Fast Recovery of Piled Deformable Objects Using Superquadrics

Dimitrios Katsoulas[1] and Aleš Jaklič[2]

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

[2] Computer Vision Laboratory, University of Ljubljana,
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
ales.jaklic@fri.uni-lj.si

Abstract. Fast robotic unloading of piled deformable box-like objects (e.g. box-like sacks), is undoubtedly of great importance to the industry. Existing systems although fast, can only deal with layered, neatly placed configurations of such objects. In this paper we discuss an approach which deals with both neatly placed and jumbled configurations of objects. We use a time of flight laser sensor mounted on the hand of a robot for data acquisition. Target objects are modeled with globally deformed superquadrics. Object vertices are detected and superquadric seeds are placed at these vertices. Seed refinement via region growing results in accurate object recovery. Our system exhibits a plethora of advantages the most important of which its speed. Experiments demonstrate that our system can be used for object unloading in real time, when a multi-processor computer is employed.

1 Introduction

This paper addresses the depalletizing problem (or robotic bin picking problem) in the context of which a number of objects of arbitrary dimensions, texture and type must be automatically located, grasped and transferred from a pallet (a rectangular platform), on which they reside, to a specific point defined by the user. The need for automated, robust and generic depalletizing systems, stems primarily from the car and food industries. Such systems are of great importance because they undertake a task that is very monotonous, strenuous and sometimes quite dangerous for humans. In this contribution we discuss the automatic recovery of piled deformable box-like objects (see Fig. 3), which are quite often encountered in distribution centers. These kind of objects tend to deform (usually bend) along their longer side, which renders their fast recovery a difficult task.

This is the reason why a multitude of systems aiming at depalletizing of rigid polyhedral objects or boxes have been reported ([1,5,7], etc.), while only a few systems which deal with depalletizing of non rigid objects exist. In [9,6] intensity imagery is employed for the recovery of neatly placed sacks. [9] is based

L. Van Gool (Ed.): DAGM 2002, LNCS 2449, pp. 174–181, 2002.

on the detection of markers on the exposed surface of the objects, [6] on the detection of edges. These approaches although very fast, do not work when the objects are jumbled and inherit the major problem of the intensity based systems, that is, dependency on the lighting conditions. The system described in [6], additionally assumes that the number of objects on the pallet is known. This limits its application range even more. Approaches utilizing range imagery seem more promising. The system described in [4] employs range imagery and aims at the generic recognition of rigid, flat and "irregular" box like objects at the same time. However, no experimental results on the detection of "irregular" objects are shown and no efficiency measurements presented.

Our system employs a time of flight laser range finder mounted on the hand of an industrial robot to depalletize neatly placed and piled box-like objects. A vacuum gripper is used for object grasping. The system employs superquadrics for object modeling and is based on a hypothesis generation and refinement scheme. Object vertices are detected on the images acquired from the laser sensor, and superquadric seeds are placed on the vertices. These seeds are the initial object location hypotheses which are then refined in a region growing manner. In our application it is assumed that the objects contained in each platform are of the same kind and (known) dimensions and are full of material. We further assume that each object's length is longer than their width and both are much longer than their height, so that their largest surface is mainly exposed to the laser source. Despite these assumptions the application range of our system is vast. The advantages of our system are: Independence from lighting conditions since range imagery is used, computational efficiency due to fast region growing, accuracy and robustness due to the high quality initial hypotheses and the constant monitoring of their evolution via region growing, versatility since our system can deal with both jumbled and neatly placed configurations, low cost since a range finder is all we need, and last but not least simplicity.

2 Object Recovery

The problem we are dealing with, belongs to the category of model based range image segmentation problems and as stated in [3], interleaving model recovery with segmentation (segment-and-fit) is one of the best approaches for handling it. We use superquadrics for object modeling, since they exhibit a multitude of advantages. Among them, a small number of parameters with a large expressive power, and a fast method for their recovery from range images. The expressive power of superquadrics (SQs) can be further enhanced by the addition of a couple of global deformation parameters. We use global bending for modeling the bending of our objects along their longest side.

