Multiple-Superquadrics based Object Surface Estimation for Grasping in Service Robotics

Tiberiu T. Cociaş, Sorin M. Grigorescu and Florin Moldoveanu
Departament of Automation, Transilvania University of Braşov, Braşov, Romania
Email: tiberiu.cocias, s.grigorescu, moldof@unitbv.ro

***Abstract*-In this paper, a method for approximating an object's surface defined by 2.5D data is presented. The goal is to determine a compact volume which may be subjected to manipulation actions under a proper grasp planning. The object volume is obtained based on a union of parameterized geometric shapes, also known as superquadrics. In order to reduce over-segmentations and also the number of fitting combinations, the object space is defined as a 3D bounding Region of Interest (ROI). Further, via voxel decomposition, the ROI is decomposed in more meaningful smaller ROIs that can capture a large number of features from the object of interest. By using a fixed size voxel grid, the process can be slowed down for the cases of large objects. A boosting speedup of the process is proposed through dynamically adjusting the sides of the voxels for each object. Finally, a method for voxels merging is proposed. Each of the newly created ROIs, related to a particular region of the object, will be hosting a superquadric model which optimally estimates the considered surface.**

***Index Terms*–Superquadrics, surface estimation, space decomposition, robot vision, service robotics.**

I. Introduction

Perception and scene understanding, although a trivial process for humans, is currently stable in robots only under carefully controlled conditions. Many human actions are driven by the subconscious without the need to think, whereas a robot must plan and analyze every intuitive action. In an ordinary service robotics scenario, for successfully completing a grasping task, several analytical issues have to be solved, such as the accurate computation of the object's *position and orientation* (pose) with respect to the robotic platform, followed by a safe path planning that drives the robot to a nearby location which enables grasping capabilities. To complete the manipulation task, a-priori knowledge regarding the geometry of the object of interest is required in order to fulfill a proper grasp.

The a-priori knowledge is usually represented by a 3D *Point Distribution Model* (PDM) of the initial query object. After the object is recognized and replaced with this PDM 3D model, various techniques, such as visual servoing, are employed for manipulation, according to a predefined metric [1], [2]. The lack of a-priori knowledge hence creates one important challenge in service robotics. This issue is encounter when the geometry of a novel object cannot be found in the available database or is too particular. In case of a service robot operating in an unstructured environment, it is impractically to design and implement an exhaustive database that can cope with all encountered entities. For this case, an approach that uses no other a priori information outside the perceived scene has to be considered. In literature, several approaches that try to cope with these issues are described in [3], [4], [5]. Some of the techniques try to recover the object's convex hull based on a sequence of images (shape from silhouettes) as the robot drives around the object [3]. This approach needs however the precise pose of the reconstructed object which must be provided beforehand. Another drawback is the large number of images and the fact that the approach requires images from around the entire perimeter of the object.

For a proper object grasping action, it is not needed to fully reconstruct the 3D hull of the considered object. This can be overcome by finding the contact points which allow an easy and safe manipulation. In these sense, by approximating the query object with a generic geometric volume or shape, a rough 3D representation can be obtained. Because of their simplicity and flexibility, the superquadrics are a very good candidate for this job. Defined by a simple formula, superquadric models can resemble many shapes like elipsoids, spheres, cylinders and even cuboids. They were first introduced by Barr in [17] and they quickly became a popular geometric modeling tool.

In [6], [7], individual superquadrics were used to model different objects in an unstructured scene. This method of estimating a surface is very efficient when the grasp deals with regular shaped objects. In case of more complex entities, such as statues or animals, one single approximation is not enough to find the least required number of grasps points [9]. However, for this case, superquadrics are still a good approach since the surface can be estimated through a collection of different generic shapes. In [8], the authors used multiple-superquadrics models for the purpose of grasping, but limited the manipulation capabilities to a structured planar environment. On the other hand, in [10], a new model for representing an unorganized object PDM is described. The PDM's set of points is described with a union of superellipsoids, firstly segmenting the object and then model the shape through a region growing approach and a series of split and merge operations. All the above algorithms require as input for the modeling process a dense 3D point representation of the object. Since a service robot usually perceives the environment from only one visual perspective, the object recognition and 3D reconstruction process has to rely only on 2.5D (2D and depth) representations. An attempt

to combine 2.5D data and superquadrics was proposed in [2]. This paper presents a fast and robust technique for estimating a surface through multiple superquadric approximations. The entire surface of the object is divided in more meaningful sub-regions, each of which will be the host of an individual superquadric shape.

