

Box-like Superquadric Recovery in Range Images by Fusing Region and Boundary Information

Dimitrios Katsoulas and Dimitrios I. Kosmopoulos
 NCSR Demokritos, Institute for Informatics and Telecommunications,
 Computational Intelligence Laboratory, 15310, Aghia Paraskevi, Athens, Greece
 {dkats,dkosmo}@iit.demokritos.gr

Abstract

This work contributes to the robotic bin-picking problem, and more specifically to the problem of localizing piled box-like objects. We employ range imagery, and use box-like Superquadrics for modeling the target objects. Our approach for Superquadric segmentation is an extension of the widespread recover-and-select framework, which employs only region information and therefore suffers from the region over-growing problem. Our approach equally considers both region and boundary-based information for performing the recovery task. Extensive experimentation with a variety of target object configurations demonstrates that it outperforms the recover-and-select framework in terms of both robustness and computational efficiency. Moreover, if implemented in a parallel hardware environment, our approach can operate in real time.

1. Introduction

We address the robotic bin picking, in the context of which a number of objects of arbitrary dimensions, texture and type must be automatically localized and grasped by a robotic hand. Here we deal with recovery of piled deformable box-like objects, as well as with rigid cardboard boxes (fig. 3). Existing industrial systems [7], [3] primarily use intensity imagery, and although fast, do not operate satisfactorily when the objects are jumbled. Furthermore, they heavily depend on lighting conditions. We employ range images from a laser sensor, which is mounted on a 6 DOF robotic arm and moves linearly thus acquiring the top side of the configuration.

Given the input range image, a model-based segmentation strategy is invoked to determine the number and pose of the objects. Superquadrics are used for object modelling. In the next section a description of our modelling elements is given; the drawbacks of the state-of-the-art for superquadric recovery are shown in section 3; our approach is described in detail in

section 4, and experimental results are presented in section 5; finally, section 6 concludes this work.

2. Superquadrics as modeling elements

Superquadrics (SQs) form a family of parametric 3D shapes, defined in (1). Points on the surface of an SQ are obtained by assigning values to the parameters η and ω in the range $[-\pi/2, \pi/2]$ and $[-\pi, \pi]$ respectively.

$$\begin{aligned} x_m(\mathbf{p}; \eta, \omega) &= a_1 \cos(\eta)^{\varepsilon_1} \cos(\omega)^{\varepsilon_2} \\ y_m(\mathbf{p}; \eta, \omega) &= a_2 \cos(\eta)^{\varepsilon_1} \sin(\omega)^{\varepsilon_2} \\ z_m(\mathbf{p}; \eta, \omega) &= a_3 \sin(\eta)^{\varepsilon_1} \end{aligned} \quad (1)$$

where vector \mathbf{p} expresses the SQ model. $\varepsilon_1, \varepsilon_2$ control the shape of the SQ, (for our target objects $0.1 \leq \varepsilon_1, \varepsilon_2 \leq 0.3$). a_1, a_2, a_3 express the size of the SQ along the $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ axes of the model coordinate system. The boundary of the SQ surface on the plane $\mathbf{Z} = -a_3$, models the boundary of the objects' surface which is exposed to the laser sensor. Points on the boundary can be obtained by fixing η to the value η_b defined in (2) [2], which will be hereinafter referred to as *SQ boundary*.

$$\eta_b = \cos^{-1}\left(\frac{a_3}{\sqrt{a_1^2 + a_2^2 + a_3^2}}\right) - \frac{\pi}{2} \quad (2)$$

We add parabolic deformation parameters a, b, c, d to the default SQ parameter vector, to express slight bending of our target objects. If we introduce p_x, p_y, p_z the translation and φ, θ, ψ the rotation angles about the $\mathbf{Z}, \mathbf{Y}, \mathbf{X}$ axes respectively in the general position, then: $\mathbf{p} = (a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, a, b, c, d, \varphi, \theta, \psi, p_x, p_y, p_z)$.

