

# Robust Extraction of Vertices in Range Images by Constraining the Hough Transform

Dimitrios Katsoulas

Institute for Pattern Recognition and Image Processing, University of Freiburg  
Georges-Koehler-Allee 52, D-79110 Freiburg, Germany  
dkats@informatik.uni-freiburg.de

**Abstract.** We describe a technique for extracting vertices from range images of cluttered box-like objects. Edge detection is performed and an edge map is acquired. Extraction of vertices is carried out using the edge map and comprises two steps: Linear boundary detection in  $3D$  and boundary grouping. In order to recover the four parameters of a  $3D$  linear segment, we decompose the problem in two  $2D$  subproblems, each recovering two line parameters. These subproblems are solved by means of the Hough Transform, constrained in this way so that accurate and efficient propagation of the edge points localization error is achieved. Pairs of orthogonal boundaries are grouped to form a vertex. The orthogonality of a boundary pair is determined by a simple statistical test. Our strategy comprises many advantages, the most important of which robustness, computational efficiency and accuracy, the combination of which is not to be found in existing approaches.

## 1 Introduction

Automatic unloading and sorting of piled objects is of great importance to the industry, because it undertakes a task that is very monotonous, strenuous and sometimes quite dangerous for humans. Objects which are often encountered in industrial sites and distribution centers are mainly rigid boxes as in Fig. 4 (a) or deformable box-like objects (sacks) full of material as in Fig. 5 (a). It is advantageous to employ range imagery for dealing with the problem mainly due to relative insensitivity on lighting conditions and object texture. It is since years known in the computer community that a three-dimensional visible vertex provides the strongest constraints for accurately determining the position of convex, three-dimensional objects and thus are very good approximations of the location of the objects in space. Since the objects we are dealing with are either boxes or box-like objects, their vertices can still be used for generating accurate object location hypotheses. For this reason the robust and accurate detection of object vertices in range images is of extreme importance to this application.

Although a variety of methods for detecting corners in intensity images have been reported, this is not the case for range images. The majority of the existing approaches (like [3],[1] and others) use region information to extract vertices. The disadvantage is that the objects need to expose more than one surfaces to

the sensor for accurate estimation of the vertex position. In [7], edge detection was performed on the input range image, object boundaries were detected using the Dynamic Generalized Hough Transform [8] and vertices were extracted by grouping orthogonal object boundaries. This allows for accurate vertex detection even if the objects expose only one surface. However the error in the localization of the edge points was not taken into consideration, which made the approach not as robust as desired.

The technique discussed in this paper comprises the same two parts, as was the case in [7]: Linear boundary detection and boundary grouping. However there are essential differences between the two approaches: The three dimensional linear object boundaries are recovered, via application of an iterative algorithm to the edge map of the range image: In each iteration Hough Transforms are executed and a set of models are recovered, followed by a model selection process which retains the best models in terms of accuracy. The Hough Transforms are constrained so that the edge points localization error is accurately propagated to the parameter space. Boundaries comprising a fixed fraction of the edge points are sought in each iteration. Finally, orthogonal pairs of boundaries are grouped to a vertex. The orthogonality of a pair of recovered linear boundaries is determined via a statistical test.

This strategy results in a variety of advantages over existing systems: Robustness due to introduction of error propagation which reduces the detection rate of false positives, and due to robust boundary grouping guided by a statistical test. Accuracy, due to the incorporation of a model selection process, which retains the most accurate boundaries. Computational efficiency since the algorithm's complexity is linear to the number of edge points. Low memory consumption since accumulations use one dimensional structures. Versatility since exposure of only one object surface is enough for vertex detection, so that the system can deal with jumbled or neatly placed configurations of boxes. And last but not least simplicity as the flow diagrams that follow indicate. Our approach is described in detail in the subsequent sections.