The rest of the paper is organized as follows. In Section II, a method of perceiving the environment together with the segmentation and initial 3D Region of Interest (ROI) extraction is proposed. The implicit surface formulation and the 3D query decomposition are presented in Section III. The testing setup and evaluation results are given in Section IV, before conclusions and outlook.

II. ENVIRONMENT SENSING

In the ideal grasping situation, the robot should determine the object of interest, compute the optimal path to reach it and finally find the best grasping points that can safely manipulate the object. The overview of a processing chain for a real grasp situation is presented in Fig. 1. The volumetric property of the grasp object is determined using a superquadric based surface estimation principle. The grasp objective is achieved through six main steps. First, the scene is perceived using a stereo vision camera. In the next step, a color based segmentation process is used to identify the object which needs to be manipulated closely followed by the third step characterized by the computation of a framework on which the reconstruction process will take place. The fourth step objective is to break the perceived 2.5D object representation in more meaningful regular sub-parts. Next, in each sub-part a separate supercuadric model is fitted. In the end, grasp points are determined as intersection points between the 2.5D representation and superquadric shape.

A. Robotic sensing capability

The field of depth perception in robotics is mainly dominated by stereo-camera systems (Point Grey's Bumblebee®) [13], structured light sensors (e.g. MS Kinect®), Time of Flight (ToF) cameras (e.g. Mesa®) [11] and laser scanners (e.g. Sick®) [12].

Each of the mentioned perception approach comes with additional advantages and shortcomings. The ToF [11] camera exploits the speed of light in order to determine the relative structure of a scene. By measuring the time needed by a light ray to reach a particular surface, the distance between the camera sensor and that surface can be estimated. Although the principle produces a precise and dense 3D representation of the environment, it has as main drawback the large errors obtained when used in outdoor applications. This phenomenon takes place when the sensor interferes with natural light. Structured light based cameras (e.g. Kinect) use projected light patterns to estimate depth. The observation of a pattern is different based on the distance to the object they are projected on. On the other hand, the laser sensor copes very well in most outdoor scenarios. It also obtains good marks in the area of sensibility and sampling rate. Its main drawbacks are the large size and high energy consumption.

One of the most used and investigated imaging mechanism in robotics is the stereo-camera. Such a system was also used for the experimental results discussed in section IV. Concerning the stereo-vision principle, the 3D information is obtained by comparing the scene information from two slightly different perspectives. Focusing on a particular feature, the relative distance between the positions of the same feature in the two images can be used as a measure of depth between the image sensor and the considered feature. Since there is no actually relation (in terms of coordinate system) between the camera sensor and the scene, a pre-calibration step has to be considered [13]. The purpose of this step is to determine the *intrinsic* and the *extrinsic* parameters of the camera. The first parameters characterize a series of built-in values such as focal length f, optical centers (c_x, c_y) of sensors, aspect ratio and skew constant. On the other hand, the extrinsic parameters contain the transformation needed to describe the pose of the camera related to different coordinate systems. The transformation is obtained based on the *rotation* and *translation* operators $[R, t]$. Since the depth estimation is achieved through the stereo configuration of two camera sensors, the pose of the second camera is actually the same as the first one but translated on the X axis with a certain quantum, known as *baseline*. A larger distance between camera sensors will provide good depth perception for distant objects, whereas a small baseline will give a good perception of objects closer to the sensor.

Before extracting any 3D information from a scene, several pre-processing steps have to be considered. Since the process of depth perception is based on projections, one important issue is that the observed images are purely projection [15]. This problem can be solved by removing the distortions in each image. Further, before comparing the features from stereo images we have to ensure that both images are on a common plane and have a standard coordinate system. This transformation is calculated through *image rectification* [14].

The goal of depth estimation is to compute a so-called disparity map $f(x, y, z)$, in the form of a 2D image. The resolution of the image is the same as the input RGB ones. Each grey level intensity in the disparity image corresponds to a disparity d computed as the difference between the x coordinate of the same features (projected point P) in the stereo image:

$$d = x_L - x_R \,, \tag{1}$$

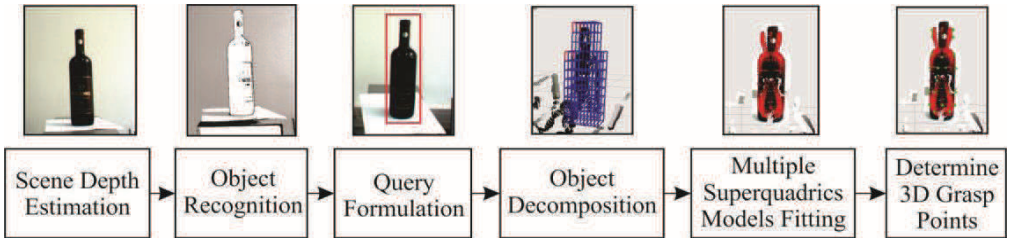


Fig. 1. Processing chain of the proposed volumetric object estimation approach.

