Reliable Recovery of Piled Box-like Objects via Parabolically Deformable Superquadrics

Dimitrios Katsoulas
University of Freiburg, Institute for Pattern Recognition and Image Processing
George Koehler Allee, 052 Freiburg, Germany
dkats@informatik.uni-freiburg.de

Abstract

Automatic unloading of piled box-like objects is undoubtedly of great importance to the industry. In this contribution a system addressing this problem is described: We employ a laser range finder for data acquisition, and globally deformable Superquadrics [2], [22] for object modeling. Our technique is based on a hypothesis generation and refinement scheme. The vertices of the piled objects are extracted and Superquadric seeds are aligned at these vertices. The model parameter recovery task is decomposed into two subproblems, each dealing with a subset of the model's parameter set. Both region and boundary based information sources are used for parameter estimation. Compared to a widespread strategy for superquadric recovery [11], our method shows advantages in terms of robustness and computational efficiency. In addition, our system exhibits versatility with regard to existing industrial systems, since it can effectively deal with both neatly placed and jumbled configurations of objects

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 place 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 (a), Fig. 7 (a), Fig. 8 (a)), which are quite often encountered in distribution centers.

The majority of existing systems [18], [14], [10] employ primarily intensity imagery and recover objects, either by detecting markers on their exposed surfaces, or by template matching. These approaches although very fast, do not work satisfactory when the objects are jumbled and inherit the major problem of the intensity based systems, that is, dependency on lighting conditions. Our system employs a time of flight laser range finder mounted on the hand of an industrial robot to unload neatly placed and piled box-like objects. A vacuum gripper is used for grasping the objects from their exposed surfaces. Deformable superquadrics are employed for modeling the target objects. In the following, we assume that the objects lying on the pallets are full of material so that they cannot arbitrarily deform and their dimensions are known. If so, our system exhibits many advantages comparing to existing approaches: Independence from lighting conditions since range imagery is used, versatility since our system can deal with both jumbled and neatly placed configurations, robustness and computational efficiency. The paper continues as follows: An overview of the most frequently employed for recovering multiple models in images is attempted in section 2.1. A coarse outline of our approach, which combines the advantages of these frameworks, is presented. In section 2.2, we introduce our modeling elements (superquadrics). Section 2.3 presents a detailed outline of our approach and explains how it improves a widespread framework specifically designed for superquadric recovery from range images. The main components of our system are presented in section 2.4. Finally, section 3 shows experimental results and discusses our system's drawbacks.

2. Object recovery

Our problem belongs to a specific category of image segmentation problems which aims at recovering multiple parametric models. A solution to this problem involves both segmentation and model parameter estimation. In re-

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

cent years, it has become clear in the computer vision community that approaches in which segmentation and parameter estimation are coupled from the early steps of the process, rather then being isolated, give superior results [11]. Such approaches could be broadly subdivided into two categories: Methods inspired from the Hough Transform and methods based on iterative seed evolution.

2.1. Finding multiple parametric models in images

Techniques stemming from the Standard Hough Transform (SHT) are frequently used for parameter estimation. The major advantage of the method is that no initialization of the sought model parameters is required. However, the SHT shows drawbacks: It is memory consuming, computationally inefficient and sensitive to errors in localization of image points. Variations of the Hough transform have been proposed to improve its performance and robustness [16], [20]. Unfortunately, all implementations break down when the number of model parameters moderately increases, which is bound to happen when more complicated models or parametric model deformations are considered. In the event, techniques based on iterative seed evolution framework of the subsequent paragraph seem to be the unique alternative.

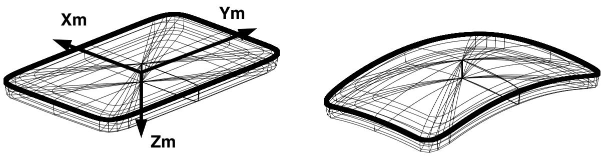
A major group of algorithms extract multiple parametric models by initializing a large number of models in the image, namely the seeds. Each seed can be considered as an hypothesis of the parameters of a scene object. Seed evolution based on image data results either to successful object recovery via hypothesis refinement, or to hypothesis rejection. Seed evolution is realized by iterative updating of the seed parameter vector via minimization of a cost function measuring the dissimilarity of the seed to the image. Seed interaction (e.g. seed merging, splitting) or selection is controlled by a process interleaved with seed evolution, which favors models describing the entire image in an optimal way mainly in terms of accuracy, consistency and simplicity. There exist a variety of techniques originating from the general framework described, the differences of which lie mainly in the domain on which the cost functions are defined on one hand, and in the degree of incorporation of global shape information in the evolving model on the other.