## 2 Detection of Linear Object Boundaries in 3D

Input of our system are range images acquired from a laser range finder. Edge detection is performed on the image and an edge map is created. Such a map is depicted in Fig. 4 (b) and corresponds to the intensity image of Fig. 4 (a). The sensor coordinate frame is attached to the edge map. A range edge point  $\mathbf{D}$  is defined by the coordinates  $\mathbf{D}(X_s, Y_s, Z_s)$ . The values  $X_s, Z_s$  express its position on the two-dimensional image plane,  $Y_s$  expresses its depth value. We decided to use the detector of [6] which performs approximation of the image scan lines with linear and quadratic segments. The major advantages of this method with regard to local edge detectors are its computational efficiency and its accuracy. The latter is due to the fact that whether a range point is classified as edge point or not does not depend on local information but on the parameters of the approximated segments which intersect at the point, the determination of

which is influenced by a big number of range points. However the localization accuracy of the edge points is still not satisfactory. The surface of the target objects can not be always well approximated by parabolic segments, especially in areas where small local surface deformations occur. This is not very likely to happen when the objects are rigid boxes or deformable but full of material. Additionally, error is introduced by the laser sensor data acquisition process. The problem we face is in which way can we robustly recover the parameters of the 3D linear object boundaries of the objects from the edge map.

## 2.1 Parameter Recovery with the Hough Transform

The Standard Hough Transform (SHT) is the most common method employed for recovering multiple parametric models from images. Despite this, the SHT technique in its original form does not address the problem of localization error. Another weakness of the SHT is its computational inefficiency when dealing with models with many degrees of freedom. Lets suppose the model sought has  $N$  parameters and each image point constraints  $p$  of them. For each image point, the SHT increments all the bins comprising a  $N - p$  -dimensional manifold of an  $N$  -dimensional accumulator. In our case the models (3D lines) have  $N = 4$  degrees of freedom and each point constraints  $p = 2$  line parameters. Applying the SHT, will be both memory consuming, since a 4D accumulator is needed, as well as computationally inefficient, since mapping of a single edge point requires updating a 2D manifold of the accumulator.

A plethora of algorithms have been proposed to address the computational inefficiency of the SHT. Lets suppose mainly for simplicity  $p = 1$ , that is, we regard 2D images. The idea is to decrease the number of required accumulator updates per mapping by constraining the pose of the model. This is done by simultaneously mapping  $k$  ( $1 < k \leq N$ ) instead of one pixels to the parameter space. If so, the dimensionality of the manifold along which the accumulator must be updated drops from  $N - 1$  to  $N - k$ . In [11],[2]  $k = N$  is regarded which implies  $N - k = 0$ . In this case update of only one accumulator cell per mapping is needed. Unfortunately these approaches are not free of problems: Mapping large sets of pixels gives rise to a combinatorially explosive number of possible sets. Randomization techniques have been proposed to reduce the number of sets examined.

Other researchers propose a somewhat different solution, which is based on decomposing the Hough Transform into subproblems. Each subproblem is solved within the context of a trial: A set of points with cardinality  $d$  ( $d < N$ ) (distinguished set) is randomly selected. Random subsets of the remaining points with cardinality  $v$  (varying sets) are then considered, so that  $d + v = N$ . The union of the two sets is then mapped to the parameter space by updating one cell, since the points in the union fully constrain the pose of the model. After all the varying sets have been examined, the accumulator maxima are extracted. A trial is considered successful if those maxima satisfy user-defined criteria. The process finishes when a fixed number of trials  $t$  has been performed. There is

a variety of algorithms which may result from this framework by assigning different values to the cardinality ( $d$ ) of the distinguished set, the number of varying subsets ( $r$ ) examined within a trial and the number of trials ( $t$ ). Leavers [8] sets  $d = 1$ ,  $r$  is automatically determined by the framework, and  $t$  is the number of connected components found in the edge map. More recently Olson with his RUDR (Recognition using Decomposition and Randomization) technique [9] considers  $d = N - 1$  and  $r = n - 1$  where  $n$  the number of edge pixels in the image. A trial is considered successful when at least  $m$  points lie on the model. Finally, if  $\gamma$  the probability of failure in finding a model in the image, then the number of trials is given by (1).