where x_L and x_R are the coordinate of a pixel along the *X axis* in the left *L* and right *R* images, respectively. One of the most popular algorithms for computing the disparity map is the *Block Matching* approach [16]. In the following, depth information will be referred to as *point cloud*.

B. 3D object query formulation

Before actually fit a superquadric shape into the scene we have to know the exact pose of the object which surface is to be approximated, followed by the implicit shape modeling approach that approximates the 3D shape of the object [17]. The optimal superquadric shape computation will be described in Section III. The pose of a superquadric shape and some additional parameters such as height, width and depth can be estimated based on a query region defined around the segmented object. This is actually a bounding box that roughly defines the object in the 2D domain. Having pointed out the approximate 2D region and based on the imaged scene (represented as a point cloud of 3D points reprojected from the disparity map) a 3D bounding ROI can be calculated. This 3D region is defined as a query for the superquadric estimation technique, that is, "*Given a 3D point cloud representation of an object of interest, which superquadric shapes optimally describe the considered object area?*".

All the 3D points that don't lie inside the 3D bounding ROI are rejected from the imaged point cloud since they do not capture any information related to the considered object. The height and width of the 3D ROI can be easily determined by reprojecting in the 3D scene its 2D corresponding coordinates. These reprojected points can actually describe only a planar surface. Since the scene is viewed only from one perspective, behind the planar object of interest no 3D points can be seen. Having this in mind, it is obvious that the highest number of 3D points belong to the 3D object. This assumption is valid only if the occlusion of the object is less than half of the bounding ROI. Under this presumption, the depth (thickness) of the bounding box can be estimated through a statistical analysis of the distribution of the point cloud along the Z axis, but only for the crop area defined by the planar bounding ROI. Fig. 2 describes the reprojection of a 2D ROI into the 3D virtual environment using the above principle.

The point distribution analysis which determines the actual thickness of the query is performed on the disparity map histogram related to the 2D region of interest. Instead of searching for the highest number of points with the same disparity intensity, we try to find the highest cluster of points which have an appropriate intensity. This operation defines a certain *aperture* for each cluster. The higher number of connected points from a certain aperture is considered to represent the query object. The aperture defines the depth difference between the front and the back faces of the 3D bounding ROI, respectively. In this way, a 3D query region which contains a point cloud representation of the object whose surface is to be estimated is obtained. This approach reduces drastically the possible fitting combination of the superquadric into the considered surface.

III. Super-Quadric Based Object Surface Estimation

One goal in a successfully handling operation is finding optimal 3D gasping points. For this purpose it is not actually needed to accurately reconstruct the object's surface. By fitting a series of superquadric models inside the query region, a rough surface estimation can be obtained. Thanks the implicit representation of the PDM models, we can determine which 3D points from the object point cloud lays on the superquadrics surfaces and use them to manipulate the object.

A. Implicit superquadric representation

A 3D surface can be obtained by the spherical product of two 2D curves [17]. In this way, by crossing a half circle in *the* plane orthogonal to the (x, y) *axes* with a full circle in *the* (x, y) plane, a 3D sphere can be obtained. Analogous to this approach, a super-ellipse (in the 2D domain) can be obtained based on the follow equation:

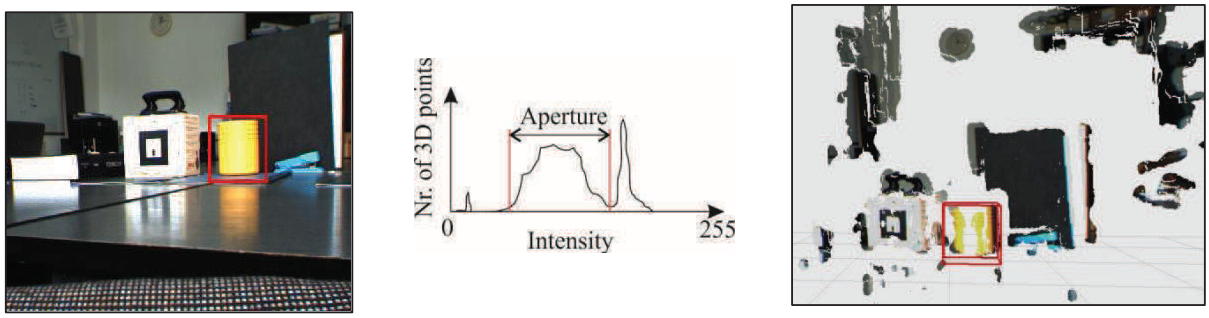
$$\left(\frac{x}{a}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\varepsilon}} = 1, \tag{2}$$

