Localization of Piled Boxes by Means of 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. Automatic unloading of piled boxes of unknown dimensions is undoubtedly of great importance to the industry. In this contribution a system addressing this problem is described: a laser range finder mounted on the hand of an industrial robot is used for data acquisition. A vacuum gripper, mounted as well on the robot hand is employed from grasping the objects from their exposed surfaces. We localize the exposed surfaces of the objects via a hypothesis generation and verification framework. Accurate hypotheses about the pose and the dimensions of the boundary of the exposed surfaces are generated from edge information obtained from the input range image, using a variation of the Hough transform. Hypothesis verification is robustly performed using the range points inside the hypothesized boundary. Our system shows a variety of advantages such like computational efficiency accuracy and robustness, the combination of which cannot be found in existing approaches.

1 Introduction

We address the depalletizing problem, in the context of which a number of objects residing on a platform, the pallet, should be automatically localized grasped and unloaded. More specifically, we present a system for automatic unloading of piled boxes of unknown dimensions, since such objects are quite often encountered in industrial sites. Existing systems utilizing intensity cameras for dealing with the problem [8], [4], depend heavily on lighting conditions at the installation sites, and deal primarily only with neatly placed configurations of objects. Systems utilizing range imagery [1], [7] on the other hand, utilize region information to determine object boundaries which makes them not as accurate as desired.

We employ a laser sensor mounted on the hand of an industrial robot for data acquisition. A vacuum gripper mounted as well on the robot hand, grasps the objects from their exposed surfaces. Both boundary and region based information provided by input range images, are used for localizing fully exposed object surfaces. Boundary information creates accurate hypotheses about the pose and the dimensions of the boundaries of the objects' exposed surfaces, which are verified or rejected using the data inside the boundaries. Exposed surfaces are modeled using parametric geometric entities. The problem of efficiently creating

B. Michaelis and G. Krell (Eds.): DAGM 2003, LNCS 2781, pp. 44–51, 2003.

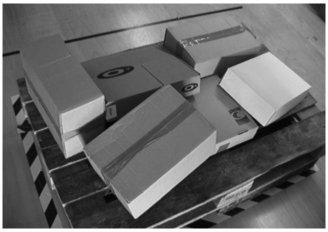
accurate hypotheses about the parameters of those surfaces in the pile is solved by decomposition into various subproblems, each recovering a subset of each surface's parameter set. Our system exhibits various advantages the combination of which cannot be found in existing systems: Insensitivity to lighting conditions, since a laser sensor is employed for data acquisition. Accuracy, due the generation of accurate pose hypotheses. Robustness, since acceptance of a hypothesis is determined by statistical tests which take into consideration the uncertainty in the calculation of features. Computational efficiency, due to problem decomposition in subproblems with lower complexity. In addition, our framework allows for parallel implementation, which can reduce its running time to a considerable extent. In the paragraph that follows, our technique is described in detail.

2 Finding Graspable Surfaces of Piled Boxes

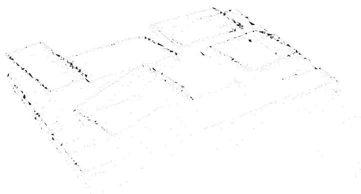
One of the most important properties of an automatic unloading system is that during its operation, it does not destroy the objects of the pile. This suggests that unloading operations should be performed in such a way, so that the objects on the top of the pile are grasped first. The particular objects are expected to fully expose one of their surfaces to the laser sensor. These surfaces are three dimensional planar areas with a rectangle boundary. Unloading of objects can be achieved by grasping the objects from the center of gravity of the fully exposed surfaces. The fully exposed surfaces will therefore be hereinafter referred to as graspable surfaces. The rectangle boundaries of graspable surfaces are geometric entities that can be expressed through eight parameters. Six of them represent their pose (translation and rotation) in space, and the remaining two their dimensions (width and length). Our system should ideally be in the position to localize all the graspable surfaces contained in the range image, which could enable the unloading of multiple objects per scan. The problem we deal with, has therefore to do with the recovery of multiple instances of geometric models in range images.