For fitting an SQ model to n range points (x_{si}, y_{si}, z_{si}) , $i = 1..n$, in the sensor coordinate system $\mathbf{X}_s, \mathbf{Y}_s, \mathbf{Z}_s$, the function L_r , shown in eq. (3) is minimized with respect to \mathbf{p} , as in [8], [4]. In this equation, $F(\mathbf{p}; x_s, y_s, z_s)$ is the SQ *inside-outside* function [4].

$$L_r(\mathbf{p}) = \sum_{i=1}^n (\sqrt{a_1 a_2 a_3} (F^{\varepsilon_1}(\mathbf{p}; x_{si}, y_{si}, z_{si}) - 1))^2 \quad (3)$$

2. Recover-and-select for superquadric segmentation

The most widespread approach for recovering multiple SQs from range data, is the recover-and-select paradigm [6], [4]. Given an input range image, image segmentation is performed via maximization of the posterior probability of the segmentation parameters (the models, and their parameters). Firstly, a *hypothesis* about the actual values of the segmentation parameters is generated, then the generated hypothesis is *refined*.

The hypothesis generation stage, performs a rough estimation of the segmentation parameters: Models are placed in the input range image in a grid-like pattern of small cells. In this way each model is associated with some image points (*seed region*,-fig.2c).

The hypothesis refinement stage comprises the *classify-and-fit* and *model selection* processes. The former *locally* refines the parameters of existing models, by interweaving image point classification to models with model parameter estimation. The iterative region growing approach is employed for this purpose [1], [6]. The model selection process updates the *number* of models in the image, by retaining only those models which allow for simple and accurate representation. The two processes are invoking iteratively, until no significant change in the parameters of the remaining models can be performed [6].

The region growing is generally robust against abrupt noise but it is not reliable on the boundaries of the objects (*region over-growing*). This problem is *implicitly* addressed in [4] by: (i) frequently invoking the model selection process, which rejects models crossing object boundaries, and (ii) initializing a large number of seeds inside the object boundaries, so that despite rejections, enough models are left available (inside the objects) for further growing. However, we have seen in our tests that the model overgrowing problem is still present (fig. 2d).

3. Box-like SQ recovery

Our approach for SQ segmentation, *explicitly* addresses the region overgrowing problem by employing except of the region information, object *boundary* information into the recovery process.

Boundary information is generated by applying edge detection to the range image I . We employ a 2D edge map I_b for deriving boundary information, a pixel \mathbf{x} of which is set to one, if the corresponding range point $I(\mathbf{x})$ is an object boundary point.

As the recover-and-select strategy, segmentation comprises two stages: Hypothesis generation, and hypothesis refinement. Boundary information is taken into consideration in both stages of the framework. In the hypothesis refinement stage in particular, information integration is inspired by a game theoretic

framework: Parameter refinement is realized by means of iterative invocations of two independent parametric modules in succession, which provides robustness and efficiency [2]: A region module, fits SQ models to the range image, assumed they belong to unique objects. A boundary module fits the boundary of models to the edge map.

4.1 Hypothesis generation

In this stage seed placement is performed via formation of closed contours in the edge map. This is realized by employing the model-based adaptive contour closure approach of [5]. The region of range points contained within each closed contour, which can be accurately modeled by a boxlike SQ model, is considered a seed. Usage of boundary information for seed placement, guarantees that seeds do not cross object boundaries on one hand and generates one seed per exposed object on the other (fig. 2g). (fig.2h) shows the SQ models fitted to the seed points and (fig.2i), shows the boundaries of the initialized models embedded in the edge map.

4.2 Hypothesis refinement

Given the parameter vectors of all models initialized in the previous stage, along with the range image I and the edge map I_b , the hypothesis refinement stage deals with the recovery of the parameters of each model independently. As in the recover-and-select strategy a classify-and-fit, and a model selection process are employed for model parameter refinement, but the implementation differs.

The iterative classify-and-fit process integrates boundary and region information to perform the recovery task: Each iteration of this process involves invocation of (a) The boundary module, which recovers the model boundary utilizing boundary information of the edge map I_b . (b) The region module, which recovers the model parameters using the range image I . Within each of the modules, model parameter recovery is interwoven with image pixel classification (each of the modules acts as the standard classify-and-fit process on its own information domain). The process iterates until the parameter vector does not change significantly.

