Generic Fitted Primitives (GFP): Towards Full Object Volumetric Reconstruction for Service Robotics

Tiberiu T. Cocias
Institute of Automation
Transilvania University of Brasov
tiberiu.cocias@unitbv.ro

Florin Moldoveanu
Institute of Automation
Transilvania University of Brasov
moldof@unitbv.ro

Sorin M. Grigorescu
Institute of Automation
Transilvania University of Brasov
s.grigorescu@unitbv.ro

Abstract

Service robotics applications, such as mobile manipulation in domestic environments, require 3D representations of the objects of interest to be grasped. Simple object recognition or segmentation cannot provide structural shape information mandatory for obtaining reliable grasp configurations. In this paper, the *Generic Fitted Primitives* (GFP) technique for volumetric reconstruction is introduced. The goal of the method is to build full 3D object shapes from a single camera perspective. In order to obtain the shape of the 3D primitive, we propose an energy-minimization algorithm based on the concept of *Active Contours* applied directly on 3D visual data. Our modeling approach produces compact closed surfaces (*volumes*) describing the objects of interest which can be further used for service robotics tasks, such as grasping or manipulation. The performance of the proposed technique has been evaluated against two different methods, i.e. generalized active contours and superquadric approximations.

Keywords

Active contours, 3D segmentation, 3D reconstruction, Robot vision systems, RGB-D sensors.

1 INTRODUCTION

In the last years, the 3D object reconstruction challenge gained a lot of attention in application fields such as service and industrial robotics, or virtual reality. In service robotics, 3D reconstruction is usually involved in providing information for path planning and object grasping in mobile manipulation [Dil09a]. In such cases, one major inconvenience regarding a service robot is that it can perceive the scene from only one camera perspective. This aspect produces large occluded areas altering the structure of the objects. Thus, an estimation of the object's 3D shape must be considered in order to obtain a full volumetric representation. Some methods try to reconstruct directly the volume of the object by discretizing the 3D space into a series of voxels [Bet00a]. Other approaches aim at defining implicit surfaces depicting different volumes through implicit functions [Bar02a]. In this paper, we propose the *Generic Fitted Primitives* (GFP) technique for fully approximating the particular volumetric information of the objects. The calculated models are intended to be used for improving the grasping capabilities of service robots. The input visual information has been acquired using structured light sensors, such as the MS Kinect®, and classical stereo cameras. An example of full 3D reconstructed objects from a typical service robotics scene is illustrated in Fig. 1.

In our GFP approach, the problem of 3D volumetric approximation has been divided into two phases. Firstly, a coarse object *detection* method is used to extract an initial object cluster for which its volume needs to be estimated. In the second phase, the cluster is used for fitting a GFP such that detected object will be fully reconstructed. The main contributions of the paper may be summarized as follows:

- the introduction of the GFP technique based on a modified formulation of the *Active Contours* principle; the deformation of the primitive shape is performed based on the normal direction of the so-called *control points* of the GFP;
- the usage of a GFP as an initial contour within the active contours framework; the modeling process time is thus improved since the number of iterations required to deform the initial shape is smaller;
- usage of the GFP approach for building full 3D volumetric models of objects of interest in the context of mobile manipulation.

3D object surface reconstruction is treated in a large number of publications. Some of the paper found in lit-

 ISBN 978-80-86943-75-6

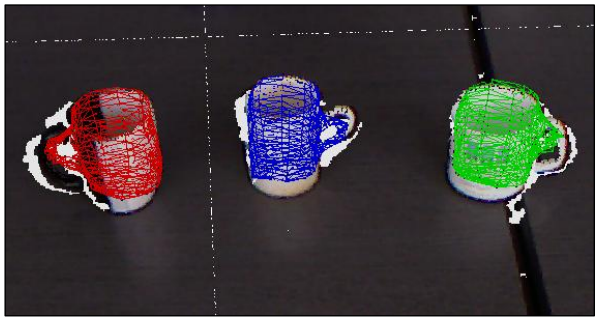


Figure 1: Full 3D volumetric reconstruction of multiple objects in a mobile manipulation scenario.

erature are based on the implicit representation of models [Bae02a, Bar02a], whereas other exploit explicit a priori surfaces (e.g. skeleton primitives) for augmenting the existing object structure [Gas01a]. One relevant implicit approach makes use of generic 3D shapes (e.g. such as spheres, cuboid, or ellipses) to roughly approximate the global object volume [Bar81a]. These types of methods are fast, do not need any a priori knowledge about the reconstructed surface, but lack the accuracy of the final approximated volume. A more refined volume can be obtain by partitioning the imaged surface in more meaningful sub-regions which can be further individually approximated using the same implicit principle as in [Coc12a].