$$t = \frac{\log(\frac{1}{\gamma})}{(\frac{m}{n})^{N-1}} \quad (1)$$

The selection of the particular cardinality for the distinguished set results to the fact that in each trial the transform is constrained to lie on a one-dimensional manifold, that is a curve (Hough Curve), in the parameter space. Many advantages are gained from this selection: Firstly, one dimensional data structures are used for the accumulation process, so the memory requirements are reduced to  $O(n)$ . Secondly, the complexity is  $O(tr)$  or  $O(tn)$ . Since all quantities in (1) are user-defined constants ( $\frac{m}{n}$  is a constant fraction of the input data), the overall algorithm complexity turns out to be linear to the number of pixels. Thirdly, if localization error is considered, a set of pixels maps not exactly on the Hough Curve, but to an area (error cloud) of the parameter space which lies close to the Hough Curve. The projection of the cloud to the Hough Curve will thus be a good approximation of it. This allows for simple, accurate and efficient error propagation to the parameter space. The error is expressed in a straightforward way via square boundaries in the image space whose side length ( $\delta$ ) is measured in pixels. The combination of these benefits are not to be found in other approaches simultaneously, up to our knowledge, and for this reason the adoption of RUDR framework seems to be the best choice for dealing with our problem.

## 2.2 Line Detection in 3D

In our case, each edge point constraints two out of the four line parameters. It is thus not possible to directly apply the decomposition technique discussed above, because we cannot select a particular cardinality of the distinguished set which will allow for constraining the transform to lie on an one-dimensional curve, as in the 2D case. Therefore we came to the idea to break the problem down to 2D subproblems, and the natural way to do it is to examine two such subproblems, each recovering two line parameters using the RUDR technique. In detail: A trial is initiated by randomly selecting a distinguished point  $\mathbf{D}(X_d, Y_d, Z_d)$ , which supposedly belongs to the linear boundary  $\mathbf{L}$ , shown in the edge map of Fig. 4 (c). At first, the two parameters of the 2D orthogonal projection of  $\mathbf{L}$  to the image plane ( $\mathbf{ZX}$ ) are estimated. Fig. 1 (a) illustrates: The orthogonal projections of all the edge points to the image plane are taken into account. Lets