Decoupled handling of boundary and region-based information leads to a decoupled framework for superquadric parameter recovery. Boundary information of the 2D edge map constrains the pose and the dimensions of the boundary of the model's exposed surface on the image plane, expressed through subset $\mathbf{p}_b = (a_1, a_2, p_s, p_s, \psi)$. Region information constrains the subset $\mathbf{p}_r = (p_s, \varphi, \theta, a, b, c, d)$. Hence, the boundary and region modules respectively update

\mathbf{p}_b and \mathbf{p}_r only, instead of the entire model parameter set. The rest of parameters $\mathbf{p}_s = (\varepsilon_1, \varepsilon_2, a_3)$ are kept constant within the classify-and-fit process.

Model selection is invoked only once after the end of the classify-and-fit process. Incorporation of boundary information abolishes the need of embedding model-selection into classify-and-fit and invoking it frequently, unlike in the recover-and-select framework.

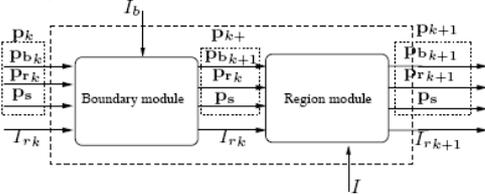


Figure 1. The k-th iteration of the classify and fit process

Boundary module: Boundary finding influenced by region information. The module inputs are the edge map I_b , the vector \mathbf{p}_{bk} (from previous cycle), the model parameter subset \mathbf{p}_{rk} , the \mathbf{p}_s (constant), as well as the *region image* I_{rk} (2D binary image of equal size to the range image, with corresponding pixels belonging to object set to 1). Output is the updated SQ model parameter subset \mathbf{p}_{bk+1} , (fig. 1). The $\mathbf{p}_k = (\mathbf{p}_s, \mathbf{p}_{rk}, \mathbf{p}_{bk})$ denotes the model parameter set before the k th cycle.

Refinement of \mathbf{p}_k here outputs \mathbf{p}_{k+1} and is inspired by [9], [2], where we maximize the function L shown in (4), ($\mu_1, \mu_2 = 0.5$ are weight constants), with respect to \mathbf{p}_{bk} , as in eq. (5).

$$L(\mathbf{p}_k, I_b, I_{rk}) = \mu_1 L_b(\mathbf{p}_k, I_b) + \mu_2 L_r(\mathbf{p}_k, I_{rk}) \quad (4)$$

$$\mathbf{p}_{bk+1} = \arg \max_{\mathbf{p}_{bk}} L(\mathbf{p}_k, I_b, I_{rk}) \quad (5)$$

L_b is given in (6) and its maximization with respect to \mathbf{p}_{bk} is equivalent to fitting the model's boundary of exposed surface to the edge map. It is proportional to the sum of squares of Euclidean distances of points on the model boundary, embedded in the image plane, from their closest edge points. This is expressed in (6), where I_{bd} denotes the Euclidean distance transformed image of the edge map I_b . Besides, $(x(\mathbf{p}_k; \eta_b, \omega_i), y(\mathbf{p}_k; \eta_b, \omega_i))$, $i = 1 \dots M$, $\omega_i = i\pi/M$ express the image coordinates of three dimensional points on the boundary of the exposed surface of the model \mathbf{p}_k , given by (1), where η_b is given by eq. (2).

$$L_b(\mathbf{p}, I_b) = - \sum_{i=1}^M I_{bd}(x(\mathbf{p}_k; \eta_b, \omega_i), y(\mathbf{p}_k; \eta_b, \omega_i))^2 \quad (6)$$

L_r is given in (7) and its maximization is equivalent to fitting the model's boundary to the boundary of the region of pixels set to 1 in the region image I_{rk} . I_{rbd} denotes the Euclidean distance transformed image of the boundary of the region of pixels set to 1 in I_{rk} .