where, a and b represent the two semi-axes of the ellipse, x and y are the coordinates of the point laying on the ellipse and ε is the squareness factor. Based on this approach, superquadrics resemble many geometric shapes such as cylinders, spindles, octahedral, cubes etc.

In the 3D space, a generic superquadric model can be defined as [17]:

$$\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} = 1, \tag{3}$$

where, a_1, a_2 and a_3 are the three semi-axes of the ellipse, ε_1 and ε_2 are the two squareness factors and (x, y, z) is a point laying on the super-ellipse. Next, a function F, as the



(a) (b) (c)

Fig. 2. 3D Reprojection of a crop area defined for a certain imaged object. (a) 2D query formulation. (b) 3D histogram of the corresponding 2D cropped region. (c) 3D bounding box estimation of the initial 2D query.

one in (4), also known as an *inside-outside* function, is used to determine if a point $P\,(x, y, z)$ lies inside or outside the superquadric region. If the result of F is < 1, the point P is inside the 3D shape, whereas if $F = 1$ implies that P is on the surface of the superquadric and if F is greater than 0, then the point is outside the considered geometric model.

$$F = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}}, \quad (4)$$

The major advantage of superquadrics is that through only eleven parameters a large variety of geometric shapes can be obtained. However, since in a real scene the pose (rotation and translation) is not in a general form, six of them describe the pose of the model, that is, via three Euler angles for rotation and three values for the translation. The actual modeling process of a superquadric shape is achieved only with the help of five parameters.

The first three, a_1, a_2 and a_3 are used to describe the global features related to the size of the shape (height, width, and depth), while the last two parameters ε_1 and ε_2 define the curvature of the superquadric edges (squareness). A higher value defined for ε_1 and ε_2 specifies a more convex shape, whereas lower values will generate a heavy non-convex model. For optimal shape representation, the values of these parameters should lie in the interval $A = [0, 2]$.

B. Fitting superquadric models on 3D data points

Given a set N of 3D data points, that is, a point cloud estimation of the real object, the challenge is to determine the parameters of the superquadric that optimally fits it into the respective 3D data. From a total of 11 parameters, three of them are easy to determine if we consider the center of the 3D bounding ROI as the mass center of superquadric. This assumption is correct since the bounding ROI was statistically positioned to represent only the point cloud of the object.

The rotation of the superquadric is obtained also statistically through *Principal Component Analysis* (PCA) [*18*]. The goal of PCA is to determine the optimal orientation for the considered superquadric model. This angle reduces the parameter space of the superquadric with another 3 parameters, leaving undefined only five parameters from a total of 11.

Since one perspective of the object is enough to establish the height and the width, parameters , a_1 and a_2 are determined by finding the top and the bottom, respectively the left most side and the right most side of the 3D data point concerning each bounding ROI. The depth semi-axe a_3 cannot be determined at this point because of the lack of information regarding the backside of the object. In the end, only three parameters are left for estimation, namely two squareness factors ε_1, ε_2 and one semi-axis a_3. As in [19], a least squares fitting method is used to determine these last parameters. Particularly, the method tries to determine which parameters combination will generate the best object fit. This optimal fit is associated with a low global distortion which is minimized thought the following equation:

$$\sum_{i=1}^{N} d(x_i, y_i, z_i)^2, \quad (5)$$

where $d(x, y, z)$ is the distance between the superquadric surface and a 3D data point. To avoid false approximations for the case when the points set is not closed to the superquadric surface but some of them still approximate it correctly, a special coefficient $\sqrt{a_1 a_2 a_3}$ is considered [19].