The Hough Transform is the most common method employed for dealing with such problems. However, the technique in its original form (Standard Hough Transform, SHT) has drawbacks: Lets suppose the model sought has N parameters and each image point constrains 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 ($3d$ rectangles) have $N = 8$ degrees of freedom and each point constrains $p = 2$ model parameters. Applying the SHT, will be both memory consuming, since a $6d$ accumulator is needed, as well as computationally inefficient, since mapping of a single image point requires updating a $4d$ manifold of the accumulator. A second drawback of the SHT is that it does not take into consideration the error in the localization of the image points. This results in both detection of false positives and missing of objects, thus negatively affects the robustness and effectiveness of the transform. The reader is referred to [9] for details on the issue.

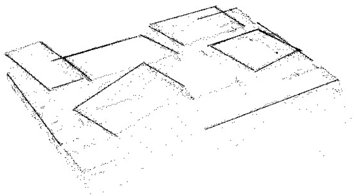
We recover the bounding rectangles of the graspable surfaces of our objects from the edge map of the range image by using a variation of the Hough Transform. We overcome the computational inefficiency of the transform by decomposing the recovery problem into two successive subproblems, each dealing with a subset of the boundary parameter set: The recovery of the pose parameters followed by the recovery of the dimensions. In addition, taking into consideration the error in the localization of the edge points when mapping them to the parameter space, results into robustness and accuracy. A detailed description of the pose and the dimensions recovery subproblems is presented in the subsequent paragraphs.



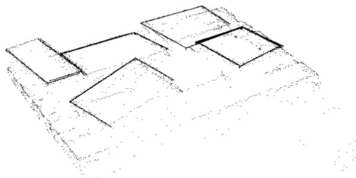
(a) Intensity image



(b) Edge Map



(c) Lines



(d) Vertices

Fig. 1. Vertex detection in range images

2.1 Recovery of Pose

It is since years known in the computer vision community [2], that a visible vertex of a convex object provides the strongest constraints for accurately determining its pose. Object vertices are recovered via the edge map of the range image of the pile. The technique comprises two steps: Firstly, three-dimensional lines corresponding to the linear boundaries of the boxes are extracted. Secondly, all pairs of lines are considered. Pairs of lines found to be orthogonal, along with their intersection point are grouped to a vertex. Line detection in $3d$ is performed via a series of Hough Transforms (see [6] for details). An interesting feature of our vertex detector is that constrains the transform in this way, so that it allows for efficient and accurate propagation of the edge points localization error in the parameter space.

The outcome of the vertex detection process is depicted in Fig.1. Fig.1 (a) is an intensity image of the pile, Fig.1 (b) depicts the outcome of the edge detection operation on the input range image. Line detection in $3d$ and vertex recovery both superimposed on the edge map are presented in Fig.1 (c) and (d) respectively. The reader may have already observed that not all the linear boundaries and as a consequence not all of the vertices of the graspable surfaces have been recovered. The adopted line detection guarantees detection of all boundaries in the image up to a user defined probability of success (see [6],[9]). The execution time of the algorithm depends exponentially on this number. In order to balance computational efficiency and functionality we set the probability of success to a value less than one, namely 0.9. We thus deliberately allow about 10 per-cent of the boundaries to be missed by our algorithm, for the sake of efficiency.

2.2 Recovery of Dimensions

The dimensions of the boundary of a graspable surface of a known pose, can be directly determined from two diagonal vertices of it. In our case, not all the linear boundaries and thereby not two diagonal vertices of each graspable surface can always be detected. To be able to infer the dimensions of the boundary of an exposed surface even in cases when two not diagonal or only one of its vertices is detected, we employ an approach which uses both the already extracted vertices as well as the edge points. The algorithm for finding graspable surfaces of boxes in range images is presented in Fig. 2.

