Exploiting Symmetries and Extrusions for Grasping Household Objects

Conference Paper *in* Proceedings - IEEE International Conference on Robotics and Automation · May 2015

DOI: 10.1109/ICRA.2015.7139713

7 authors, including:

Ana Huaman Quispe
Georgia Institute of Technology
7 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Marco A. Gutierrez
Universidad de Extremadura
11 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Henrik Iskov Christensen
University of California, San Diego
478 PUBLICATIONS **6,419** CITATIONS
SEE PROFILE

Heni Ben Amor
Arizona State University
56 PUBLICATIONS **434** CITATIONS
SEE PROFILE

Exploiting Symmetries and Extrusions for Grasping Household Objects

Ana Huamán Quispe[1] Benoît Milville[1] Marco A. Gutiérrez[2] Can Erdogan[1] Mike Stilman†
Henrik Christensen[1] Heni Ben Amor[1]

***Abstract*—In this paper we present an approach for creating complete shape representations from a single depth image for robot grasping. We introduce algorithms for completing partial point clouds based on the analysis of symmetry and extrusion patterns in observed shapes. Identified patterns are used to generate a complete mesh of the object, which is, in turn, used for grasp planning. The approach allows robots to predict the shape of objects and include invisible regions into the grasp planning step. We show that the identification of shape patterns, such as extrusions, can be used for fast generation and optimization of grasps. Finally, we present experiments performed with our humanoid robot executing pick-up tasks based on single depth images and discuss the applications and shortcomings of our approach.**

I. Introduction

The ability to grasp and manipulate objects is an important skill for autonomous robots. Many important tasks, e.g., assisting humans in household environments, require robots to reliably plan and execute grasps on surrounding objects. To generate plans for manipulation tasks, information about the shape of the object is required. A frequent approach to grasp planning is to use a database of polygonal meshes representing the different objects that the robot can manipulate [8]. Such information about object geometry can be used by grasp planners to synthesize an appropriate hand shape and orientation for physical interaction. While this approach is valid for structured domains with a small set of different objects, it does not scale to unstructured environments in which many objects may have never been seen before.

Other approaches to grasp planning employ depth cameras to acquire 3D point clouds of new objects, which in turn are used to generate grasps. Since the point clouds are acquired from a specific perspective, they only hold partial shape information about the visible frontal part. Using only partial point clouds to plan manipulation tasks can be very limiting, since many grasps involve placing fingers on opposite sides of an object. To fill any gaps and produce a complete point cloud, multiple images can be acquired by either iteratively moving the camera or the object. This process is time-consuming and introduces new challenges such as the precise matching of the individual point clouds of each view.

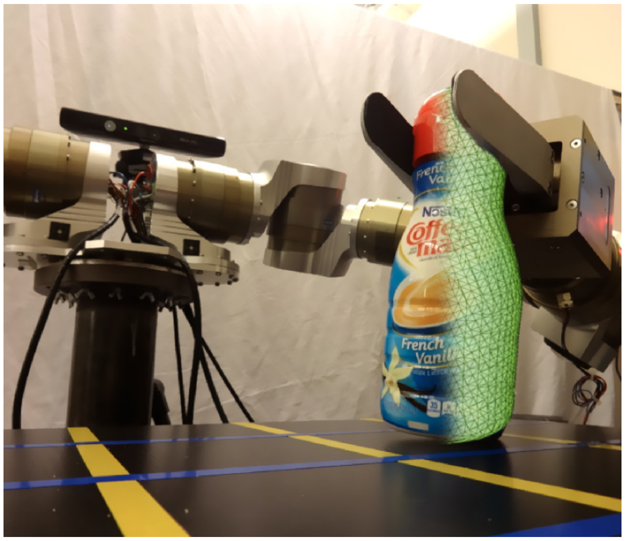

Fig. 1: Extracted information of rotational symmetries in the object is used to create a complete shape from a partial point cloud. The generated mesh is used by a grasp planner to generate a continuous set of grasps around the symmetry axis.