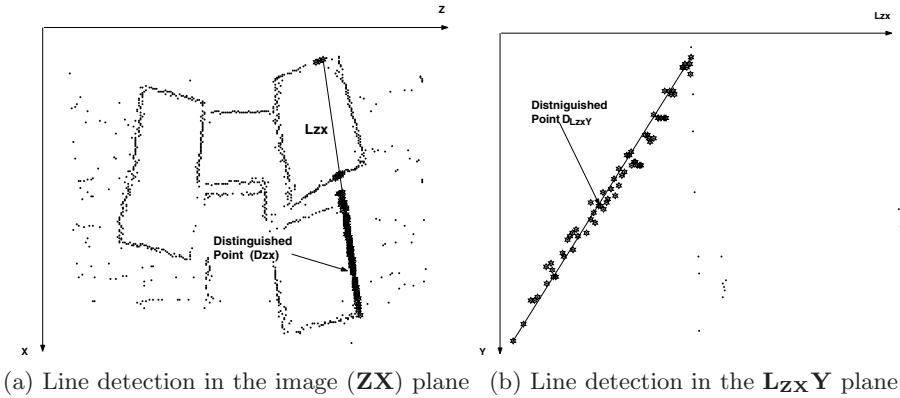


Fig. 1. 3D Line detection in two steps

consider  $D_{zx}$  the projection of  $D$ . A two-dimensional RUDR trial is performed on the image plane with  $D_{zx}$  as distinguished point and the parameters of  $L_{zx}$  are retrieved. The corresponding range points to the pixels contributed to the accumulator's maximum (drawn as "\*" in Fig. 1 (a)) are then projected to the plane defined by  $L_{zx}$  and the axis  $Y$  of the sensor coordinate system. Lets consider now  $D_{LzxY}$  the projection of  $D$  to this plane. A second two-dimensional RUDR trial is performed on this plane with  $D_{LzxY}$  as distinguished point to retrieve the remaining two parameters of  $L$ . Fig. 1 (b) illustrates. The range points finally determined to belong to the line  $L$  correspond to the 2D points drawn as "\*" in this figure. The flow diagram of the trial for detection of a 3D line is depicted in Fig. 2 (a).

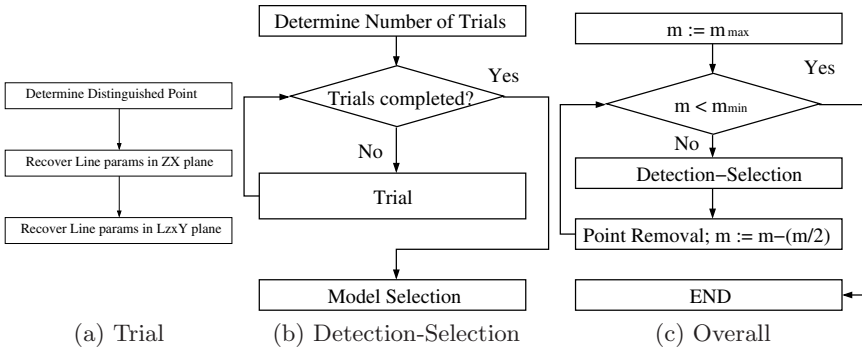


Fig. 2. Flow diagrams

### 2.3 Model Selection

Boundary detection to the edge map of Fig. 4 (b) is presented in Fig. 3 (a), where the detected lines are superimposed to the edge map. The problem observed, is that the line detection process outputs redundant lines. This is a consequence of the randomization. Our algorithm as is, cannot guarantee that more than one points belonging to the same boundary will not be used as distinguished pixels of the recovery process. Simply removing the range points determined to belong to a boundary after a successful trial and continue the process is questionable. We cannot assure that some of these points cannot be used by a later trial to recover a model which represents the boundary better. Instead of retaining a locally sufficient model it is preferable to wait until all the trials take place and retain the models which satisfy some global optimality criteria. It is logical to assume that a recovered model should be favored over another if it describes a bigger number of image points more accurately. The latter statement is a simplified version of the Minimum Description Length (MDL) principle for model selection, which has been used quite frequently in various computer vision applications, lately in [5], [10]. We adopt the strategy of in [5] p.123 for formalizing our approach, mainly due to its compactness and simplicity: Lets suppose that the recovery process outputs  $M$  models (in our case  $3D$  lines). We regard a vector  $\mathbf{m}$  of size  $M$ , the element  $m_i$  of which has the value 1 if the model  $i$  is contained in the final description and 0 if not. We consider as well a  $M \times M$  matrix  $\mathbf{Q}$ , the diagonal terms  $q_{ii}$  of which express the benefit value for a model, while the others  $q_{ij}$  handle the interaction between the possibly overlapping models  $i$  and  $j$ . A model benefits when considered in the final description if it describes many data points with high accuracy as expressed by (2), where  $V_i$  the variance and  $|M_i|$  the number of points of the model  $i$ ,  $K_1, K_2$  user defined constants.

$$q_{ii} = K_1|M_i| - K_2V_i \tag{2}$$

We always penalize overlapping models, thus the benefit of overlapped models is negative and analogous to the number of points explained by both models, as expressed by (3).