2.1 Object Modeling with Superquadrics

Superquadrics are a family of parametric shapes, the implicit form of which is given by Eq. 1. The parameters a_1, a_2, a_3 express the length of the SQ in the

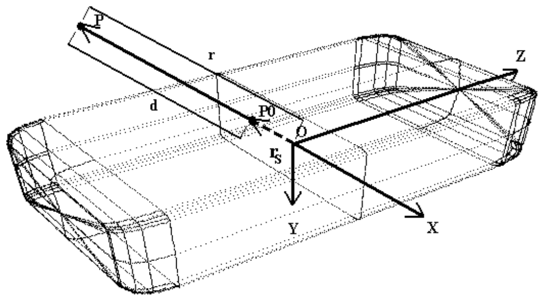


Fig. 1. Radial Euclidean distance from a superquadric

x, y, z axes respectively and ϵ_1, ϵ_2 its shape. The function F in Eq. 1, is called the inside-outside function, because it provides a simple test whether a given point lies inside or outside a SQ. If $F < 1$, then a given point lies inside the SQ. If $F = 1$, the point lies on the surface of the SQ, and if $F > 1$ the point lies outside the SQ.

$$F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} \tag{1}$$

The radial Euclidean distance $|d|$ of a point $P(x, y, z)$ from a SQ (Fig. 1), is defined in terms of the inside-outside function, and is given in Eq. 2.

$$|d| = |\mathbf{r} - \mathbf{r_S}| = |\mathbf{r}||1 - F^{-\frac{\epsilon_1}{2}}(x, y, z)| = |\mathbf{r_S}||F^{\frac{\epsilon_1}{2}}(x, y, z) - 1| \tag{2}$$

We employ two global deformation parameters to model object bending and transform the z(elongated) axis of the SQ into a circular section. These are the curvature of the circular section k and the z axis bending plane angle a (see [3] for details).

Note that Eq.1, expresses a SQ placed at the origin of the model coordinate system. Six more parameters should be incorporated (three for translation and three for rotation) to allow for expressing the inside-outside function in the general position and two more for the global bending. This increases the number of our model parameters to 13. If we denote by $\mathbf{\Lambda}$ the set of model parameters, the inside-outside function takes the form $F(x, y, z; \mathbf{\Lambda})$.

We use the minimization approach of [10] to fit SQs to data points. This method minimizes the expression (3), where n is the number of points to fit.

$$\min_{\mathbf{\Lambda}} \sum_{i=1}^{n} \left(\sqrt{\lambda_1 \lambda_2 \lambda_3} (F^{\epsilon_1}(x_i, y_i, z_i; \lambda_1, \ldots, \lambda_{13}) - 1) \right)^2 . \tag{3}$$

The method is very fast,but has the inherent problem of favoring the recovery of superquadrics which produce larger values for $\mathbf{r_s}$, since as pointed out in [11]:

$$F^{\epsilon_1}(x, y, z) - 1 = \frac{d}{|\mathbf{r_S}|} \left(\frac{d}{|\mathbf{r_S}|} + 2 \right) . \tag{4}$$

2.2 Segmentation Approach

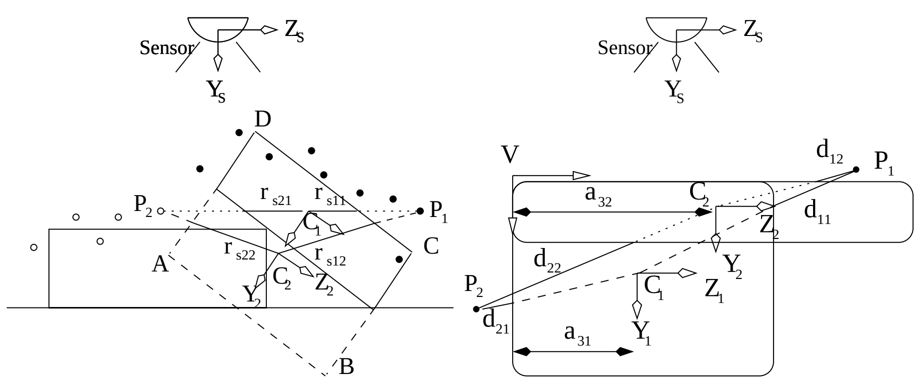
One of the most successful approaches for recovering SQs from range data is presented in [8,3]. The authors attempt recovery of multiple dissimilar occluded objects from range images using SQs, in a segment-and-fit framework. SQ seeds are placed in the image in a grid-like pattern of windows. Each seed encompasses a set of range data points. A SQ is fitted to this data set, and an initial set of SQ parameters is determined. The seeds are then allowed to grow by adding neighboring points to their point data set which have a small radial Euclidean distance from the already recovered models. The growing process continues until the average error of fit of a model to its data points is bigger than a threshold, no more points with a small distance from the recovered model exist (fully grown models), or at least one model reaches twice its original size. In the latter case, a model-selection procedure is initiated. This procedure retains a set of models which are determined to optimally (in the MDL sense) describe the data. In this way non-redundant models which accurately describe the data set are retained. The remaining models are then further grown and the whole process continues until no SQs can grow any more.