The procedure **findGraspableSurfaces** (see Fig. 2, line 1) attempts to recover the graspable surfaces. Input of the procedure is the set of detected vertices $\mathbf{V}$. For every element V_i of the set, a rectangle graspable surface boundary R is initialized (line 2). The pose of R is recovered, by alignment with V_i. Thereby V_i will be hereinafter referred to as the *generating vertex* of R. Then, the algorithm finds the dimensions of R: At first it attempts to do so by finding a scene vertex which lies diagonal to V_i (line 4). If such a vertex cannot be found it attempts to recover the dimensions from edge points (line 7). If one of the two processes is successful R is added to the list of found graspable surface boundaries $\mathbf{R}$ (line 5,8).

The procedure **dimensionsFromVertices** (line 11) aims at recovering the dimensions of the input rectangle R by finding a scene vertex which is diagonal to the rectangle's generating vertex. Such vertex should be on the same plane to which the generating vertex belongs and its direction vectors should be parallel to the corresponding direction vectors of the generating vertex (line 13). In addition, its intersection point should reside at the first quadrant of the coordinate frame defined by the generating vertex (line 14). When a vertex satisfying the above criteria is encountered, the algorithm updates the width and length parameters of the rectangle R (line 15). There are cases however when a vertex with correct properties is found, which belongs to the boundary of an exposed surface of a different box. In order to identify such cases we regard the range points inside

R. If the average distance of the points to the plane defined by R is small enough, we consider the rectangle successfully localized. This test is realized by the procedure **verify**, invoked in line 16. Points inside R are acquired via a computationally efficient region rasterization framework [10].

1. **findGraspableSurfaces**($\mathbf{V}$, α, p):
2. *For* every vertex $V_i \in \mathbf{V}$ /* $\mathbf{V}$ is the set of detected vertices*/
3. consider **Rectangle** R; align R with V_i
4. *If* **dimensionsFromVertices**(R, $\mathbf{V}$, α) *Then*
5. add R to $\mathbf{R}$ /* $\mathbf{R}$ is the set of recovered graspable surface boundaries*/
6. *Else*
7. *If* **dimensionsFromEdges**(R, α, p) *Then*
8. add R to $\mathbf{R}$
9. **select**($\mathbf{R}$) /* Retain the “best” boundaries*/
10. *Return* $\mathbf{R}$

11. **dimensionsFromVertices**(R, $\mathbf{V}$, α):
12. *For* every vertex $V_j \in \mathbf{V}$
13. *If* **coplanar**(R, V_j, α) and **parallel**(R, V_j, α)
14. *If* **inFirstQuadrantOf**(R, V_j) *Then*
15. update dimensions of R
16. *Return* **verify**(R, α)
17. *Return False*

18. **dimensionsFromEdges**(R, α, p):
19. $\mathbf{P_c} \leftarrow$ **preProcess**(R, α) /* $\mathbf{P_c}$: the set of candidate edge points */
20. $\mathbf{A_x}$, $\mathbf{A_y} \leftarrow$ **accumulate**($\mathbf{P_c}$)/*$\mathbf{A_x}$,$\mathbf{A_y}$: one dimensional accumulators*/
21. *For* every peak $A_x \in \mathbf{A_x}$
22. $M_x \leftarrow$ parameter value corresponding to A_x (width)
23. *For* every peak $A_y \in \mathbf{A_y}$
24. $M_y \leftarrow$ parameter value corresponding to A_y (length)
25. $\mathbf{P_i} \leftarrow$ points which contributed to A_x, A_y
26. $\mathbf{P_f} \leftarrow \{\text{points } P(x, y) \in \mathbf{P_i} : x \leq M_x \wedge y \leq M_y\}$
27. *If* $\mathbf{P_f}$.**size**() $> p$
28. dimensions of $R \leftarrow M_x, M_y$
29. *Return* **verify**(R, α)
30. *Return False*