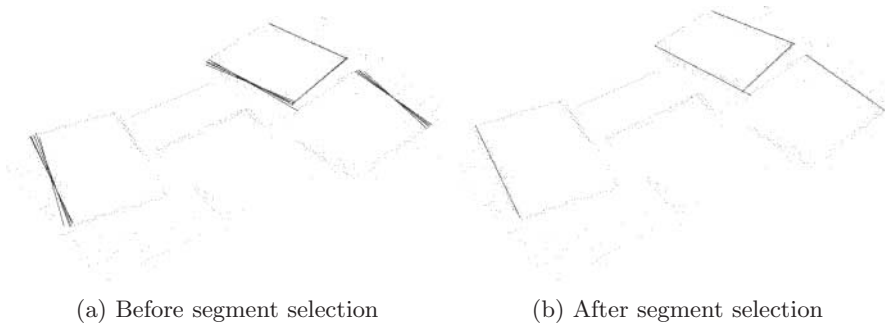
$$q_{ij} = -\frac{K_1|M_i \cap M_j|}{2} \tag{3}$$

The function the maximization of which will result to the selection of the optimal set of models, is given by (4).

$$F(\mathbf{m}) = \mathbf{m}^T \mathbf{Q} \mathbf{m} \tag{4}$$

Many approaches can be applied for maximizing (4), among them simulated annealing or neural networks. A greedy algorithm of  $O(M^2)$  is selected [5] for efficiency reasons. Fig. 3 (b) illustrates the results of model selection on the lines detected in Fig. 3 (a).

We name the process of line Detection followed by model Selection the **D-S** process, the flow diagram of which is illustrated in Fig. 2 (b). The algorithm's complexity is the complexity of the line detection plus the complexity of the



**Fig. 3.** Effect of segment selection

selection that is  $O(tn) + O(M^2) \approx O(n)$ , since  $t = O(1)$  and  $n \gg M$ . In terms of memory consumption, all we need is one dimensional accumulator of size  $O(n)$  plus a two dimensional matrix of size  $M \times M$ . The **D-S** process inherits its robustness from the RUDR approach but it is more accurate because the best segments are retained by the selection process.

## 2.4 Acceleration via Point Removal

Our algorithm as is, performs  $t$  trials followed by model selection to extract 3D lines from the input edge map on which at least  $m$  points lie. The model selection, guarantees that the remaining linear segments describe the edge points to which they correspond in an optimal way. In other words, it is highly probable (this probability is given by the quantity  $1 - \gamma$ ) that no other segments can be found comprising  $m$  or more points other than those already discovered. This observation results to a substantial algorithm acceleration: We adopt an iterative approach, every iteration of which comprises a **D-S** process retrieving lines with at least  $m$  points followed by a point removal, so that all the points determined by the **D-S** to lie on lines are eliminated from further consideration. We start by looking for long segments, so  $m$  is assigned a big value ( $m_{max}$ ) and then we gradually reduce the number of points expected to be found on a segment by  $\frac{m}{2}$  until a lower threshold ( $m_{min}$ ) is reached. The execution time of the **D-S** step is proportional to the number of edge points in the image times the number of trials. The latter is inversely proportional to the number of points expected to lie in the lines. By looking for lines comprising many points first, we reduce the number of trials and thus the execution time of the current **D-S** step. By point removal reduction of execution time of the subsequent **D-S** steps is guaranteed. Thus an overall algorithm acceleration is realized. The flow diagram of the entire algorithm is depicted in Fig. 2 (b).