A different approach to 3D modeling is based on *3D Object Retrieval* (3DOR) search engines [Tan08a]. The main drawback of this technique is that it needs a high amount of computational power to find the optimal match between a query representation (e.g. the imaged object) and a set of targets (predefined models from a database). In [Hua12a], the combination of RANSAC and Procrustes analysis is used for recovering the joint axes of objects. The algorithm does not make use of any a priori object knowledge, but it requires a large number of images depicting the object of interest. In [Mar09a], the authors present a primitive based approach for approximating simple regulated objects like plates, boxes, cans, etc. As opposed to our work, in [Mar09a], the primitives are represented by simple geometric models, such as cuboids, spheres, or cylinders, and not by primitive shapes that can capture different particularities that the objects might have. Krainin et al. [Kra11a, Kra11b], applied the concept of object tracking during manipulation for building online 3D models of objects using range sensors and 3D data processing techniques. Nevertheless, the model is represented by a *Point Distribution Model* (PDM) and not by a full 3D shape.

The rest of the paper is organised as follows. In Section 2 the components of the primitive based modeling apparatus, along with the involved methodology, is presented. Performance evaluation results are given in Section 3, before the conclusions from Section 4.

2 METHODOLOGY

In mobile manipulation, activities of daily living scenarios typically involve a large numbers of objects. The first step in the proposed framework is to segment the different objects and obstacles in the scene. As a result of segmentation, different 3D object clusters, or PDMs, are obtained. Along with the clustering procedure, the process also returns the object's class. These PDMs are used for modeling the GFP in such a way that it captures the particularities of the object. The block diagram of the GFP architecture is shown in Fig. 2.

Instead of using a large number of models as a priori information about a particular object, thus requiring a large number of shapes to be stored, we propose a more general approach through the use of GFPs. By using only one primitive per object class (e.g. mug, plate, bottle etc.), a considerably smaller sized primitives database is obtained. At the same time, the computation time is improved because the number of discrete items that need to be searched is reduced.

2.1 Initialisation: Cluster Extraction

The extraction of the scene clusters is important for the accuracy of the final primitive representation. Firstly, the objects of interest need to be recognised in order to select the correct GFP. We use a contextual object recognition approach [Son11a] through a classification process. Having obtained the object's class, the corresponding 3D cluster of the same object is extracted. The clusters are extracted as described in [Rus09a]. Namely, plane segmentation is used to divide an organized point cloud P into smaller meaningful clusters $C = [c_0, c_1, \ldots c_n]$ representing different entities. In Fig. 3 an example of object recognition (labelling) and object cluster extraction for a table-top scene is presented.

The output of the detection component is an isolated PDM representation of the objects of interest. Further, this representation is used to model the shape primitive such that in the end it models the particularities of the object as accurate as possible.

2.2 Generic Primitives

A generic primitive is considered to be an a priori known shape describing a number of particular objects from the same class. It is constructed in such a manner that it resembles many similar objects. In this way, an universal model for a certain class of objects is obtained. For example, different types of bottles can be roughly approximated by a joint pattern. The most important attribute of a shape is actually its global structure (*frame*) which, in a majority of cases, is similar to a

 ISBN 978-80-86943-75-6

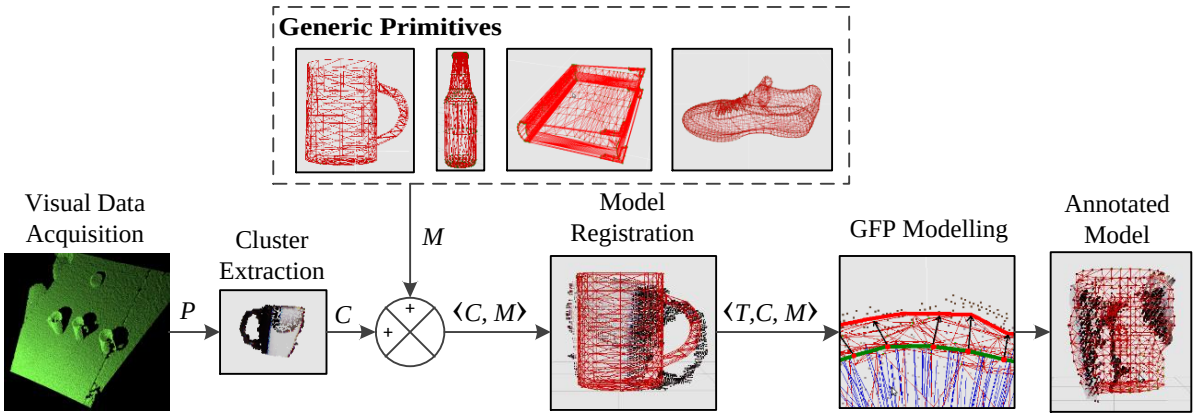


Figure 2: Block diagram of the GFP 3D volumetric estimation approach.

large number objects of the same class. In this sense, instead of finding the optimal object (from a considerably large number of different shapes from the same class) which best fits the PDM data, a modeling step applied to a generic primitive M is addressed for estimating its global object volume.