In order to make the fitting process time efficient, the distance between points is estimated with an approximation based on the implicit form of the superquadric and not by the standard Euclidian distance:

$$d(a, x, y, z) = f(a.x.y.z)^{\frac{\varepsilon_1}{2}} - 1, \quad (6)$$

where a represents the superquadric parameters and (x, y, z) are 3D PDM point coordinates. In the end, the mean distortion D per point given is by:

$$D(a, x, y, z) = \frac{\sqrt{a_1 a_2 a_3}}{N} \left(f(a.x.y.z)^{\frac{\varepsilon_1}{2}} - 1 \right)^2. \quad (7)$$

In order to minimize D, a non-linear regression method has to be considered [20]. In this way, the remaining three parameters can be efficiently determined leading to an optimal fit of a superquadric shape onto the input 3D dataset.

C. 3D target decomposition

Let us suppose that the object which surface we want to approximate is a tomato situated on a flat surface. Generally speaking, a tomato has a spherical shape which surface can be then estimated through a spherical superquadric with the parameters $a_1 = a_2 = a_3 = 2 \cdot radius$ and $\varepsilon_1 = \varepsilon_2 = 1$. In service robotics, this approximation through one single superquadric may not be enough for many household items which have more complex shapes. For example, a glass can be easily estimated by a single cylindrical model, whereas a cup, because of its handle, must be constructed from multiple models.

Within the initial 3D bounding ROI, only one superquadric, which must be globally optimized using all 3D points from the box, can be fitted. We propose a decomposition of the initial 3D bounding ROI into more meaningful sub-regions, each containing a relevant density of 3D points. Inside of each sub-bounding ROI, a separate superquadric can be optimally fitted, leading to a more accurate object approximation.

The decomposition process starts by generating an unsteady voxel grid inside the initial 3D bounding box. In order to capture more information from the scene, the resolution of the grid is not set to a fixed value, which, for the case of large objects, can be to complex and time consuming, while for small objects the voxelization can miss important features. Instead, the grid sides are dynamically chosen through percents of the side of the initial bounding ROI. In this way, a voxel can have different side lengths for the width, height and depth. For example, a book has after the decomposition process 22 grid voxels describing the width, 31 for height and 10 for depth, although the real dimensions

of the book is 220 mm by 300 mm by 15 mm. Fig. 3 describes a voxel grid defined for a cup.

Next, al voxels inside the object bounding ROI are labeled. A voxel is considered valid if it contains 3D point information inside. The voxels with no 3D points inside are further rejected. In order to avoid sparse or false voxels (voxels containing very few points inside), an additional threshold value is considered during the labeling process. The exact number of 3D points inside each voxel is not relevant if it is greater than the above mentioned threshold value. In this way, regions with low 3D information (caused by bad illumination or lack of texture) are not discriminated by high point density agglomeration.

Based on this rough object definition, the next goal is to establish a series of connection between voxels. In order to capture as much information as possible the appearance of each voxel cluster will be represented by a rectangular prism. In some situations, a more complex voxel merge process can generate fewer regions but with the risk of losing many important features from the object. Actually, the objective is to generate as much sub-regions as possible that can capture even smaller object characteristics. Although the clustering approach generates only rectangular volumes, after the optimized fitting stage, a smooth surface is obtained.

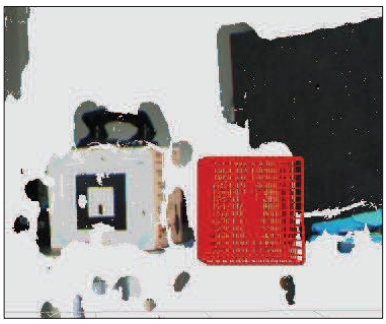


Fig. 3. 3D shape voxelization.

Since in the 3D ROI the only depth information is the one describing the entire object, each of the clusters will inherit this generic depth, allowing the actual fit process to establish the real depth of each region. Because the initial 3D bounding ROI is centered onto the object, all the new regions mass centers are related to the same depth.

The clustering process is applied as follows. Because of the single input perspective of the object, the voxels are scanned only in the 2D domain. By checking the label of each 4-neighboring voxel, spatial dependences are obtained. The proposed pseudo–code algorithm is described in Fig. 4.

The method scans the voxels in the same way as for the 2D domain, that is, from top-left to bottom-right. The third dimension representing the depth is at this point omitted. In this approach, the voxel cropping process can occur in only two directions. The scan searches in each direction only for connected valid voxels. If a discontinuity along a certain axis (X or Y) is found, the side of the rectangular prism ends in the point where the discontinuity starts. When a valid voxel is checked, the first action is to determine if the left or above neighbor is valid. If this is the case, the voxel is skipped since it is obvious that it is a part of another rectangular prism. The final result of this process is a lower number of overlapping rectangular prisms. Each parsed prisms gets a parsing flag. After an initial fusion search, all the voxel have the flag on. Additional valid voxels, which will be also kept for representing the object, will be detected by using the same algorithm but applied to unparsed voxels.