Alternatively, the robot can use geometric cues to predict the shape of the object in unseen regions. Through the analysis of inherent shape properties such as mirror symmetries and rotational extrusions, estimates of the complete point cloud can be generated from a single image. The extracted symmetry parameters can be used to extend observed shape patterns, e.g., the profile curve of an object, to occluded regions.

In this paper, we show how compact object representations for manipulation tasks can be generated from a partial point cloud. Given a *single* RGB-D image, we generate a complete mesh model of the observed object as well as additional shape information, e.g., axis of symmetry or superquadric approximations. We show that these compact representations can be later exploited for the fast synthesis of a continuous set of grasps. In turn, the set is used to plan robot manipulation tasks. Our approach builds both upon recent developments in symmetry-based [3, 18], as well as extrusion-based object representations [16]. Symmetry-based representations mirror observed object parts into occluded regions. Extrusion-based approaches, on the other hand, try

[1] Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA. ahuaman3@gatech.edu, cerdogan@cc.gatech.edu, benoit.milville@gadz.org, hic@cc.gatech.edu, hbenamor@cc.gatech.edu

[2] Robotics and Artificial Laboratory, Univesity of Extremadura, Cáceres, 10003, Spain. marcog@unex.es

to identify a two-dimensional profile which can be linearly or rotationally extruded to complete an object. In this work we show how symmetries and extrusions can be used to extract two different types of object representations, namely superquadric approximations and 2D shape profiles. We also show how these representations can used to generate grasps on the object.

The rest of this paper is organized as follows: Section II summarizes relevant literature. Section III introduces two compact object representations that are based on detecting symmetries and extrusions. Section IV shows how compact object representations based on extrusion patterns can be exploited for fast grasp planning with a small number of parameters. Section V presents experimental results of the object completion, as well as its application to robot grasping tasks. Finally in Section VI we discuss our approach and its advantages and shortcomings.

II. Related Work

For a robot to physically interact with its environment, algorithms for both grasp planning and perception are required. Traditional approaches for grasp generation are often based on fitting 3D CAD models to the observed scene [14, 15]. Such an approach, however, cannot be used to grasp novel objects since it requires accurate, prior knowledge about the shape. With the advent of depth cameras, various researchers have turned towards point cloud representations for perception and grasp planning. Huebner et al. [11] showed that bounding boxes computed from point clouds can be used to grasp novel objects. In a similar vein, Jiang et al. [12] proposed a so-called grasping-rectangle representation which can be used to infer the best grasp parameters given an RGB-D image of a novel object (given an offline training step). Przybylski et al. [21] showed simulation results in which a medial axis representation of objects can be used to find successful grasps without compromising on the approximation quality. Other than boxes and spheres [17], superquadrics [9] have also been considered for grasping applications given their compactness and ability to represent many diverse shapes with a limited number of parameters. Recently, Duncan et al. presented a fast hierarchical approach to fit superquadrics online [5].

On the side of grasp generation, a popular metric used to predict grasp robustness is the ϵ metric proposed by Ferrari and Canny [6]. While many popular grasp generators, such as GraspIt! use this metric to evaluate and refine the grasp search, it has been noted [4] that a grasp with a good metric does not translate to a robust grasp in a real-world execution. Researchers such as Hsiao [10] and Balasubramanian [1] have shown that grasps obtained using simple human heuristics can produce comparable or even better results when evaluated in a real, non-simulated environment.

A real world scenario - contrary to a simulated one - presents its own set of challenges: errors in perception, control and modeling must be considered and might render an optimal simulated grasp into an infeasible one. Regarding incomplete perceptual information, such as one-view point clouds for a given object, Bohg et al. [3] proposed a simple approach that exploits the symmetry of most common household objects to predict the full shape of an object on a tabletop scenario.

Following Bohg's observation that most common household objects present similar characteristics (such as symmetry, extrusion-like geometry and primitive shapes), we use them to approximate the shape of objects. This is also useful in the event of occlusion, in which a complete point cloud is not available.