Cost functions can be devised so that boundary-based information is taken into consideration [8], [19], [24], region-based information [3], [17] or both [21],[27], [6]. The latter, that is techniques combining both approaches are superior in terms of robustness: Boundary image points inhibit the evolving seeds from crossing object boundaries. Region properties are taken into consideration in the model parameter estimation. Additionally, incorporation of global (or prior) shape information in the parametric models (as in [26], [24], [6]) reduces the number of parameters needed for expressing the model and results in faster fitting to image data.

However, methods based on iterative seed parameter refinement tend to fail when the seed parameters are not initialized near their optimum values. As already mentioned, the Hough transform based techniques do not suffer from this problem. Thereby, in our recovery approach the Hough transform is used to initialize the seeds in the image, the parameters of which are then refined via an iterative seed parameter updating framework. Model parameter refinement utilizes both boundary and region-based image information, which results in system robustness. Globally deformed superquadrics are used as models. Their small number of parameters results in fast fast parameter estimation. This renders our system computationally efficient.

2.2. Parabolically deformable superquadrics

There is a multitude of parametric models incorporating global shape information, commonly employed in the computer vision literature for object recovery from images: generalized cylinders [5], implicit polynomials [15], Geons [4], spherical harmonic and Fourier surfaces [23], [24] and others. Among them, Superquadrics [2], [22], [11], whose shape can vary from an ellipsoid to rectangular parallelepiped, are perhaps the most popular mainly for four reasons: Firstly they comprise a small number of model parameters with large expressive power, so that they can describe many shapes. Secondly, their expressiveness can be further enhanced by the addition of a couple of global deformation parameters. Thirdly, fitting of Superquadrics to $3D$ data is a problem which has been throughly investigated. Robust and fast fitting methods have been developed. Finally, all parameters of the superquadrics have an intuitive meaning, which makes their handling straightforward. Additionally, by constraining their shape parameters these models can be made to represent our symmetric box-like objects in a remarkable way. The combination of those advantages can not be found in other geometric entities. This led us to naturally adopt them as modeling elements for our application.



(a) Undeformed SQ (b) Parabolically deformed SQ

Figure 1. Box-like superquadrics

Superquadrics (SQs) form a family of parametric shapes, defined in (1). Fig. 1 (a) depicts a superquadric with its center of gravity at the origin of the model coordinate frame.

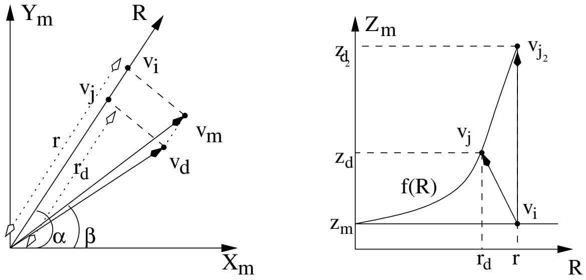
Points on the surface of an SQ are obtained by assigning values to the parameters η and ω in the range $[-\pi/2, \pi/2]$ and $[\pi, \pi]$ respectively.

$$\begin{aligned} x_m(\mathbf{p}; \eta, \omega) &= a_1 \operatorname{sign}(\cos(\eta)\cos(\omega))|\cos(\eta)^{\epsilon_1}\cos(\omega)^{\epsilon_2}| \\ y_m(\mathbf{p}; \eta, \omega) &= a_2 \operatorname{sign}(\cos(\eta)\sin(\omega))|\cos(\eta)^{\epsilon_1}\sin(\omega)^{\epsilon_2}| \\ z_m(\mathbf{p}; \eta, \omega) &= a_3 \operatorname{sign}(\sin(\eta))|\sin(\eta)^{\epsilon_1}| \end{aligned} \tag{1}$$

The vector $\mathbf{p}$ incorporates the SQ parameters: ϵ_1, ϵ_2 control the shape of the SQ, which takes the form of our target objects when $0.1 \leq \epsilon_1, \epsilon_2 \leq 0.5$. a_1, a_2, a_3 express the size of the SQ along the $\mathbf{X_m}, \mathbf{Y_m}, \mathbf{Z_m}$ axes. As will become clear in later sections, the boundary of a SQ surface plays an important role in our segmentation framework. Points on the boundary of the surface contained in the plane $\mathbf{Z_m} = -a_3$ can be obtained by fixing η to the value η_b defined in (2). This boundary is shown in bold in Fig. 1 (a),(b).

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

We add deformation parameters to the default SQ parameter vector, to express slight bending of our target objects. Superquadric global bending along a circular section by introducing two additional deformation parameters is the most widespread [22], [11]. However such kind of bending does not agree with the deformations occurring to our objects. Parabolic deformations output objects with greater similarities to the real deformations happening to the piled target objects on one hand (as illustrated in Fig. 1 (b)), and results into considerably smaller fitting error residuals than both those obtained from undeformed or from circularly bended models on the other.