1: **for** each valid voxel v_{x_i} along the X axis
2: **for** each valid voxel v_{y_j} along Y axis
3: **check** the neighbors $n(x_i, y_j)$ of $v(x_i, y_j)$
4: **if** left $n(x_{i-1}, y_j)$ is valid
5: **mark** $v(x_i, y_j)$ as parsed voxel
6: **then** ignore $v(x_i, y_j)$
7: **else if** upper $n(x_i, y_{j-1})$ is valid
8: **mark** $v(x_i, y_j)$ as parsed voxel
9: **then** ignore - $v(x_i, y_j)$
10: **else**
11: **for** each right consecutive valid voxel $v(x_{i+a}, y_j)$
12 : **Rectangular prism** width $L = voxel\_length * a$
13: **for** each down consecutive valid voxel $v(x_i, y_{j+b})$
14 : **Rectangular prism** height $l = voxel\_length * b$
15: **end**
16: **end**
17: **check** again for unparsed voxels $v(x_i, y_j)$
18: **while (** unparsed $v(x_i, y_j) > 0$ **)**
19: **jump** to line **1**
20: **finish**

Fig. 4. Pseudo-code for the voxels fusion algorithm.

In the end, a series of different rectangular prisms driven by the 3D information extracted from the scene are obtained. Each of these prisms will host a particular superquadric model. As a whole, the superquadrics describe an approximation of the initial query model.

Having defined an implicit surface parameterization for the object of interest, the next step is to determine the points that can ensure a proper grasp action. Since the superquadrics can roughly approximate a surface, not all the points from the implicit surface can be used for this process. In order to find only good points, a filtering approach is considered. The filter takes into consideration only the 3D points, belonging to the query object, which verify (4). In this way, only the 3D points which are located on the surface of the superquadrics will be kept. The grasp planning and the analysis of the manipulation capabilities are performed using the tool proposed in [21]. The GraspIt! approach [21] tries to find the points of contact between a hand and a 3D object.

IV. Experimental results

The objects used during the tests contain both regular surfaces, as also articulated or prominent regions. For imaging the robotic scene, a Bumblebee® X2 stereo camera was used, while for grasping and manipulation a Pioneer® 3 platform equipped with a 5 Degrees of Freedom (DoF) robot arm. All the tests were conducted in an indoor environment with constant illumination. The objects were placed on flat surfaces in random unknown positions and orientations in order to avoid over-segmentation.

During performance evaluation al the measurements and approximations were compared against a *ground truth* (GT). The GT consist of a number of manual and automated measurements conducted on the objects: width, height, thickness, translation and rotation. The translation and rotation was measured with respect to a fixed reference coordinate system represented by an ARToolKit® [22] marker placed near the object of interest.

A comparative analysis between the estimated data and the ground truth can be observed in Tab. 1. Each object was evaluated against its real width, height, orientation and translation. Since no depth information can be extracted from a single view, the evaluation against depth or volume is questionable. The depth information of each superquadric component is computed as the mass center of the respective sub-bounding ROI.

In Fig. 5 an implementation of the proposed method against a general purpose object (e.g. bottle) is illustrated. By using a single superquadric approximation, the neck of the bottle will be omitted and considered as a body. The consequence of this issue is that no grasp points can be determined on the neck of the bottle. In our approach, the entire object is divided into two main regions: one for the body and one for the neck. In this way, as it can be seen from Fig. 5, the estimated shape approximates better the initial query silhouette. Because of the matching technique and the 3D bounding box, the contour of the object contains noise information. The noise is represented by a series of additional 3D points describing the surrounding region around the object border. To reduce this noise to an acceptable value an additional texture based filter which efficiently determines the object border was considered. The computational process is highly dependent on the number of 3D points parsed during the superquadric fitting process. As an example, a number of 22.998 3D points describe the object from Fig. 4.

V. Conclusions and outlook

In this paper, a fast and robust method for object surface estimation using multiple superquadrics was presented. The usage of a query bounding ROI describing the object of interest gives a primary assumption about the object's depth, enabling the algorithm to work efficiently on 2.5D data. By using a collection of parameterized models, the accuracy of object reconstruction is considerably increased. The voxel grid approach reduces the discrimination between regions with different 3D information density and in the same time increases the speed of the fitting process. By using different voxel side lengths the majority of the object features can be captured.