$$L_r(\mathbf{p}, I_{rk}) = - \sum_{i=1}^M I_{rbd}(x(\mathbf{p}_k; \eta_b, \omega_i), y(\mathbf{p}_k; \eta_b, \omega_i))^2 \quad (7)$$

Region module: Region growing influenced by boundary information Inputs are the range image I , the model \mathbf{p}_{rk} , and the region image I_{rk} (from previous cycle) and the \mathbf{p}_{bk+1} . Outputs are the \mathbf{p}_{rk+1} , and I_{rk+1} (fig. 1). The vector $\mathbf{p}_{k+1} = (\mathbf{p}_s, \mathbf{p}_{rk+1}, \mathbf{p}_{bk+1})$, denotes the SQ model, as is before the k th cycle.

The region module performs an iterative region growing process of two steps: The *model fitting step* estimates the parameter vector \mathbf{p}_{rk+1} , by maximizing (3) with respect to \mathbf{p}_{rk} , as in (8) (the image coordinates of the range points corresponding to the object of interest are enclosed by the boundary of the model \mathbf{p}_{k+1}).

$$\mathbf{p}_{rk+1} = \arg \max_{\mathbf{p}_{rk}} L_r(\mathbf{p}_{k+1}) \quad (8)$$

In the *classification step* the SQ model $\mathbf{p}_{k+1} = (\mathbf{p}_s, \mathbf{p}_{rk+1}, \mathbf{p}_{bk+1})$ (after the fitting step), and the region image I_{rk} are used to generate the updated region image I_{rk+1} . The pixels in the neighbourhood of the boundary of the region in the region image, which have a small distance from the model \mathbf{p}_{k+1} , are added in the region of I_{rk} to obtain I_{rk+1} .

Post processing, determines if the recovered model corresponds to a *graspable* object. Such an object is characterized by the full exposure of its largest surface and thus its boundary. The fitting residual error regarding the SQ and the boundary have to be small otherwise the object is rejected.

5. Experiments

We have tested the performance of our recovery approach using 40 range images corresponding to configurations containing sacks, box-like pillows, box-like objects wrapped in transparent foil, and card-board boxes. Representative results are shown in (fig.3).

For 174 graspable objects in the images, we had 159 true positives, 6 false positives and 15 false negatives. The system failures occur when no boundary information exists. This happens when objects of similar dimensions are uniformly placed in layers, and the distance between them is very small (less than 1 cm), since then the sensor resolution is not high enough to capture boundary information.

To assess accuracy, we *manually* isolated the regions corresponding to the graspable objects in the images. We then measured the average Euclidean distances of these points to the corresponding recovered models, which was less than 1cm per model on the average. Besides, we measured the average Euclidean distance between the embedded boundary of the exposed surfaces of the recovered models in the image plane,

and the actual boundary of the exposed surfaces of the objects, which was about 1 pixel/object on the average. The average time per image needed for the recovery of all objects in the image, was about 91 seconds on a Pentium 4 2,8GHz PC (using Matlab optimization functions), while the recover-and-select framework requires about 15 minutes for this task (using a C compiler). The average recovery time per object, which will be the time required for the complete process if our system is implemented in a parallel architecture (where the each model is assigned a processor) was about 15 seconds.

6. Conclusions and future work

We presented an approach for SQ recovery in range images of box-like objects. This approach outperforms the state-of-the-art in superquadric segmentation, in terms of all computational efficiency, accuracy and robustness, and has been employed in a novel robotic system for automatic box-like object unloading. Incorporation of boundary information in the segmentation process has led to high computational efficiency and robustness.

Our system, is able to recover box-like superquadrics. However, it can be easily extended to recover all kinds of superquadrics, by (i) incorporating all parameters kept constant here, in the optimization processes, and

(ii) using the superquadric rim defined in [4] instead of the SQ boundary for boundary finding. Our initial experiments in this direction are very encouraging.