(a) Projection on the bending plane (b) Parabolic deformation

Figure 2. SQ global deformations

The implementation of parabolic deformations is analogous to the technique presented in [22]. We define a plane perpendicular to the $\mathbf{Z_m} = 0$ plane of the model coordinate system crossing its origin, the bending plane, which forms an angle α with the $\mathbf{X_m}$ axis (see Fig. 2 (a)). The intersection of the bending plane with the $\mathbf{Z_m} = 0$ plane define the bending axis $\mathbf{R}$. Three steps comprise the deformation process which transforms a point $\mathbf{v_m}(x_m, y_m, z_m)$ on the surface of a non-deformed SQ to the corresponding point $\mathbf{v_d}(x_d, y_d, z_d)$ of the deformed SQ: Firstly $\mathbf{v_m}(x_m, y_m, z_m)$ is projected to the bending plane so that point $\mathbf{v_i}(r(x_m, y_m), z_m)$ is acquired (Fig. 2 (a)), the coordinates of which are now expressed in the bending plane coordinate system. The quantity $r(x_m, y_m)$ is computed in (3)-(4).

$$\beta = atan2(y_m, x_m) \tag{3}$$

$$r = cos(\alpha - \beta)\sqrt{x_m^2 + y_m^2} \tag{4}$$

Secondly the deformation on the bending plane is performed: The SQ bends in such a way so that the point $\mathbf{v_i}$ transforms to $\mathbf{v_j}(r_d, z_d)$. This is done via a user defined deformation function $f(R)$, a parabola in our case, so that $z_d = z_m + f(r_d)$ (Fig. 2 (b)). Finally, the point $\mathbf{v_j}$ is back projected to a plane parallel to the bending plane passing from the original point $\mathbf{v_m}$ to obtain the final point $\mathbf{v_d}$ (see Fig. 2 (a)).

We introduce two deformation parameters: The parameter b, which expresses the curvature of the parabola at its vertex c, as well as the position of the vertex. These two along with the angle of the bending plane α comprise the set of deformation parameters. The overall deformation mechanism for recovering the coordinates of the final point $\mathbf{v_d}(x_d, y_d, z_d)$ given the initial point $\mathbf{v}(x_m, y_m, z_m)$ is expressed by the set of equations (5).

$$\begin{aligned} x_d &= x_m + (r_d - r)cos(\alpha) \\ y_d &= y_m + (r_d - r)sin(\alpha) \\ z_d &= z_m + f(r_d) = z_m + b(r_d - c)^2 \end{aligned} \tag{5}$$

Ideally, the deformation process should be defined so that the superquadric spine on which the point $\mathbf{v_m}$ belongs bends to a parabolic segment of the same length (see Fig. 2 (b). This implies that the abscissa r_d of $\mathbf{v_j}$ should be such so that the length of the parabolic arc ending at $\mathbf{v_j}$ is equal to r. Therefore, the desired quantity r_d should satisfy (6), which cannot be analytically solved thus introduces inefficiency in the parameter recovery process.

$$\int_0^{r_d} \sqrt{1 + f'(R)^2}\, dR = \int_0^{r_d} \sqrt{1 + (2b(R-c))^2}\, dR = r \tag{6}$$

We can avoid this problem by considering that since our objects are full of material, the parameter b is expected to be close to zero. Actually, the experimentally observed value of b is in the order of 10^{-3}, which allows us to set $r = r_d$. Consequently, the deformation process brings the point $\mathbf{v_i}$ to $\mathbf{v_{j2}}$ (see Fig. 2 (b)).

Finally, we introduce an additional parabolic deformation along the axis $\mathbf{X_m}$ controlled by the parameter d, with its vertex lying at the origin of the axis, to express slight bending occurring simultaneously with the bending along the arbitrary axis $\mathbf{R}$. Following the same procedure, we end up with a simple set of equations (7)-(9) describing the overall global bending deformations.

$$x_d = x_m \tag{7}$$

$$y_d = y_m \tag{8}$$

$$z_d = z_m + b(r_d - c)^2 + dx_m^2 \tag{9}$$

The parameter set of our model $\mathbf{p}$ comprises nine elements, that is five scale and shape as well as four global deformation parameters. Six more model parameters are introduced to allow for model representation in a general position of the global (world) coordinate system. If we denote as p_x, p_y, p_z the translation and ϕ, θ, ψ the rotation angles about the $\mathbf{Z}, \mathbf{Y}, \mathbf{Z}$ axes respectively, the parameter set of our model becomes: $\mathbf{p}(a_1, a_2, a_3, \epsilon_1, \epsilon_2, \alpha, b, c, d, \phi, \theta, \psi, p_x, p_y, p_z)$. A point on the surface of a deformed SQ in the world coordinate system $\mathbf{v_w}(x_w(\mathbf{p};\eta,\omega), y_w(\mathbf{p};\eta,\omega), z_w(\mathbf{p};\eta,\omega))$ originating from the point $\mathbf{v_m}$ on the undeformed SQ expressed in the model coordinate system is generated by first deforming then rotating and then translating $\mathbf{v_m}$.