As future work, the authors consider the speed-up enhancement, since for the case of complex shapes, the number of approximated superquadrics increases dramatically. In this sense, parallel processing (e.g. Graphic Processors) represent a good candidate for this job. The authors also consider enhancing the accuracy of surface approximation by using structured light sensors (e.g. MS Kinect) which provide a denser and more accurate depth map representation.

Acknowledgements

This paper is supported by the Sectoral Operational Program Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the projects POSDRU/107/1.5/S/ 76945 and POSDRU/89/1.5/S/59323.

References

[1] D. Kim, R. Lovelett, and A. Behal, "Eye-in-Hand Stereo Visual Servoing of an Assistive Robot Arm in Unstructured Environments," in *Intern. Conf. on Robotics and Automation*, pp. 2326–2331, May 2009.

[2] S. Effendi, R. Jarvis, and D. Suter, "Robot Manipulation Grasping of Recognized Objects for Assistive Technology Support Using Stereo Vision," in *Australasian Conference on Robotics and Automation*, 2008.

[3] C.R. Dyer, *Volumetric Scene Reconstruction From Multiple Views* Foundations of Image Understanding, Boston, 2001 pp. 469–489.

[4] K. Shanmukh, and A. Pujari, "Volume Intersection with Optimal Set of Direction," in *Pattern Recognition Letters*, vol. 12, pp. 165–170, 1991.

[5] R. Szeliski, "Rapid Octree Construction from Image Sequences," in *CVGIP: Image Understanding*, vol. 58, pp. 23-32, July 1993

[6] D. Katsoulas, and A. Jaklic, "Fast Recovery of Piled Deformable Objects Using Superquadrics," in DAGM, pp. 172-181, 2002.

[7] M. Salganicoff, L. Ungar, and R. Bajcsy, "Active Learning for Vision-Based Robot Grasping," in *Machine Learning*, vol. 23, May 1996.

[8] F. Boughorbel, Y. Zhang, S. Kang, U. Chidambaram, B. Abidi, A. Koschan, and M. Abidi, "Laser Ranging and Video Imaging for Bin Picking," in *Assembly Automation*, vol. 23, pp. 53-59, Mar. 2003

[9] S. Colbert, R. Alqasemi, G. Franz, and K. Wollhag, "Development and Evaluation of a Vision Algorithm for 3D Reconstruction of Novel Objects from Three Camera Views," in *Inter. Conf. on Intelligent Robots and Systems*, (Taipei, Japan), pp. 5438 – 5445, 2010

[10] L. Chevalier, F. Jaillet, and B. Baskurt, "Segmentation and Superquadric Modeling of 3D Objects," in *Journal of Winter School of Computer Graphics*, 2003.

[11] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-Flight Sensors in Computer Graphics," in *EUROGRAPHICS*, pp. 119-134, May 2009.

[12] E. Trucco, and A. Verri, *Introductory Techniques for 3-D Computer*

TABLE I
COMPARATIVE EXPERIMENTAL RESULTS

Object	Approximation Type	Width [m]	Height [m]	Thickness [m]	ϕ [deg]	θ [deg]	φ [deg]	x [m]	y [m]	z [m]	Global Distortion	Nr. of superquadrics
Bottle	*Ground Truth*	0.053	0.35	0.053	0.1	1.3	4.1	0.1	0.23	1.134	-	-
	Single Superquadric	0.065	0.362	0.057	2.73	1.23	3.52	0.093	0.25	1.131	5.7313	1
	Multiple Superquadrics	0.057	0.349	0.057	2.73	1.23	3.52	0.097	0.27	1.131	2.0139	2

