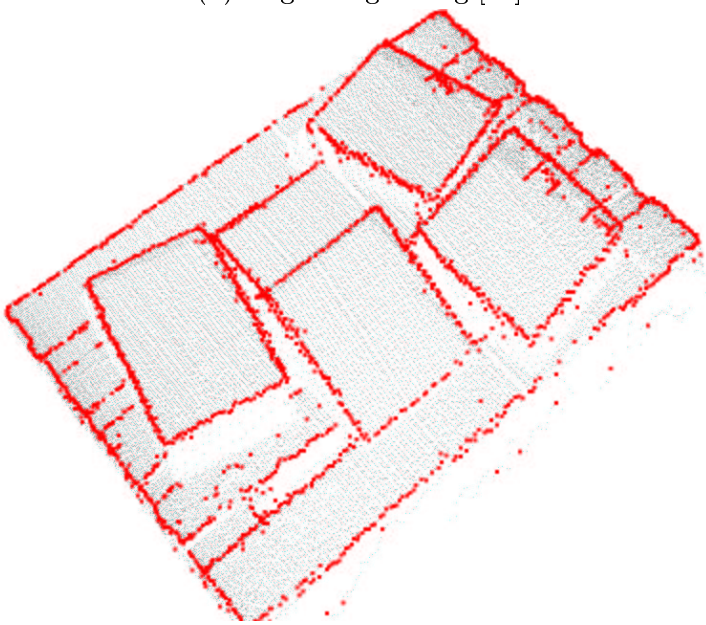
(b) Final result

Figure 4.4: Detection of noisy segments

(a) Intensity image



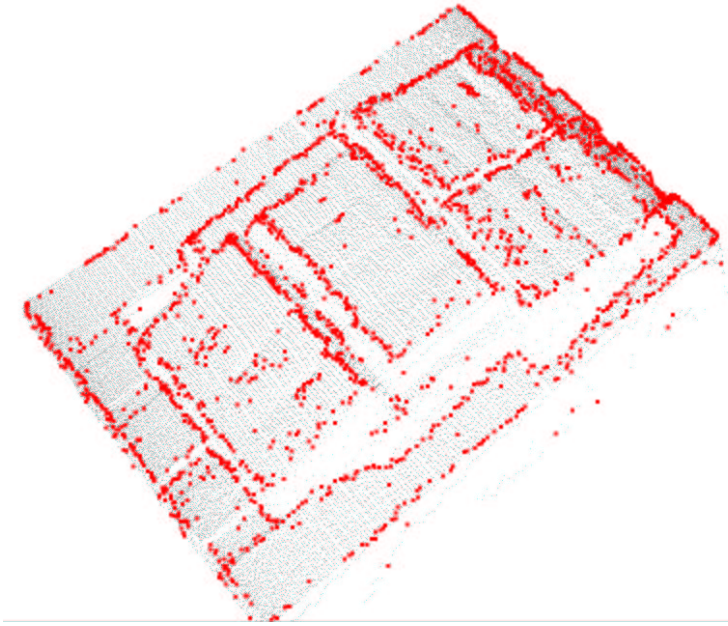
(b) Edge image using [81]