Depending on the geometric surface complexity, the generic primitive can be defined by a high density of 3D points. This aspect directly influences the processing time. A down-sampling filter used to reduce the number of PDM points is not encouraged because the global point cloud structure is altered. We approach this issue from the GFP's point of view. Namely, not all feature points are relevant for modeling the cluster's structure. For example, many of them are used only for the purpose of creating a volumetric surface. In this sense, each point in the GFP will receive a special flag or type. Hence, two point types are defined: *control* and *regular* points. A point which has received the *control* flag is considered to be part of the frame and it is positioned according to the 3D information in the point cloud, whereas a *regular* point is used simply to smooth the global structure of the shape, meaning it is moved according to the positions of the control points.

(a) (b)

Figure 3: Object detection through cluster extraction. (a) Scene labelling based on object recognition. (b) Euclidean cluster extraction.

The classification of GFP points in different types can be done either manually or automatically. The first procedure requires a human to manually select the point's type. Having in mind the required human interaction, the manual labelling of points is time consuming. On the other hand, the *automatic* type assignment has a lower accuracy, but it is usually much more efficient and does not suffer from subjectivity.

The automatic point selection is governed by a set of rules used to established the point's type [Cot95a]. Namely, control points are those obeying the following statements:

- points describing sharp corners of a boundary, detected as in [Web10];
- points marking the boundaries of M along the widest axis;
- points located at equal distance around a boundary between two control points obeying rule one;
- points marking a curvature extreme or the extreme points of the object [Wat01].

An example GPF of a bottle is illustrated in Fig. 4.

In terms of the GFP definition, the primitive model is a complex data structure composed of a vector M storing the 3D coordinates of all the features describing the generic model, a vector A containing the point type attributes of M, a vector W describing the mesh triangulation indexes used for 3D surface representation and, in the end, global characteristics such as height, length, width, rotation R, translation T and the overall number of features.

The objects are defined in a *local* coordinate system attached to each considered model. Thus, a *common* coordinate frame for both the GFP models and the cluster of the object of interest, must be computed.

 ISBN 978-80-86943-75-6

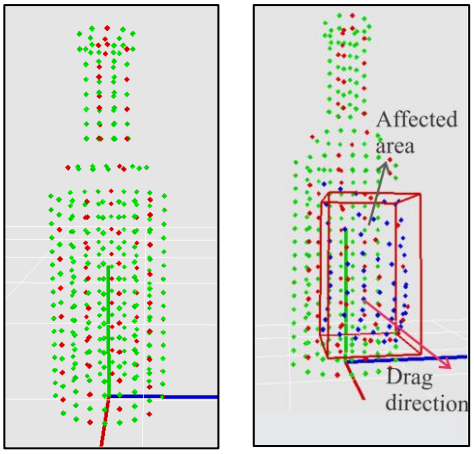


Figure 4: Generic primitive of a bottle. Control points are marked with red, while regular points are green. Modelled points are shown in blue.

2.3 Model Registration

In order to correctly transfer the particularities of the considered object to the GFP model, the involved shapes must be aligned. For this purpose, the scene model frame is considered to be the reference frame. The primitive will be initialy aligned according using a rigid body transformation. Therefore, the rotation R, translation T and scale s of the GFP has to be determined relative to the scene.

The scale factor s between the GFP and the segmented cluster is determined by approximating each cluster using a circumscribed sphere. The ratio between the radii of the shapes will act as a scale factor which will resize the primitive to the size of the object.

By subtracting the center of mass $m_M(x,y,z)$ of the primitive M from the center of mass $m_C(x,y,z)$ of the object cluster c_i, a relative translation T_{3x1} can be obtained. Finally, the rotation R_{3x3} is determined by incrementally rotating the primitive along all the three axes and minimizing a sum of Euclidean distances between the closest corresponding neighbor points of the two forms. Further, a fine model fitting is obtained through the *Iterative Closest Point* (ICP) algorithm [Zan94]. In Fig. 5(a), 5(b) and 5(c) the registration of a mug primitive is depicted.

Having computed all the prerequisite information for the final modeling process, the primitive cloud is aligned to the scene object using as:

$$M_{new}(i) = s \cdot R(M_{old}(i) + T), \quad i = 0 \ldots size(M), \quad (1)$$

where $M_{new}(i)$ are the new coordinates of the primitive point and $M_{old}(i)$ are the initial point coordinates.