Vision, Prentice Hall PTR, 1998.
[13] R. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Great Bretagne, 2004.
[14] D. Oram, "Rectification for Any Epipolar Geometry," in *British Machine Vision Conference*, pp. 653-662, 2001.
[15] J. Heikkila, and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Puerto Rico), pp. 1106 – 1112, 1997.
[16] M. Brown, D. Burschka, and G. Hager, "Advances in Computational Stereo," *in IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, 2003.
[17] A.H. Barr, "Superquadrics and Angle-Preserving Transformations," in *IEEE_CGA*, vol. 1, no. 1, pp. 11–23, 1981.
[18] I.T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer, NY, 2002.
[19] F. Solina, R. Bajcsy "Recovery of parametric models from range images: the case for superquadrics with global deformations," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 2, pp.131–147, 1990.
[20] W.H. Press, S.A. Teukolsky, W. T. Vetterling, and B.P. Flanery, *Numerical Recipes in C,* Cambridge University Press, 1992.
[21] A. Miller, and P.K Allen, "Graspit!: A Versatile Simulator for Robotic Grasping," in *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110-122, 2004.
[22] *P. Malbezin, W. Piekarski, B. Thomas, "Measuring ARToolKit* Accuracy in Long Distance Tracking Experiments", 1st Int. Augmented Reality *Toolkit Workshop*, Darmstadt, Germany, 2002.

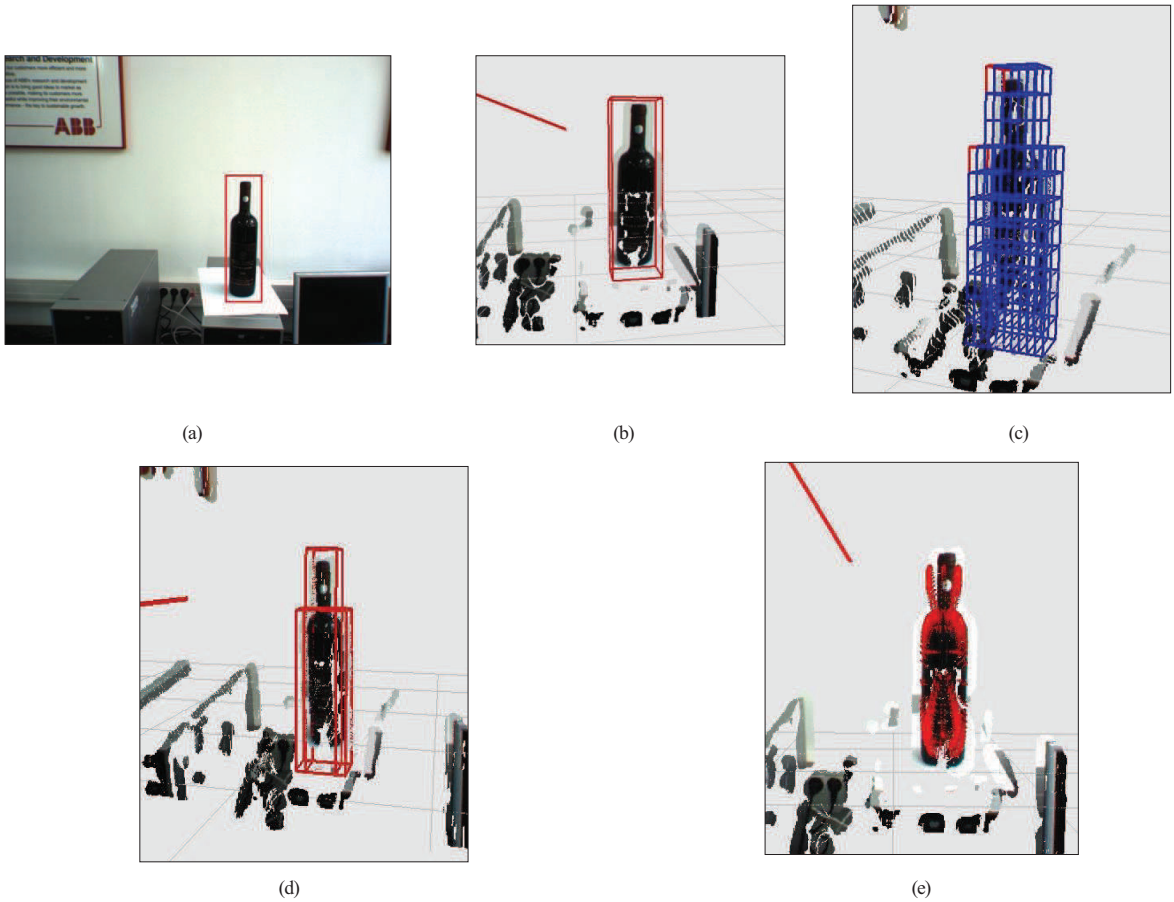


Fig. 5. Surface estimation process flow. (a) 2D segmented object. (b) 3D query formulation (bounding ROI). (C) Voxel based 3D shape representation. (d) Rectangular prisms which optimally crop the considered shape. (e) Final estimation of the initial query.