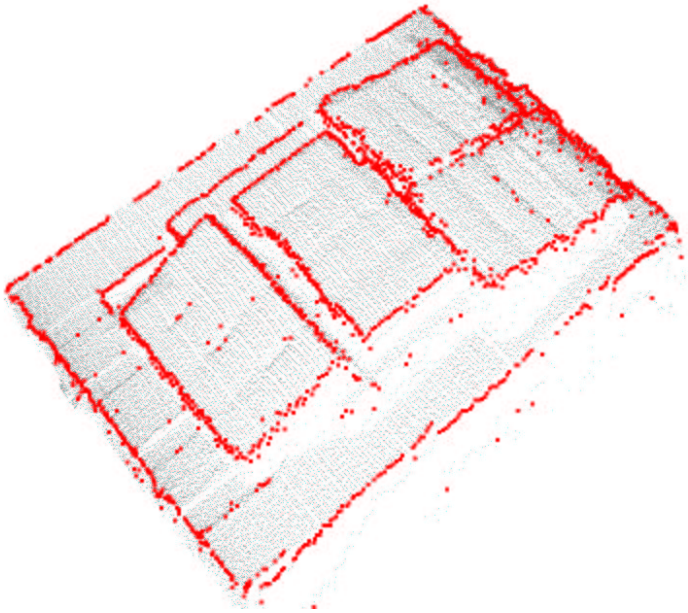
(b) Edge image using our approach

Figure 4.5: Card board boxes

(a) Intensity image



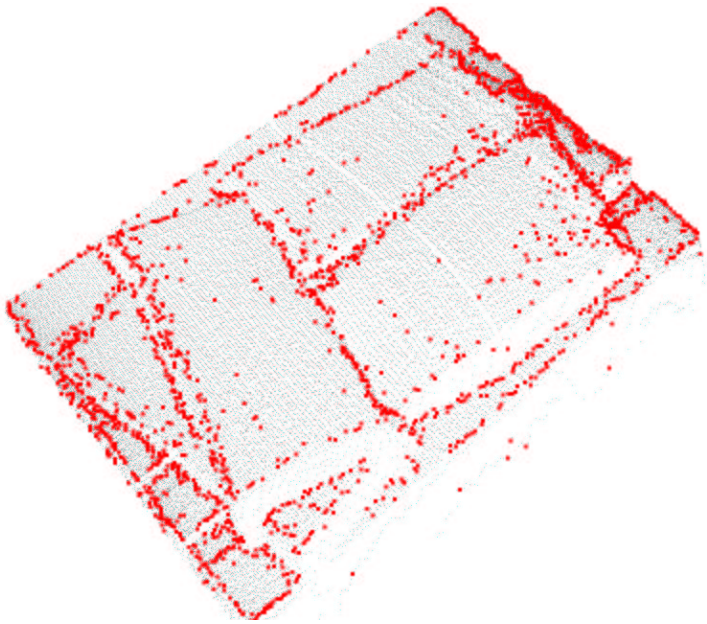
(b) Edge image using [81]



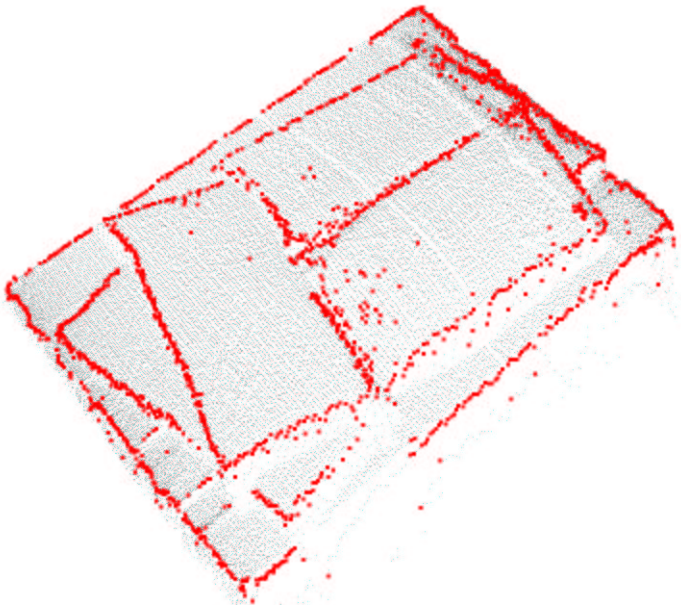
(b) Edge image using our approach

Figure 4.6: Box-like objects

(a) Intensity image



(b) Edge image using [81]



(b) Edge image using our approach

Figure 4.7: Sacks

4.5 Discussion

We presented an edge detection approach for range images, inspired by the scan line approximation technique. The main advantage of the approach with regard to local edge detectors, is accuracy in the localization of the edge points, and this stems from the fact that global shape information is incorporated in the edge detection process. In addition, our approach is more robust than the standard scan line approximation technique [81], due to the introduction of a threshold free merging step, and the incorporation of a mechanism for identification and rejection of noisy segments in the scan lines. Besides, our approach is computationally efficient. This is mainly due to the fact that the scan line segmentation mechanism is based on a fast splitting operation, as well as to the speed of the merging operation.

In the future, we plan to continue experimentation with our approach by considering a bigger variety of target objects and configurations. In addition, we plan to compare our edge detection approach with other edge detectors, using ground truth segmented images of various sensors, in the context of the range image segmentation comparison framework of [72].

Target of our overall system is to recover the graspable objects in the input image. The characteristic of these objects is that they are not occluded by others. Hence, the edge detection process should favor delivering edge points lying on the boundaries of the non occluded objects. This requirement is taken into consideration in the way in which the jump edges are retrieved. Given a jump edge, both the end point of the current segment and the starting point of the next are probable edge points. However, the point which is further away from the laser sensor, is a point which belong to an occluded object. This is the reason why given a pair of end points of neighboring segments only the point lying closer to the sensor is marked as an edge point. To exemplify, given the jump edge formed by the segments **A** and **D** of figure 4.4 (b), where both the starting point of **D** and the end point of **A** are valid edge points (since the depth difference between the segments is big enough), only the starting point of **D** is marked as an edge point since it lies closer to the sensor.

In conclusion, the information contained in the output of the edge detection process, corresponds to the position of the boundaries of the non-occluded objects in the image plane. These boundaries correspond to the boundaries of the *exposed surfaces* of the objects (see section 3.8). Hence, in our case, solution of the segmentation problem is equivalent to the localization of the the boundaries of the exposed surfaces of the objects in the image. This is not easy to realize using edge information only: As we have already seen, since we have selected very simple models for approximating the objects, there are inaccuracies in the localization of the edge points, when the objects in the pile are non-planar. In addition, the edge points detected do not form close contours, that is, there are gaps between the edges. Besides, still false alarms, that is noisy edge points, occur.

As we will see in the following chapter, boundary information is not the only information source used for segmenting the image. Region information is as well incorporated, which complements the output of the edge detection process. More specifically, the edge information produced by our top-down scan line splitting approach, is used in conjunction with a bottom up region based segmentation process to achieve robust segmentation of the objects in the input range images.