2.4 Primitive modeling

The purpose of the modeling process is to release the primitive model from his generality. Through this step, the primitive will capture the *local* geometry information directly from the scene. Since initially no reliable information regarding the global structure can be identified, the modeling procedure occurs at a local level around each primitive point. If a particular vicinity lacks sensed information, the GFP will fill up the missing information with generic data, that is, the stored volumetric information in its shape. To make the entire process time efficient, the modeling process will occur only for *control* points while the regular points will be repositioned relative to these *control* points using a linear motion law. The basic principle underlying the primitive modeling step is known as *Active Contours* or *Snakes* [Kas1988]. In the initial formulation, a snake is a 2D curve which moves through an image domain driven by a set of energies computed based on particular image features. The behavior of a snake in the 3D space can be approximated with the weaving of a textile material. A major drawback of a snake is his inflexibility to topological changes. To cope with this issue, in [McI00], *topological snakes* (*T-Snakes*) are presented. Using an affine cell decomposition [All03], the authors succeded to create in this sense a framework that significantly extends the abilities of standard snake model.

The initial snake representation is described by a small circle in the 2D image domain [Kas1988], while its analogous in 3D is a sphere. Instead of using a sphere as the starting closed surface, we address the usage of a generic primitive, which already stores a rough structure of the considered object [Coc12b]. In comparison with [Coc12b], in this paper the movement of a contour point is constrained to only two directions, given by the normal direction. Thus, an important computations reduction is achieved.

In 3D, a *snake* structure is harder to control because of the extra degrees of freedom introduced by the third dimension. While for the 2D case there are only 8 possible moving directions, for 3D the number of candidate directions reaches 26 (given by the grid representation of the space). The *Active Contours* method tries to minimize a functional of energies $\varepsilon(c)$ in order to incrementally sculpt the initial contour c to an optimal final form as described in Eq. 2. Two types of energies are formulated for this purpose. The first type, E_{int}, is used to constrain the model deformation such that the structure integrity of the shape is kept at any moment during the modeling process, while the second one, E_{ext}, drives the considered modelled point to a its best candidate position:

$$\min\left(\sum_{0}^{N}(E_{int} + E_{ext})\right), \quad (2)$$

and

 ISBN 978-80-86943-75-6

$$E_{int} = \alpha(i) \cdot E_{cont}(i) + \beta(i) \cdot E_{curv}(i), \quad (3)$$

where N is the number of modelled feature points, α and β are the internal energy weight factors, E_{cont} represents the energy which ensures that the model surface is continuous (such that during the modeling process the newly rearranged points will not produce large gaps) and E_{curv} is the energy responsible for the smoothness of the surface. The α and β parameters are constrained by an empirical established threshold value. Thus, if the respective energy has a value below that particular threshold, then the respective weight factor (α and β), will be set 0, otherwise it will be 1.

As in Eq. 3, the internal energy is composed of two energies: E_{cont} and E_{curv}. They are used exclusively to constrain the movement of the points and at the same time to keep the model as compact and intuitive as possible. The internal energies are computed based on the first and second derivatives of the points which are to be modelled. The computation of the derivative of a certain snake point implies a neighborhood knowledge of the contour points. For example, in 2D the first derivative of a snake point is computed based on the previous and current position of the considered point. Similar, the second derivative is computed using the position of the previous, current and next point in the snake contour. In the 3D space, the previous respective the next snake contour points are evaluated as the closest, respective second closest nearest neighbor. Nevertheless, this approach is not always correct. At sharp corners this type of selection is erroneous. To avoid that, the relations between the points of the countour should be established using the mesh like representation. In this approach, the points making up the 3D contour will be the vertices of a mesh. Each face of the mesh describe a relation between minimium 3 points, thus the correct previous, respectively next countour points can be easly established. Considering these aspects, the mathematical formulation of the internal energy computed in a certain contour point p_i, can be stated as follows:

$$\begin{aligned} E_{cont}(i) &= |\frac{dc}{ds}|^2 + |\frac{dc}{dr}|^2 \\ &= ||p_i(s) - p_{i-1}(s)||^2 + ||p_i(r) - p_{i-1}(r)||^2, \end{aligned} \quad (4)$$

respectively,

$$\begin{aligned} E_{curv}(i) &= |\frac{d^2c}{ds^2}|^2 + |\frac{d^2c}{dr^2}|^2 + |\frac{d^2c}{dsdr}|^2 \\ &= ||p_{i-1}(s) - 2p_i(s) + p_{i+1}(s)||^2 + \\ &\quad ||p_{i-1}(r) - 2p_i(r) + p_{i+1}(r)||^2 + \\ &\quad ||p_{i-1}(s,r) - 2p_i(s,r) + p_{i+1}(s,r)||^2 \end{aligned} \quad (5)$$

where, s and r are the axis used to represent the 3D contour (as an topological manifold)