This system shows many advantages. Computational efficiency is achieved, because the fast fitting method of [10] is used. Furthermore, the fact that each seed grows independently, allows for parallel implementation of the algorithm. Accurate and robust object recovery is accomplished by the placement of numerous, redundant seeds in combination with the frequent invocations of the model selection procedure which allows only the best models to grow. However, even if the method is quite fast it is not appropriate for a real time implementation, due to the big number of seeds and the fact that the model-selection procedure is non-parallelisable. Algorithm acceleration requires reducing both the number of seeds as well as the number of invocations of the model-selection procedure. But this will inevitably reduce the recovery accuracy.

The solution to the problem lies in the experimental observation in [3], p. 135. It is there reported, that in the event of good quality of initial seeds, the output of the algorithm is equally satisfactory when the model-selection is invoked only once, after all seeds are fully grown. It is as well observed, that in this case no need for initializing many redundant seeds exists any more, since all the models would yield almost the same result at the end. Ideally one reliable seed per model would be enough to recover all the models in the image. This observation is the kernel of our approach. In [2] is stated 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. In [5] this principle has been employed for detecting boxes in piles. Since our objects are box-like, their vertices can still be used for generating accurate object location hypotheses. We therefore detect $3D$ object vertices in the scene, place SQ seeds to those vertices and let them grow to their full size. After all the seeds are fully grown the model-selection procedure is invoked only once and the best descriptions are retained. We thus reduce the number of seeds and minimize the number of model-selection invocations without reducing the recovery accuracy. The high quality vertex seeds result in fast region growing. The minimization of the number of model-selection invocations accelerates the algorithm and renders it fully parallelisable.

Vertex detection and seed placement: $3D$ object vertices are detected in the images in the way presented in [5] which is based on the joint use of edge detection and a technique inspired by the fast dynamic generalized Hough transform. A scene vertex is represented as a triplet which comprise the position of the vertex point in the scene and the two normalized vectors pointing in the direction of the edges joining at the vertex. The vertex is aligned with the xz surface of the superquadric seeds. The alignment procedure is described in detail in [5].

Region growing: We adopt the region growing method of [3,8], a description of which has already been presented. However the method as is has some drawbacks which stem from the fitting function problem expressed by Eq.4. Let's suppose that in the object configuration shown in Fig. 2 (a), the "black" $3D$ range points correspond to the inclined object, while the white ones to the horizontally placed. When using (3) to fit a model to the black points, the model C_2 $(ABCD)$ depicted with dashed line in the figure is recovered, because then the $|\mathbf{r_s}|$'s of all the black points are maximized (see Eq. 4). Actually in Fig. 2 (a) $|\mathbf{r_{s12}}| > |\mathbf{r_{s11}}|$. In the next region growing operation, the point $\mathbf{P_2}$, will be added to the set of data points of model C_2, although it belongs to another object. Due to monotonic region growth, inclusion of outliers in a model's point set deteriorates the performance of the least square fitting and hinders its growth. A solution to the problem could be the restriction of the size of the recovered model along the y axis so as not to surpass its known maximum value. However, it was observed in practice that this reduces the recovery speed. Another issue is that the radial Euclidean distance function equally favors points which belong to the model's growing direction (which is reliably defined by the 3D seed vertex) and others which are probable outliers.

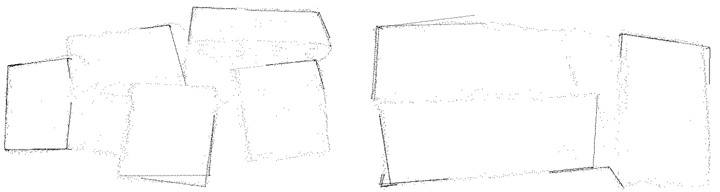


(a) Problem of fitting function. (b) Leading model definition (side view).

Fig. 2.

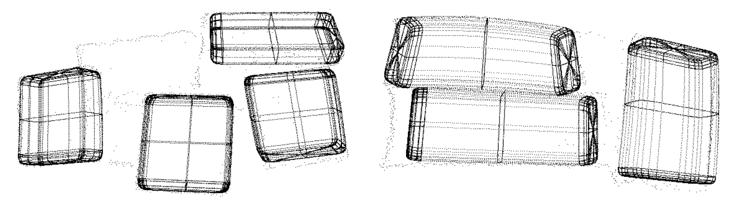
(a) Box like packets (b) Sacks

Fig. 3. Object configurations

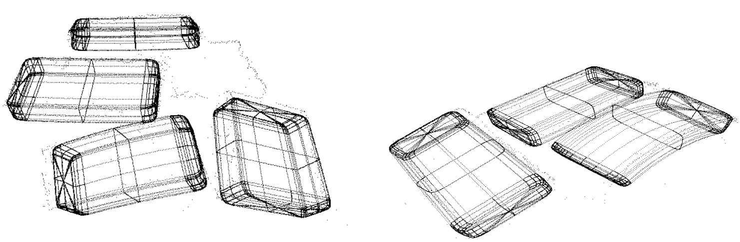


(a) Box - like packets (b) Sacks

Fig. 4. Vertices



(a) Box - like packets (View 1) (b) Sacks (View 1)