### 3 Boundary Grouping

We define a  $3D$  vertex as an aggregate consisting of two orthogonal  $3D$  linear segments and a vertex point defined by their intersection. In the ideal case, two linear segments  $\mathbf{X}, \mathbf{Y}$  comprise a vertex if the dot product of their direction vectors  $\mathbf{x}, \mathbf{y}$  is zero, that is:  $r = \mathbf{x}^T \mathbf{y} = 0$ . However due to uncertainty in the estimation of the segment parameters the dot product can never be exactly zero and a threshold must be introduced to determine the validity of a grouping hypothesis. The threshold depends on the uncertainty in the calculation of the line parameters and thus is difficult to define. The dot product is a bilinear function of the direction vectors. Thus rigorous uncertainty propagation can be achieved and a statistical test can determine whether the grouping hypothesis is to be rejected or not, based on a user defined significance value. In [4] a compact framework for testing uncertain geometric relations is presented, on which our method is based. If we assume Gaussian noise and if  $\Sigma_{xx}$  and  $\Sigma_{yy}$  the covariance matrices of the direction vectors  $\mathbf{x}$  and  $\mathbf{y}$  respectively, the variance of their dot product is given by the expression:

$$\sigma_r^2 = \mathbf{x}^T \Sigma_{yy} \mathbf{x} + \mathbf{y}^T \Sigma_{xx} \mathbf{y} \quad (5)$$

The optimal test statistic for the hypothesis  $H_0 : r = 0$  is given by:  $z = \frac{r}{\sigma_r} \sim N(0, 1)$ . We select a significance value  $\alpha$  and compare the value  $z$  with the value  $N_{1-\alpha}(0, 1)$ . If  $z > N_{1-\alpha}(0, 1)$ , the grouping hypothesis is rejected. In all our experiments  $\alpha$  was set to 0.05. The overall grouping algorithm has as follows: All possible pairs of detected lines are considered and those pairs passing the statistical test along with their intersection points are inserted to the set of the detected vertices.

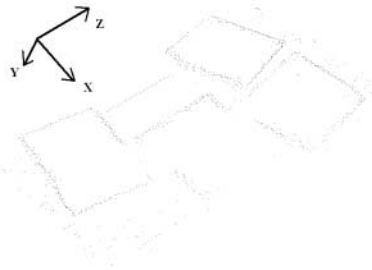
### 4 Experimental Results

We applied our algorithm in various range images corresponding to piled objects. Two test cases are presented here: Rigid card-board boxes (Fig. 4) and sacks (bags) full of material (Fig. 5). The edge map, the recovered  $3D$  line segments and the extracted vertices are as well depicted. In both cases all the objects linear boundaries were successfully recovered except those which were very noisy and comprised few number of points. For the boxes case we had  $n = 1015$  edge points and assumed error of  $\delta = 0.6$  pixels during the detection in the image plane and  $\delta = 4$  pixels during line detection in the  $\mathbf{L}_{Z\mathbf{X}}\mathbf{Y}$  plane. The corresponding values for the sacks test case were  $n = 1260$ ,  $\delta = 1$ , and  $\delta = 4$ . In both cases, the probability of failure was  $\gamma = 0.01$  and the model selection parameters were set to  $K_1 = 1$ ,  $K_2 = 0.1$ . Two algorithm iterations were executed: The first detected lines comprising at least 60 and the second 30 range points. The execution time for vertex detection was about 12 seconds in both cases in a Pentium 3,  $600MHz$ . Note that if we execute only one iteration in the context of which lines comprising 30 points are sought from the first place, the execution time rises to about 19 seconds. This verifies that the iterative algorithm and point removal actually reduce the overall execution time.