The implicit from of an SQ which can be obtained from (1) via simple algebraic manipulations, is given by (10).

$$\left(\left(\frac{x_m}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y_m}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z_m}{a_3}\right)^{\frac{2}{\epsilon_1}} = 1 \tag{10}$$

For fitting an SQ model to n range points $(x_i, y_i, z_i), i = 1..n$, in the world coordinate system, an initial estimate of the parameter vector $\mathbf{p}$ is required. The range points are then brought back from the world coordinate system to the model coordinate system using $\mathbf{p}$. The fitting procedure assuming points corrupted by independent Gaussian noise, updates the vector $\mathbf{p}$ by minimizing $L_r(\mathbf{p})$ of (11), with regard to $\mathbf{p}$. $F(\mathbf{p}; x, y, z)$ is the SQ inside-outside function which equals the left part of (10) (see [22], [11] for details).

$$L_r(\mathbf{p}) = \sum_{i=1}^{n} \left(\sqrt{a_1 a_2 a_3}(F^{\epsilon_1}(\mathbf{p}; x_i, y_i, z_i) - 1)\right)^2. \tag{11}$$

2.3. Recovery of multiple superquadrics

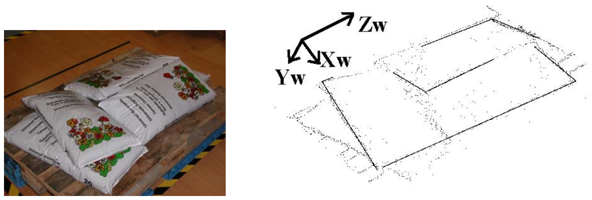
One of the most successful approaches for recovering multiple SQs from range data is described in [11], and is based on the Recover and Select segmentation framework originally presented in [17]. This technique belongs to the general category of iterative seed evolution methods utilizing region-based information. Seeds evolve through region growing. An inherent problem of the region growing based approaches, is their weakness in classifying points on the object boundaries (model overgrowing), which negatively affects the system's accuracy and robustness. In order to address this problem, the authors initialize the system by placing numerous, small seeds in the image. In addition, frequent invocations of a model selection procedure based on the Minimum Description Length (MDL) principle, allows growing of the seeds which do not cross object boundaries. However initializing and processing many seeds, results into computational inefficiency. There is thereby a clear trade-off between reliability and computational efficiency in the approach.

The model overgrowing problem can be overcome if boundary based image information is taken into consideration in the evolution process. This kind of information, can be the outcome of the application of an edge detector to the image. If attraction forces are associated to the detected edge points, and if the seeds are placed near the boundaries of the scene objects, evolution under the influence of the object boundaries' forces will ultimately lead to seed alignment with the detected object boundaries. In such case no model overgrowing will occur. Thereby, it is of extreme importance to ensure that the initial seeds are placed close to the detected object boundaries, or in other words, to guarantee accurate estimation of the seeds' pose parameters, prior to starting the evolution process. It is known since years in the computer vision community [7] that a visible vertex of box-like objects provides the strongest constraints for accurate determining their pose parameters. Since our target objects are box-like, detection of vertices of the piled objects and seed alignment to the vertices, will result in the generation of seeds with accurate pose parameters, as desired. In addition, this leads to a significant reduction of the overall number of seeds: The objects usually expose one of their surfaces (the largest one) to the sensor, which comprises no more than four vertices. A limited number of seeds is therefore associated with each object in the scene.

Based on the observations presented above, our technique for recovering box-like superquadrics in images comprises the following steps: $(\mathbf{i})$: Edge detection in the range image is performed, and the edge map is acquired. The adopted edge detector [12] is based on a top-down detection technique and as such compensates to some extend the bottom-up, data driven nature of the overall recovery approach. $(\mathbf{ii})$: The vertex detection process is initiated: Three-dimensional line segments representing the object boundaries are detected by means of a variation of the Hough Transform technique [20]. All possible pairs of detected lines are considered and orthogonal pairs are inserted to the set of the detected vertices. Vertex detection is described in detail in [13]. $(\mathbf{iii})$: Surfaces of superquadric seeds are aligned to the detected vertices. $(\mathbf{iv})$: Seed evolution takes place, in the context of which the model parameter recovery problem is decomposed into two subproblems:

Recovery of the boundary pose of the SQ seed aligned to the vertex in two dimensions, followed by fitting of the three-dimensional points inside the boundary. The former utilizes edge based information, the latter region based. Each problem recovers a subset of the seed parameter vector. $(\mathbf{v})$ After the end of the seed evolution a model selection procedure [11] is invoked, to ensure global consistency by retaining the optimal models.