The process of minimizing the functional of energy $\varepsilon(c)$ implies resolving the following Euler-Lagrange ecuation:

$$\begin{aligned} E_{ext} + \alpha(s)|\frac{dc}{ds}|^2 + \alpha(r)|\frac{dc}{dr}|^2 - \beta(s)|\frac{d^2c}{ds^2}|^2 - \\ \beta(r)|\frac{d^2c}{dr^2}|^2 - \beta(s,v)|\frac{d^2c}{dsdr}|^2 = 0. \end{aligned} \quad (6)$$

The equation is true when the energies used in the process are in equilibrium. This also means that the contour has touched a relevant characteristic from the space (corner, edge, etc.).

The most important energy in our context, E_{ext}, is represented, in the 2D domain by the intensity of the greyscale candidate pixel. Similarly, in 3D, the pixel's intensity is equivalent to the neighboring density of points. The amount of neighbors lying in a given area is determined using the *kdtree* principle [Ben75a]. To avoid searching for the optimal candidate trough all 26 possible directions of a control point, given by the grid based representation of the space, the *normal* direction is used to reduce the search space to only 2 candidate directions. Fig. 5(d) shows the computed normal of a given generic primitive. Moving a control point exclusively along his normal directions deforms the surface in a natural and intuitively manner. By doing this, the overall processing time is considerably improved. Along the normal direction, the best position candidate for the control point is determined using the next set of rules:

- if the primitive candidate point m_{cd} is already lying in a dense region, move m_{cd} along both normal directions and find the first point with the number of nearest neighbors nn_{cp} closest to 0: $nn_{cp} \geq 0$;
- if m_{cd} has $nn_{cp} = 0$, search along both normal direction for $nn_{cp} > 0$; if, after searching, $nn_{cp} = 0$, freeze the control point in its initial position since no reliable surface information was found; if $nn_{cp} \neq 0$, set the position of the control point in the candidate position (with $nn_{cp} > 0$) closest to m_{cd}.

Based on this set of rules, the best candidate position of a given modelled point can be determined. In Fig. 5(e), the movement of a control point towards the local density information is illustrated. The final model of the GFP is depicted in Fig. 5(f).

To help create a more smoother surface and reduce the number of snake iterations, we propose an Euclidean distance based linear constrained. When a point is

 ISBN 978-80-86943-75-6

moved from an initial to a candidate position, all the neighbors within a vicinity radius equal to the Euclidean distance between the initial and the candidate position, will be gradually affected by a linear factor as:

$$M_{new}(i) = M_{old}(i) \cdot \left(1 + \frac{d_{curr}}{d_{max}}\right), \quad i = 0 \dots size(M) \tag{7}$$

where $M_{new}(i)$ and $M_{old}(i)$ are the new and old coordinates of the points lying inside the affected area of radius d_{max}. d_{curr} is the Euclidean distance between the current affected point and the control point. d_{max} is the Euclidean distance between the farthest point inside the affected area. Figs. 4 and 5(e) show the behavior of such Euclidean linear constraint principle, where a single point is dragged along the normal direction to find the correct surface location.

Inside the sphere described by the d_{max} radius, the points are modified accordingly to a computed ratio based on the distance between the neighboring and the initial primitive points. In this sense, the deformation is gradually applied, having the greatest effect on the neighbor points lying closer to the considered control point. Thus, the neighbor points at the margin of the sphere are slightly deformed. The proposed algorithm, not only is time efficient, but the movement of the points occur in a more intuitively way.

3 EXPERIMENTAL RESULTS

3.1 Setup

For evaluation purposes, two types of sensors have been used: a MS Kinect® RGB-D sensing device, used mainly for indoor testing, and a Point Grey Bumblebee® stereo camera for acquiring outdoor scenes. Both sensors output dense 3D information from the imaged scene. During tests, a constant illumination was ensured.

As GFP models, we have used the benchmark database in [Shi04a]. By using GFPs, the size of the database has been reduced from 1814 objects to only 142 general primitives. The GFPs were created as average shapes of the initial database. The point type assignment of all the primitives in the database was performed offline using the automated process, which took, in average, around 8 minutes for each model. During testing, all objects were placed on flat surfaces for detection and segmentation[1].

[1] The source code of the GFP approach is part of the ROVIS machine vision system, available at the svn repository `http://rovis.unitbv.ro/rovis/`. Please ask the authors permission for downloading.