References

- [1] P. Besl and R. Jain. Segmentation Through Variable-Order Surface Fitting. *IEEE T-PAMI*, 9(2):167–192, 1988.
- [2] A. Chackraborty, J. Duncan. Game Theoretic Integration for Image Segmentation. *IEEE T-PAMI*, 21(1):12–30, 1999.
- [3] M. Hashimoto and K. Sumi. 3D Object Recognition Based on Integration of Range Image and Grey-scale Image. In *British Machine Vision Conference*, pages 253–262, 1999.
- [4] A. Jaklic, A. Leonardis, and F. Solina. *Segmentation and recovery of Superquadrics*, volume 20 of *Computational Imaging and Vision.*, Kluwer, Dordrecht, 2000.
- [5] X. Jiang. An Adaptive Contour Closure Algorithm and its Experimental Evaluation. *IEEE T-PAMI*, 22(11):1252–1265, 2000.
- [6] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for Segmenting and Modeling Range data. *IEEE T-PAMI*, 19(11):1289–1295, 1997.
- [7] D. Newcorn. Robot gains eyesight. *Packaging World*, October 1998.
- [8] F. Solina and R. Bajcsy. Recovery of Parametric Models from Range Images: The case for Superquadrics with Global Deformations. *IEEE T-PAMI*, 12(2):131–147, 1990.
- [9] L. Staib and J. Duncan. Boundary Finding with Parametrically Deformable Models. *IEEE T-PAMI*, 14(11):1061–1075, Nov. 1992.

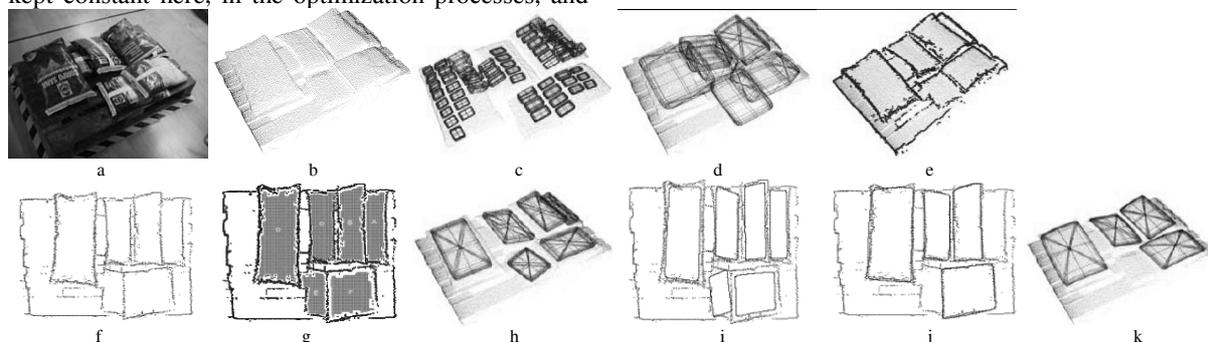


Figure 2 Segmentation comparison. The input data: (a) intensity image (b) range image I . The recover and select method: (c) the placed seeds (d) the final results. Our method: (e) edge image I_e superimposed on I (f) edge map I_b (g) seeds on image plane, (h) initial models based on seeds (i) model boundaries on image plane (j) recovered boundaries (k) recovered 3D models

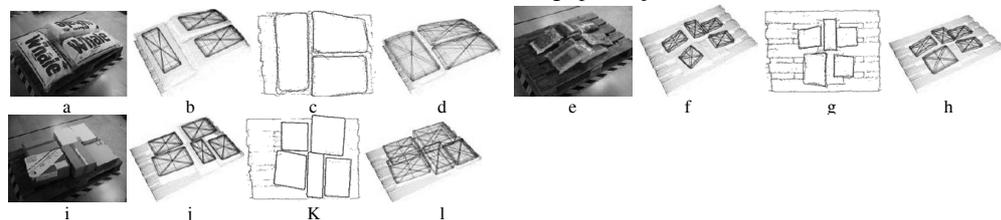


Figure 3 Recovery of (a-d) piled pillows, (e-h) Bags (i-l) cardboard boxes. For each type we illustrate the intensity image, the seed placement, the recovered boundaries and the recovered models.