This approach shows many advantages: Accuracy and robustness are the outcomes of effectively dealing with the model overgrowing problem. In addition, model selection ensures the output of the most reliable models. Computational efficiency is primarily earned because of the small number of initialized seeds, and by decomposing the fitting problem into subproblems. Moreover, model fitting through minimization of cost functions like (11), (13), the derivatives of which with respect to the parameters can be analytically calculated contributes to the system's efficiency. Note, that independent seed evolution allows for parallel implementation of our technique. If one processor per seed is assigned our system can be used for object recovery in real time.



(a) Intensity image (b) Detected edges-vertices

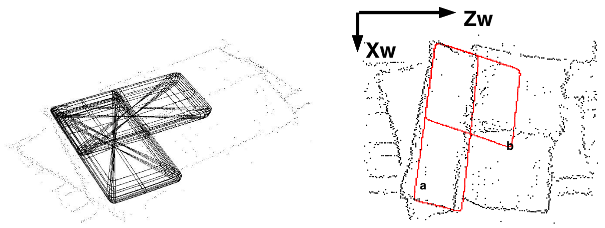
Figure 3. Sacks

2.4. Decomposing the problem of superquadric recovery

Input of the system is a range image acquired by a laser range finder. Edge detection and vertex extraction to the range image corresponding to the intensity image of Fig. 3 (a) is depicted in Fig. 3 (b). The sensor (world) coordinate frame is attached to the image, defined by the axes $\mathbf{X_w}$, $\mathbf{Y_w}$, $\mathbf{Z_w}$. The x_w,z_w coordinates of a range point (x_w, y_w, z_w) express its position on the two-dimensional image plane and y_w expresses its depth value. All points acquired by the laser sensor are points which either belong to objects or the background, The range points corresponding to an object lie inside the boundaries of the object's surfaces the vertices of which are detected (exposed surfaces). This suggests a straightforward solution to the recovery problem consisting of two steps: The determination of the position of those surfaces' boundaries in the image and the model parameter recovery from the range points surrounded by the boundaries.

The boundary pose is recovered using the edge map. Since we do not know which edge points belong to the boundaries, the boundary position can only be determined by assigning attraction fields to each edge point, initializing the boundary parameters, and recover them by minimizing a function expressing the boundary's energy. The fact that we are interested in finding points inside the boundary suggests dealing with the problem of boundary finding in two dimensions. The boundary curve of an aligned seed's exposed surface, is a parametric curve depending on the parameter vector of the superquadric model $\mathbf{p}$ (defined in section 2.2). Not all the elements of $\mathbf{p}$ need to take part in the curve evolution process, but only those defining its location and orientation of the boundary on the image plane, namely the translation parameters along the $\mathbf{X_w}$ and $\mathbf{Z_w}$ axes, (p_x and p_z respectively) as well as the rotation on the image plane expressed through the parameter ψ. We consider three subcomponents of the parameter vector $\mathbf{p}$, namely: $\mathbf{p_{pr}}(a_1, a_2, a_3, \epsilon_1, \epsilon_2)$, $\mathbf{p_e}(p_x, p_z, \psi)$ and $\mathbf{p_r}(p_y, \phi, \theta, \alpha, b, c, d)$. $\mathbf{p_e}$ contains parameters recovered by the $2D$ boundary finding process. Range points inside boundary are used to recover $\mathbf{p_r}$. Finally, the size and shape parameter vector $\mathbf{p_{pr}}$ remains constant during the recovery process, and its parameters are determined off-line. The paragraphs that follow give a more detailed description of the internals of our approach.