3.2 Metrics

For evaluation purposes, an Euclidean based fitting measurement has been considered. Because the main goal of the modeling approach is to create a particular representation of a GFP, the distance between these two representations can be considered to be a similarity measure. Thus, by summing the distances between each point from the scene's objects and the nearest neighboring GFP point, the following fitting metric is obtained:

$$fit_{dist}(C,M) = \frac{1}{N}\sum_{i=0}^{n} \frac{1}{1 + \underset{fit_{dist}}{\operatorname{argmin}} ||C(i) - nn_i(M)||^2 \cdot \gamma}, \tag{8}$$

where $fit_{dist} \in [0,1]$ is the fitting error and N the number of points in the GFP. C and M are the PDMs of the clustered object and of the modeled GFP, respectively. $C(i)$ is the closest scene point to a GFP point $nn_i(M)$, while γ represents a scale factor. The better the GFP modeling is, the lower the value of fit_{dist} is.

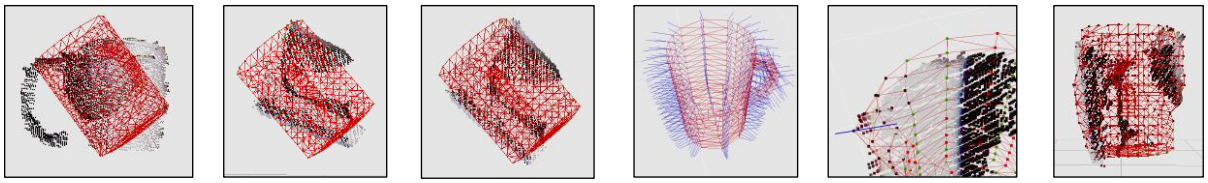
3.3 Case Study: modeling a Mug

From a total number of 410 primitive points describing a mug, only 203 (107 *control* points and 96 *regular* points) were moved during the modeling process. The rest of the points were assessed as optimal positions since they do not have any 3D data in their vicinity. The modeling computation time is approx. 680ms. The final mug model was obtained after only 21 modeling iterations. At each step, the contour was pushed through the scene with a 1mm offset further along the normal directions. Concerning the original formulation with the initial contour depicting a sphere, the number of iterations needed to obtain a shape similar to the modelled primitive is around 38 iterations. A comparative analysis of the energy evolution is illustrated in Fig. 6. The computation time, for the case of the sphere, reached 9sec.

For a different mug shape (e.g. highly deformed cup), the automatic point type assignment will generate a greater number of control points, directly influencing the computation time. Having a larger number of points describing the structure, the computation time increased from 0.6sec. to 0.9sec. for the GFP modeling.

For objects describing complex surfaces, keeping the primitive defined trough a low number of 3D points will cause the binding of some irregularities on the object surface. An up-sampling filter can be used to augment the initial primitive representation [Bre05a] and to produce high accurate models. When using denser representations, the fitting metric fit_{dist} is lower than the one obtained from sparse representations. Particular to the

 ISBN 978-80-86943-75-6



(a) (b) (c) (d) (e) (f)

Figure 5: GFP modeling of a mug. (a) Initial overlapping between the object's cluster and the GFP. (b) Euclidean distance based rotation approximation. (c) Fine alignment using ICP. (d) Primitive points normals described by the straight blue lines. (e) Searching along the normal directions. (f) Final annotated GFP.

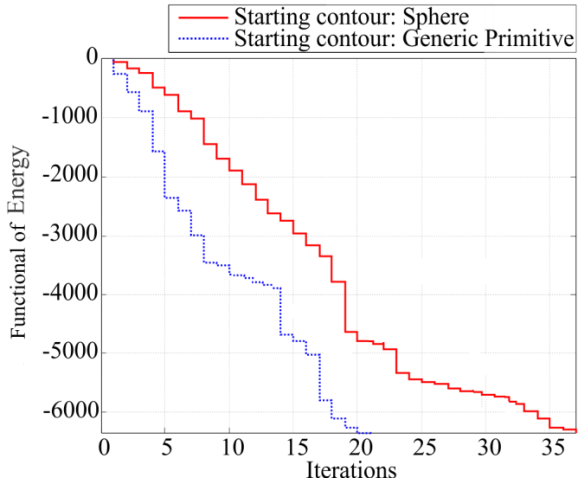


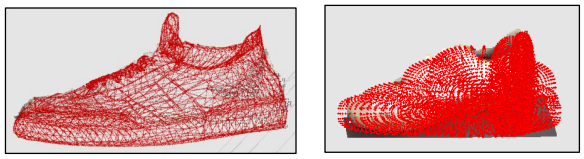
Figure 6: Number of iterations required for modeling using for different initial contours. A comparative analysis between the GFP approach and 3D active contours.

mug model, the overall volumetric error has been lowered to 5% by using the up-sampled representation of the same shape.