(c) Box - like packets (View 2) (d) Sacks (View 2)

Fig. 5. Recovered superquadrics

In order to solve both problems, we exploit the fact that the growing direction is defined by the vertex with which the seed was aligned and thus the position of the inliers is grossly known. We include neighboring range points to a model's point data set only if they are close not to the growing model but to another model which is defined in this way so as to be near the probable inliers and away from the outliers. We name this model *Leading Model*, because it guides the region growing process. This model is aligned to the same vertex of the growing model which was fitted to the scene vertex. Its length along the y axis is kept to a small value, and the length along the x and z axes equals the corresponding lengths of the growing model increased by a user-defined small value. Fig. 2 (b) illustrates. The Leading Model (C_2) is aligned to the vertex V of the original model C_1. C_2 is bigger than C_1 along the z (and x) axis, but much smaller along the y axis. Points which lie in the growing direction (P_1) are favored because they have smaller radial Euclidean distance ($d_{12} < d_{11}$), while others (P_2) which are probably outliers are inhibited ($d_{22} > d_{21}$).

3 Experimental Results

For testing our system we used configurations of objects like the ones depicted in Fig. 3, more specifically box-like packets and sacks. For the configurations in Fig. 3, the detected vertices superimposed to the edge map are depicted in Fig. 4. Two views of the recovered superquadrics on the top of the edge map are presented in Fig. 5, where the reader can get an impression of the recovery accuracy. The time needed to process each of the range images was about 3 minutes while the time required for the recovery of a unique superquadric was about 30 seconds in a pentium III $650MHz$ PC. Since our algorithm is fully parallelisable the latter will approximately be the overall processing time required in a parallel architecture if one processor per seed is used. In terms of robustness, our experiments demonstrated that the system only occasionally fails to find at least one object in the pile.

Our system can as well deal with neatly placed object configurations, since a vertex is defined via only two direction vectors on the edges of the object's exposed surface. However, problems are encountered, if the objects are placed too close one after the other,when no edges and as a result no vertices can be detected. We expect that a sensor with a higher accuracy or an additional sensor could be used to overcome this problem. The alert reader may have already noticed that in Fig. 5 (b) the front sack has not been fully recovered but the length along the z axis is about 90 per cent of the actual length. This is due to the fact that the global bending we have used for modeling can not cover all the ways in which long objects may deform. We plan to solve this problem by introducing a few more global deformation parameters to our model.

4 Conclusions

We described a system for depalletizing piled deformable objects. Globally deformed superquadrics have been used for object modeling. We placed superquadric seeds to object vertices, the refinement of which via a region growing approach resulted in reliable object recovery. In the future we plan to continue experiments with the system so as to retrieve detailed accuracy measurements. In addition, we plan to introduce more deformation parameters to our model so as to be able to describe more accurately all possible target object deformations. Further target is the recovery of piled deformable objects of dissimilar dimensions.

References

1. A.J. Baerveldt. *Robust Singulation of Parcels with a Robot System using multiple sensors*. PhD thesis, 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. A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer Academic Publishers, Dordrecht, 2000.
4. A.C. Kak, A.J. Vayda, K.D. Smith, and C.H. Chen. Recognition strategies for 3-d objects in occluded environment. In T. C. Henderson, editor, *Traditional and Non-Traditional Robotic Sensors*, volume F 63 of *NATO Advanced Research Workshop Series*, pages 365–401. Springer-Verlag, Berlin, 1990.
5. D. Katsoulas, L. Bergen, and L. Tassakos. A versatile depalletizer of boxes based on range imagery. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2002), Washington D.C.*, pages 4313–4319. IEEE Robotics and Automation Society, May 11–15 2002.
6. M. Kavoussanos and A. Pouliezos. Visionary automation of sack handling and emptying. *IEEE Robotics and Automation Magazine*, 2000.
7. S. Kristensen, S. Estable, M. Kossow, and R. Brosel. Bin-picking with a solid state range camera. *Robotics and autonomous systems*, 2001.
8. A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmenting and modelling range data. *IEEE Transactions on PAMI*, 19(11):1289–1295, 1997.
9. D. Newcorn. Robot gains eyesight. *Packaging World*, 1998.
10. F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.
11. P. Whaite and F. P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1038–1049, October 1991.