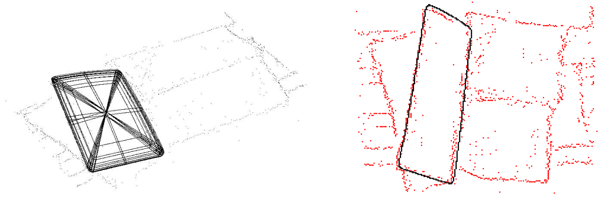
Model training. The purpose of this initial phase which is executed off-line is dual: The determination of the shape and size model parameters one hand, and the estimation of the statistical properties of the deformation parameters on the other. For each type of object expected to appear on the platform, we consider the possible graspable surfaces. These are surfaces with an area large enough allowing for object grasping. Note that in most of the cases our rectangular shaped objects expose only one graspable surface. We place one object of each type alone on the pallet so that the surface is shown to the sensor, remove the background and fit an SQ model to the points belonging on the surface, using (11), with the constraint $0.1 \leq \epsilon_1, \epsilon_2 \leq 0.5$. If the residual is above a threshold (which is usually not the case when the objects are box-like and full of material) the object cannot be handled by our application. Otherwise, the estimated size and shape parameters are inserted to $\mathbf{p_{pr}}$. In all optimization procedures subsequently described, $\mathbf{p_{pr}}$ will remain constant. The object is then placed on the pallet along with other objects so that it bends in various ways, always exposing the particular surface to the laser source. The images obtained are manually segmented, an SQ model is fitted to the points belonging to the surface of the object and deformation parameters $\mathbf{d}(\alpha, b, c, d)$ are used to construct their mean $\tilde{\mathbf{d}}$ and covariance matrix $\mathbf{D}$. In addition, an estimate of the noise variance σ_r^2 is calculated. The vectors $\mathbf{p_{pr}}$, $\tilde{\mathbf{d}}$ and the matrix $\mathbf{D}$ are then inserted to a model database and linked with the particular surface of the model.



(a) Seeds in 3D (b) Seeds in 2D

Figure 4. Seed initialization

Seed initialization. Each detected vertex is aligned with the surfaces of the model database as in [7]. The alignment determines the rotation and translation parameters of the model. Its size and shape parameters, associated with the surface during model training, are retrieved from the model database. There are two possible ways in which the model can be aligned to the vertex each corresponding to two possible orientations of the graspable surface, as shown in Fig. 4 (a). Thus, if N the number of objects and k the number of graspable surfaces of each object in the model database, the number of hypotheses generated by each vertex becomes $2kN$. The parameters of the aligned models are then better approximated by projecting the boundary of each model's aligned surface to the image plane (see Fig. 4 (b)) and fitting an SQ model to the points. The seeds are then passed to the next step of the algorithm, where their refinement will take place.



(a) Recovered object (b) Recovered boundary

Figure 5. Successful recovery

Seed evolution. Evolution of the boundary of a seed's aligned surface on the image plane is inspired by the approach described in [24] . We suppose that $i(x,z)$ is the binary edge image resulting from the projection of the edge range points on the image plane. $b(x,z)$ is the image obtained when a boundary measure $m(u)$ is applied to $i(x,z)$, so that $b(z,x) = m(i(z,x))$. We now assume that the image b contains the real boundary corrupted by independent and additive Gaussian noise of standard deviation σ_e. If $C_\mathbf{p}(x,z)$ the model boundary curve depended on the parameter set $\mathbf{p}$, the posterior distribution of $\mathbf{p}$ is proportional to the function M_e defined in (12), where k a constant template magnitude and $Pr(\mathbf{p})$ the prior distribution of $\mathbf{p}$ (see [24] for the derivation)

$$M_e(b,\mathbf{p}) = ln(Pr(\mathbf{p})) + \frac{k}{\sigma_e^2}\sum_{C_\mathbf{p}} b(x,z) \qquad (12)$$

Maximization of M with respect to $\mathbf{p}$ will result to a maximum a posteriori boundary matching the data. The sum in (12) is taken along points on the boundary. N such points can be retrieved using the parameterization introduced in (1), by setting $\eta = \eta_b$ (see (2)) and letting ω take N values such that $\omega_i = i\pi/N, i = 1..N$. This parameterization however supplies more points in the large curvature areas of the curve. To achieve uniform distribution of the points along the template alternative parameterizations can be adopted, like the one presented in [1]. Assuming uniform prior makes the first term in (12) constant, and thus unimportant for the optimization. The second term expresses the likelihood of the parameters which is proportional to the quantity L_e defined in (13). The seed alignment provides the initial set of parameters $\mathbf{p}$. The boundary is recovered by maximizing (13) with respect to the subset $\mathbf{p_e}$ of $\mathbf{p}$.

$$L_e(b,\mathbf{p}) = \sum_{i=\omega_1}^{\omega_N} b(x_w(\mathbf{p};\eta_b,i), z_w(\mathbf{p};\eta_b,i)) \qquad (13)$$

The boundary measure m can be defined in a variety of ways (see [8] for a more detailed discussion about boundary measures). Defining m as an exponential function of the distance of the pixel to the nearest edge point is one of the most widespread solutions. The Euclidean distance used in our experiments.