3.4 GFP vs. Superquadrics

Among the existing object volumetric estimation methods, *superquadrics* represent a real competitor for the GFP principle presented in this paper. A *superquadric* is a parametrized geometric shape obtained by the spherical product between two curves modelled through a series of parameters [Bar81a]. It resembles many geometrical models starting from simple cubes or cylinders and ending with complex ones such as toroid or hyperboloid. Because of the roughness provided by the generic shape, it is not desirable to use only one superquadric to approximate an object. In this sense, multiple joined superquadics produces a more precise and fine object [Coc12a].

By constantly changing the 11 parameters which defines a superquadric and by evaluating the newly obtained shape through an *in-out* function, the optimal model, given the point cloud representation of a particular object, can be obtained. This principle is considered to be fast because only a few parameters, which actually



(a) (b)

Figure 7: Estimated object volume described as a red point cloud. (a) GFP technique. (b) Multiple superquadrics approach.

deform the output shape, are controlled. The complex structure of a particular object can be approximated, in some case, with a very large number of superquadrics. Indeed, the global object volume will be more precise if this value is large, while the computation time will exponentially increased. On the other hand, by using a small number of superquadric models, only a rough volume will be obtained in the end. It can be stated that for simple objects, the superquadric based method is faster, whereas for complex models, the GFP technique excels both on precision, as well as on time efficiency.

One major advantage of superquadrics, in contrast to the GFP technique, is that it does not need pose normalization or any a priori knowledge regarding the class of the segmented object. This is the compromise that the GFP method is paying for its precision. In Fig. 7 a shoe modeling example using both methods is presented. Comparative numerical results are given in Table 1.

Method	Processing time [sec]	Fitting accuracy [%]
Superquadrics	4.79	0.7689
GFP	1.53	0.9762

Table 1: Comparative results between Superquadric based volume estimation and the GFP technique for the case of a shoe.

By using a primitive as an initial contour, the volumetric fit error of the GFP method is the smallest. Using superquadrics, the final volume is obtained as a reunion of a series of superelipses which best approximate the segmented object. Since the superquadric approach doesn't need pose normalization, the volumetric fit error is the only comparison parameter used during

 ISBN 978-80-86943-75-6

the comparative evaluation. From the grasping point of view, both methods output reliable grasp models, the difference being that the GFP technique, because of its accuracy, provides more grasping configurations. The grasping configurations were calculated using the GraspIt [Mil01a] methodology. A grasping simulation for a shoe object is depicted in Fig. 8.

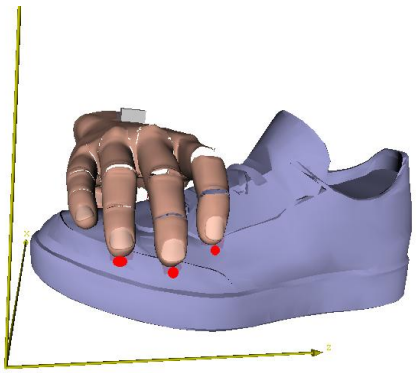


Figure 8: Grasp candidate configuration for a shoe. The world frame axes are coloured in yellow whereas the pressure points intersecting the GFP are marked with red.

4 CONCLUSIONS

In this paper, the GFP 3D object volumetric estimation technique has been presented. The goal of the approach is to estimate as accurately as possible the 3D structure of objects found in robotic mobile manipulation scenarios. As future work the authors consider the time computation enhancement of the proposed procedure through parallel computational devices (e.g. Graphic Processors), as well as the application of the method to other computer vision related areas, such as 3D medical imaging.

ACKNOWLEDGMENT

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.

5 REFERENCES

[All03] Allgower, E.L. and Georg, K. Introduction to Numerical Continuation Methods, in Classics in Applied Mathematics, Philadeplhia, USA: SIAM, 2003

[Bae02a] Baerentzen, J.A. and Christensen N.J. Volume sculpting using the level-set method, in Proceedings of the Shape Modeling, USA, 2002, pp.175-182.

[Bar81a] Barr A.H. Superquadrics and angle-preserving transformations, IEEE Computer Graphics and Applications, 1, pp.11-23, 1981.

[Bar02a] Museth, K. and Breen, D.E. and Whitaker, R.T. and Barr, A.H. Level set surface editing operators, in ACM Transactions on Graphics, 2002, pp.330-338.

[Ben75a] Bentley, J.L. Multidimensional binary search trees used for associative searching, in Commun. ACM, 18, No.9, pp.509-517, Sep. 1975.