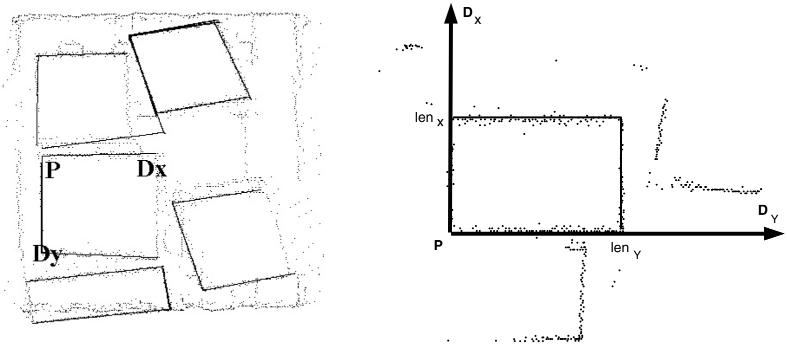
Fig. 2. Algorithm for finding graspable surfaces of piled boxes

The procedure **dimensionsFromEdges** (line 18) recovers the dimensions of the input rectangle R, in the event of insufficient vertex information, that is when no diagonal vertex to the generating vertex of R can be found. We infer dimension information from the edge points expected to reside on R. These points should satisfy the following requirements: Firstly they should be coplanar to the plane defined by R. Secondly they should be in the first quadrant of the coordinate frame defined by its generating vertex. Procedure **preProcess** (line 19) realizes these actions. To illustrate, we consider the scene vertex **P** of Fig. 3 (a), which depicts a top down view of Fig. 1 (d), as the generating vertex of R. Fig. 3 (b), shows the coordinate frame defined by the generating vertex and the edge points found to be coplanar to the vertex. **preProcess** will output the set of edge points $\mathbf{P_c}$ on the first quadrant of the frame.

Application of a Hough transform -like technique on this set of edge points will determine the rectangle dimensions: The coordinates of the points in $\mathbf{P_c}$ along the $\mathbf{D_x}$ and $\mathbf{D_y}$ axes of the two dimensional vertex coordinate frame are accumulated in two one dimensional arrays $\mathbf{A_x}$ and $\mathbf{A_y}$ respectively (line 20 of Fig. 2). A search procedure for the rectangle dimensions in the accumulators follows: For each pair A_x, A_y of accumulator peaks, we examine the corresponding parameter values M_x and M_y which form an hypothesis about the width and length of the rectangle (see lines $21-24$). We then consider the set of edge points $\mathbf{P_i}$ which contributed to the current peaks A_x and A_y (line 25). The subset $\mathbf{P_f}$ of this set, containing points which belong to the rectangle should have coordinates lower or equal to the parameter values M_x and M_y (line 26). If the number of elements of $\mathbf{P_f}$ is bigger than a user defined threshold p, we regard the rectangle hypothesis to be successfully supported by boundary information and we update its dimension parameters (line $27-28$). A region based verification approach as in line 16 takes the final decision about the validity of the hypothesis (line 29). The advantage of this technique with regard to a standard implementation of the Hough transform is efficiency, since accumulation and search for peaks is performed in one dimensional structures.

Our framework attempts to recover graspable surface boundaries by examining every detected vertex (see line 2 of Fig. 2). This results to the localization of redundant graspable surfaces when more than one vertices per surface have been detected. The procedure invoked in line 9 selects those recovered boundaries which describe the scene in terms of global accuracy and consistency by applying a minimum description length (MDL) approach. The reader is referred to [5] p.122 for implementation details. In addition, independent graspable surface boundary recovery triggered by each detected vertex allows for parallel implementation of the algorithm: A separate processor can be used for dealing with each vertex of the vertex set.