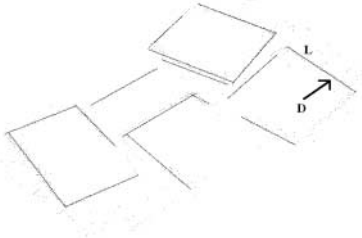




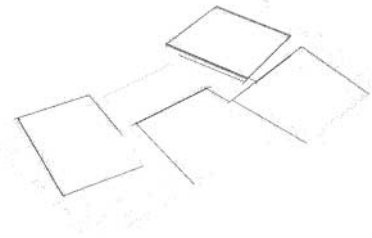
(a) Intensity



(b) Edge Map



(c) 3D Line Segments

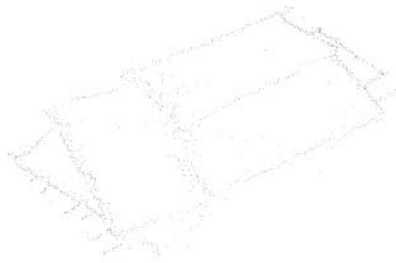


(d) Vertices

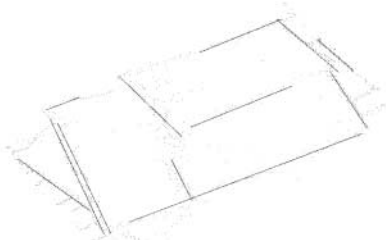
**Fig. 4.** Boxes



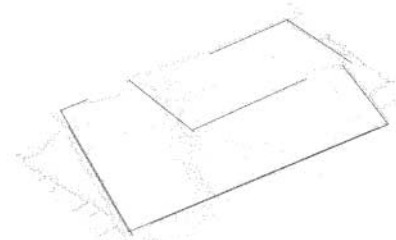
(a) Intensity



(b) Edge Map



(c) 3D Line Segments



(d) Vertices

**Fig. 5.** Sacks

## 5 Conclusions

We presented a technique for detecting vertices in range images of cluttered objects in two steps: Boundary extraction and grouping. Recovery of the four parameters of the linear boundary segments was performed by a sequence of problem decomposition, model selection and point removal, integrated into an iterative framework. Efficient and accurate propagation of error to the parameter space was achieved so that robustness was realized. Boundary grouping via a statistical test contributed to the system's robustness. Vertex extraction corresponds to the recovery of a subset of the parameter set describing each object on the pallet. In the future we intend to use this framework as a starting point for the recovery of the remaining parameters of each object.

## References

- [1] A. J. Baerveldt. Robust *Singulation of Parcels with a Robot System using multiple sensors*. PhD thesis, Swiss federal institute of technology, 1993. 360
- [2] R. C. Bolles and M. A. Fischler. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 637643 August 24-28 1991. 362
- [3] C. H. Chen and A. C. Kak. A robot vision system for recognizing 3-D objects in low-order polynomial time. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1535-1563 November-December 1989. 360
- [4] W. Foerstner, A. Brunn, and S. Heuel. Statistically testing uncertain geometric relations. In G. Sommer, N. Krueger, and Perwass Ch., editors, *Mustererkennung*, pages 17-26. Springer, September 2000. 367
- [5] A. Jaklić A. Leonardis, and F. Solina. *Segmentation and recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer Academic Publishers, Dordrecht, 2000. 365
- [6] X. Jiang and H. Bunke. Edge detection in range images based on scan line approximation. *Computer Vision and Image Understanding: CVIU* 73(2):183-199 February 1999. 361
- [7] D. Katsoulas and L. Bergen. Efficient 3d vertex detection in range images acquired with a laser sensor. In B. Radig and S. Florczyk, editors, *Pattern Recognition, Proc. of 23rd DAGM Symposium*, number 2191 in LNCS Pattern Recognition, pages 116123. Springer, September 2001. 361
- [8] V. F. Leavers. The dynamic generalized Hough transform: its relationship to the probabilistic Hough transforms and an application to the concurrent detection of circles and ellipses. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 56(3):381-398 November 1992. 361, 363
- [9] C. F. Olson. A general method for geometric feature matching and feature extraction. *International Journal of Computer Vision*, 45(1):39-54 October 2001. 363
- [10] S. Sclaroff and L. Liu. Deformable shape detection and description via model-based region grouping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):475-489 2001. 365

- [11] L. Xu and E. Oja. Randomized Hough transform (RHT): basic mechanisms, algorithms, and complexities. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 57:131-154 1993. [362](#)