[Bet00a] Betz, A. 3-D object reconstruction using spatially extended voxels and multi-hypothesis voxel coloring, in Proceedings of the Intern. Conf. on Pattern Recognition, USA, 2000, pp.1774-1782.

[Bre05a] Breitkopf, P. and Naceur, H. and Rassineux, A. and Villon, P. Moving least squares response surface approximation: Formulation and metal forming applications, in Computers and Amp Structures, 83, No.18, pp.1411-1428, 2005.

[Coc12a] Cocias, T.T. and Grigorescu, S.M. and Moldoveanu, F. Multiple-superquadrics based object surface estimation for grasping in service robotics, in 13th Intern. Conf. on Optimization of Electrical and Electronic Equipment, 2012, pp.1471-1477.

[Coc12b] Cocias, T.T. and Grigorescu, S.M. and Moldoveanu, F. Object volumetric estimation based on generic fitted primitives for service robotics, in VISAPP (2), 2012, pp.191-197.

[Cot95a] Cootes, T.F. and Taylor, C.J. and Cooper, D. and Graham, J. Active shape models-their training and application, in Comp. Vision and Image Understanding, 61, No.1, pp.38-59, 1995.

[Dil09a] Huebner, K. and Welke, K. and Przybylski, M. and Vahrenkamp, N. and Asfour, T. and Kragic, D. and Dillmann R. Grasping known objects with humanoid robots: A box-based approach, in Intern. Conf. on Advanced Robotics, 2009, pp.1-6.

[Gas01a] Ferley, E. and Cani, M.P. and Gascuel, J.D. Resolution adaptive volume sculpting, in Graph. Models, 63, No.6, pp.459-478, Nov. 2001.

[Hua12a] Huang, X. and Walker, I.D. and Birchfield, S. Occlusion-aware reconstruction and manipulation of 3D articulated objects, in Intern. Conf. on Robotics and Automation, 2012, 1365-1371.

[Kas1988] Kass, M. and Witkin, A. and Terzopoulos, D. Snakes: Active contour models, in Intern. Journal on Computer Vision, 1, No.4, pp.321-331, 1988.

[Kra11a] Krainin, M. and Henry, P. and Ren, X. and Fox, D. Manipulator and object tracking for in-hand 3D object modeling, in Intern. Journal of Robotic Research, 30, pp.1311-1327, Sep. 2011.

[Kra11b] Krainin, J. and Curless, B. and Fox, D. Autonomous generation of complete 3D object mod-

els using next best view manipulation planning, in Proceedings of the IEEE Intern. Conf on Robotics and Automation, Shanghai, China, May 2011.

[Mar09a] Marton, Z.C. and Goron, L.C. and Rusu, R.B. and Beetz, M. Reconstruction and verification of 3D object models for grasping, in Intern. Symp. on Robotics Research, 2009, pp.315-328.

[McI00] McInerney, T. and Terzopoulos, D. T-Snakes: Topology adaptive snakes, in Medical Image Analysis, 4, no. 2, pp. 73-91, 2000.

[Mil01a] Miller A. GraspIt!: A Versatile simulator for robotic grasping, Columbia University, 2001.

[Rus09a] Rusu, R.B. Semantic 3D object maps for everyday manipulation in human living environments, Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, 2009.

[Shi04a] Shilane, P. and Min, P. and Kazhdan, M. and T. Funkhouser, The princeton shape benchmark, in Shape Modeling International, 2004.

[Son11a] Song, Z. and Chen, Q. and Huang, Z. and Hua, Y. and Yan, S. Contextualizing object detection and classification, in Proceedings of the 2011 IEEE Conf. on Computer Vision and Pattern Recognition, USA, 2011, pp.1585-1592.

[Tan08a] Tangelder J.W. and Veltkamp R.C. A survey of content based 3D shape retrieval methods, in Multimedia Tools Appl., 39, No.3, pp.441-471, Sep. 2008.

[Zan94] Zhang, Z. Iterative point matching for registration of free-form curves and surfaces, in Intern. Journal on Computer Vision, 13, No.2, pp.119-152, Oct. 1994.

[Web10] Weber, C. and Hahmann, S. and Hagen, H. Sharp Feature detection in point clouds, in Proceedings of the 2010 Shape Modeling International Conference (SMI '10). IEEE Computer Society, Washington, DC, USA, 2010, pp. 175-186.

[Wat01] Watanabe, K. and Belyaev, A. G. Detection of salient curvature features on Polygonal Surfaces, in Computer Graphics Forum, 20, No.3, pp. 385-392, 2001.

 ISBN 978-80-86943-75-6