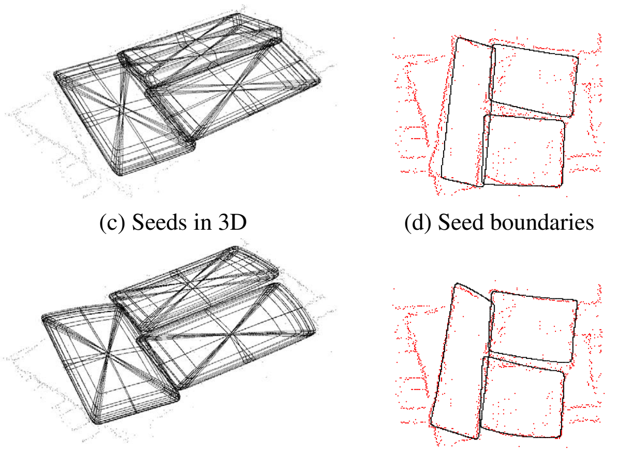
For fitting the points inside the region surrounded by the boundary we minimize the function M_r which corresponds to the posterior distribution of model parameters, with regard to the subset $\mathbf{p_r}$ of the parameter set $\mathbf{p}$. The prior term considers only the deformation parameters and biases the recovered model towards their mean value. The likelihood term is proportional to the function $L_r(p)$ defined in (11). The mean value of the deformation parameters $\tilde{\mathbf{d}}$ their covariance matrix $\mathbf{D}$ and the variance of the data points σ_r^2 are determined during the model training phase.

$$M_r(\mathbf{p}) = (\mathbf{p_d} - \tilde{\mathbf{d}})^T\mathbf{D}^{-1}(\mathbf{p_d} - \tilde{\mathbf{d}}) + \frac{1}{\sigma_r^2}L_r(\mathbf{p}) \qquad (14)$$

Seed evolution is based on an iterative framework. Each iteration consists of two steps: Boundary finding by maximizing (13) followed by region fitting by minimizing (14). The former updates the $\mathbf{p_e}$ and the latter $\mathbf{p_r}$ subsets of the parameter vector $\mathbf{p}$. The algorithm continues iterating while

L_e increases and M_r decreases. An object recovery is considered successful when the costs associated with the objective functions drop below user defined thresholds. A successfully recovered model is depicted in Fig. 5 (a), and its boundary in Fig. 5 (b). The model corresponds to the seed (a) initialized in Fig. 4 (a),(b).

The alert reader is perhaps driven to the conclusion that although we claim utilizing volumetric models like superquadrics, we actually use surfaces for segmentation. This does not hold for the following reason: After the end of the evolution of all seeds, range points which have a small distance from each model are associated with it. These may include points which do not belong to the exposed surface of the model. An MDL-based model selection procedure is now invoked (see [11] p.122 for a detailed discussion) which retains the models which more accurately describe a bigger number of range points than others. Thus, models satisfactory supported by points along many bounding surfaces are favored against others supported only by points on the exposed surface.



(c) Seeds in 3D

(d) Seed boundaries

(e) Recovered objects

(f) Recovered boundaries

Figure 6. Configuration recovery

3. Experimental results

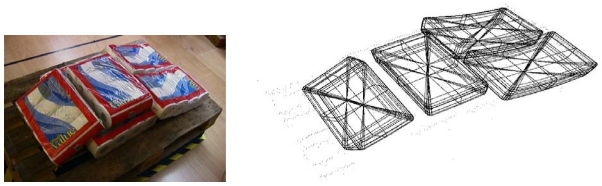
We performed extensive experiments with the system using a variety of target objects, like sand sacks Fig. 3 (a), box-like boxes wrapped in a transparent foil Fig. 7 (a) and cardboard boxes Fig. 8 (a). The models recovered by our technique are depicted in Fig. 6 (e)-(f), Fig. 7 (b), Fig. 8 (b) respectively. In addition, Fig. 6 (c) and Fig. 6 (d) show the seeds and projections of their boundaries in the image plane, which successfully recovered the objects of Fig. 3 (a). These figures may give to the reader an impression of the recovery accuracy of our approach. In terms of robustness, our experiments demonstrated that the system only occasionally fails to find at least one object in the pile. However, problems are encountered, if the objects are tightly placed too close one after the other, when no edges can be detected. We expect that a sensor with a higher accuracy or an additional sensor could be used to overcome this problem. The edge and vertex detection algorithms were implemented in C. Boundary and region fitting were implemented in MATLAB, using the Levenberg-Marquardt algorithm [9]. For all our experiments a Pentium $3, 600MHz$ was employed. We used $N = 100$ uniformly distributed points on the boundary for fitting. For all the cases edge and vertex detection lasted about 12 seconds on the average. The average time for seed evolution was about 10 seconds. The overall execution time depends on the number of initialized seeds which as aforementioned depends on the number of vertices and the number of different objects expected to be found on the pallet. 44 seeds were initialized in the boxes case, because two different objects are expected to appear. 20 and 8 seeds were initialized in the box-like and sack object case respectively. The overall execution time was 452 seconds for the boxes, 252 for box-like objects and 124 for the sacks. Note the if one processor per seed is used the execution time becomes less then 22 seconds. Execution time is expected to reduce by porting the entire code to C.