Throughout our analysis we had to test relations of various geometric entities. We had to find out for example whether a two detected vertices are coplanar (look at line 13 of Fig. 2), if the direction vectors of two vertices are parallel (line 13), if an edge point belongs to a plane defined by a detected vertex (line 19), or if the points inside a hypothesized boundary belong to the plane it defines (lines 16, 29). Introduction of empirically defined thresholds for deciding the validity of the relations leads to a non robust system. This problem can be avoided when taking into consideration the error in calculating the geometric entities and statistically testing the geometric relations. If so, all thresholds can be replaced by a unique value, the significance level. We have performed all tests statistically, using the framework in [3], because of its simplicity and compactness. We denote the



(a) Top- down view of configuration (b) Recovered rectangle

Fig. 3. Recovery of graspable surface dimensions from edge points

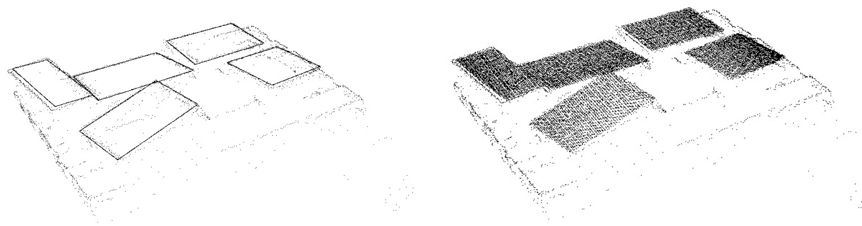
significance level by α in our pseudo code, appearing as input to every procedure where geometric relations are tested (e.g in lines 13,19,16,29).

3 Experimental Results

The output of our algorithm applied on the test case of Fig.1 (a) is given in Fig. 4. Fig. 4 (a) depicts the detected boundaries of the graspable surfaces and Fig. 4 (b) shows the range points inside the detected boundaries, which led to the verification of the particular boundary hypotheses. We have performed a number of experiments of the algorithm using card board boxes. A Pentium 3, $600Mhz$ was used for our experiments. The overall average execution time of the algorithm algorithm was 55 seconds. Edge detection lasted 10 seconds, vertex detection 14 and the dimension recovery about 31 seconds. The average processing time for dimension recovery from a single vertex was 3 seconds. This means that in the event a parallel implementation for object recovery is employed the overall execution time will be less than 30 seconds on the average. In terms of robustness, our experiments demonstrated that the system only occasionally fails to recover all the graspable surfaces in the pile. According to initial accuracy measurements the translational grasping accuracy was less then 1.5 cm, almost equal to the accuracy of the sensor employed. In the future we intend to continue experiments for the system evaluation.

4 Conclusions

We presented a framework for automatic unloading (depalletizing) of piled boxes of unknown dimensions. We employed a laser sensor for data acquisition and detected graspable surfaces of objects from images acquired from the sensor. Major characteristics of our approach is the usage of both boundary and region



(a) Boundaries (b) Points inside boundaries

Fig. 4. Recovered graspable surfaces

based sources of information obtained from the range images and the recovery problem decomposition into subproblems. Experiments demonstrated that our system shows advantages such as computational efficiency and robustness. However, our system as is does not recover the height of the objects it grasps. This is a problem when we want to automatically sort the objects grasped. This problem can be solved by the usage of an additional sensor for measuring the objects' height after their grasping.

References

1. A.J. Baerveldt. *Robust Singulation of Parcels with a Robot System using multiple sensors*. PhDthesis, Swiss federal institute of technology, 1993.
2. 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.
3. 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.
4. M. Hashimoto and K. Sumi. 3d object recognition based on integration of range image and grey-scale image. In *Britisch Machine Vision Conference*, pages 253–262, 1999.
5. A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and recovery of Superquadrics*, volume 20 of Computational imaging and vision. Kluwer Academic Publishers, Dordrecht, 2000.
6. D. Katsoulas. Robust extraction of vertices in range images by constraining the Hough transform. In *ibPRIA 2003, First Iberian Conference on Pattern Recognition and Image Analysis*, LNCS. Springer, June 2003. To appear.
7. S. Kristensen, S. Estable, M. Kossow, and R. Brosel. Bin-picking with a solid state range camera. *Robotics and autonomous systems*, 35(3–4):143–151, June 2001.
8. D. Newcorn. Robot gains eyesight. *Packaging World*, October 1998.
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.
10. R.W. Swanson and L.J. Thayer. A fast shaded-polygon renderer. In David C. Evans and Rusell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, pages 95–101, August 1986.