4. Conclusions

We presented a system for automatic unloading of piled slightly deformable box-like objects, using SQ as models. Superquadric seeds are aligned at the vertices of the objects. Iterative seed evolution led by a two-dimensional object boundary finding resulted in a robust and fast system. Experimental results demonstrated the applicability of our approach to recovering a variety of target objects. We assumed however that the size of the target objects is known. If this is not the case, boundary finding should additionally recover the unknown size of the objects. This would involve initialization of small seeds growing along their normal vector until they encounter edge points. The boundary measure used to define the attraction forces should change: The closest boundary point along the normal direction of the model should be determined, as in [25] p.141. The distance values should be recomputed each time the model grows. Such solutions however are computationally intensive. An efficient implementation of such a framework is the target of future work.

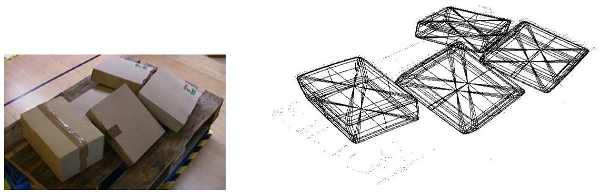
Acknowledgments

I wish to thank S. Romdhani, A. Jaklic and S. Rahmann for useful discussions. This work was supported by



(a) Intensity (b) Recovered objects

Figure 7. Box-like objects



(a) Intensity (b) Recovered objects

Figure 8. Boxes

the German ministry of economy and technology under the PRO INNO program, Grant No. KF 0185401KLF0.

References

[1] E. Bardinet, L. D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding: CVIU*, 71(1):39–54, July 1998.

[2] A. Barr. Superquadrics and angle preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23, 1981.

[3] P. Besl and R. Jain. Segmentation Through Variable-Order Surface Fitting. *IEEE TPAMI*, 9(2):167–192, 1988.

[4] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision Graphics and Image Processing*, 32:29–73, 1985.

[5] T. Binford. Visual perception by a computer. In *IEEE Confernece on Systems and Controls*, pages 116–123, 1971.

[6] A. Chakraborty, L. H. Staib, and J. S. Duncan. Deformable boundary finding influenced by region homogeneity. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 624–627, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.

[7] 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, Nov.-Dec. 1989.

[8] L. D. Cohen and I. Cohen. Finite-Element Methods for Active Countour Models and Balloons for 2D and 3D Images. *IEEE TPAMI*, 15(11):1131–1147, Nov. 1993.

[9] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.

[10] 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.

[11] A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer Academic Publishers, Dordrecht, 2000.

[12] 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, Feb. 1999.

[13] D. Katsoulas. Robust extraction of vertices in range images by constraining the hough transform. In F. Perales et al, editor, *Pattern Recognition and Image Analysis*, number 2652 in LNCS, pages 360–369. Springer, June 2003.

[14] M. Kavoussanos and A. Pouliezos. Visionary automation of sack handling and emptying. *IEEE Robotics and Automation Magazine*, 7(4):44–49, 2000.

[15] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE TPAMI*, 16(1):38–53, 1994.

[16] 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, Nov. 1992.

[17] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of Range Images as the search for Geomatric Parametric Models. *International Journal of Computer Vision*, 14(3):253–277, Apr. 1995.

[18] D. Newcorn. Robot gains eyesight. *Packaging World*, October 1998.

[19] H. Nguyen, M. Worring, and R. Boomgaard. Watersnakes: Energy-driven watershed segmentation. *IEEE TPAMI*, 25(3):330–342, 2003.

[20] C. F. Olson. A general method for geometric feature matching and feature extraction. *International Journal of Computer Vision*, 45(1):39–54, Oct. 2001.

[21] S. Sclaroff and L. Liu. Deformable shape detection and description via model-based region grouping. *IEEE TPAMI*, 23(5):475–489, 2001.

[22] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE TPAMI*, 12(2):131–147, 1990.

[23] L. Staib and J. Duncan. Model based deformable surface finding for medical images. *IEEE Transactions on Medical Imaging*, 15(5):720–731, 1996.

[24] L. H. Staib and J. S. Duncan. Boundary Finding with Parametrically Deformable Models. *IEEE TPAMI*, 14(11):1061–1075, Nov. 1992.

[25] C. Xu, D. L. Pham, and J. L. Prince. Image segmentation using deformable models. *Handbook of Medical Imaging Medical Image Processing and Analysis*, 2:129–174, May 2000.

[26] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[27] S. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE TPAMI*, 18(9):884–900, 1996.