III. Generating Compact Object Representations from Single RGB-D Images

In this section, we present two compact representations of objects that can be generated from partial point clouds. These representations can be used to plan grasps on objects involving regions of the point cloud that are currently invisible. As a result, a wider range of grasps can be planned, including, for example, side grasps which are based on an opposition of fingers placed at the front (seen) and the back (unseen) of the object.

We will first present a superquadric representation which is based on determining symmetries in point clouds. After that, we will turn towards a more detailed representation which makes use of rotational symmetries and linear extrusions to characterize an object.

A. Superquadric Representation

Superquadrics are a family of geometric shapes that can represent a wide range of diverse objects. The equation describing superquadrics in their canonical form can be written as

$$F(\mathbf{x}) = \left(\left(\frac{x}{a}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{b}\right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{c}\right)^{\frac{2}{\epsilon_1}} = 1. \qquad (1)$$

where a,b,c are the scaling factors along the principal axes, ϵ_1 is the shape factor of the superquadric cross section in a plane orthogonal to XY containing the axis Z, and ϵ_2 is the shape factor of the superquadric cross section in a plane parallel to XY. If a general transformation is considered, then the total number of parameters required to define a superquadric is 11 (the 6 additional being the rotational and translational degrees-of-freedom (DoFs) $\{x, y, z, \rho, \psi, \theta\}$). By minimizing the error between each point and the general superquadric equation, a shape that best fits the point cloud can be obtained:

$$\min_{k} \sum_{k=0}^{n} \left(\sqrt{abc} F^{\epsilon_1}(\mathbf{x}; \Lambda) - 1 \right)^2 \qquad (2)$$

As mentioned in Section II, superquadrics have previously been used to generate grasp configurations for simple objects [2, 22]. Most of these approaches assume that the complete shape of the object is given or that the parameters can be learned beforehand. However, when working with depth cameras this is not a reasonable assumption to make. In recent work, Duncan et al. [5] presented a superquadric fitting

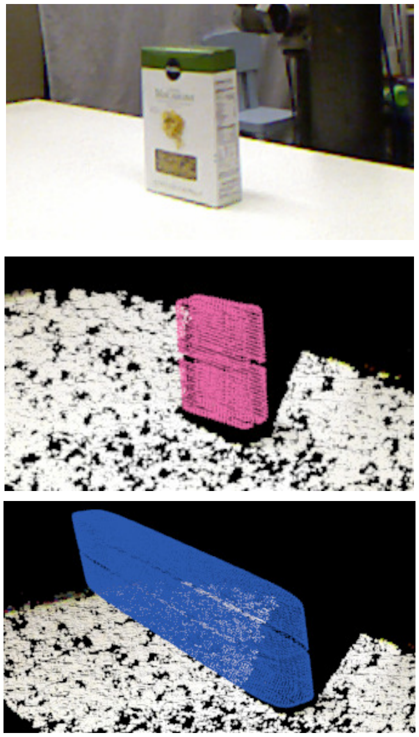

Fig. 2: An example for the superquadric fitting with symmetry analysis (middle) and without it (bottom).

approach which uses a voxel representation to reduce the computational complexity of the task. We found that this approach worked well when the segmented point cloud of the object had a good viewing point (i.e. the front, side and top of the object were seen). For point clouds in which only one side of the object was seen (i.e. only front), the performance quickly deteriorated, producing fitting parameters that in many cases exceeded greatly the original dimensions of the objects. While this could be partially alleviated by hard-coding limits in the dimension of the axes, this is not practical when dealing with novel objects, for which we might not know the dimensions beforehand.

Inspired by work presented by Bohg et al. [3], we added an additional pre-processing step to the superquadric minimization process. Instead of using the original point cloud as input, we generated a mirrored version (see Fig. 2) by finding an optimal symmetry plane perpendicular to the table where the object resides (for more details of this process, please refer to the original paper [3]).

B. Object Completion from Extrusions

Planning task-specific grasps requires information about the complete shape of the object to be manipulated. Many household objects are based on extrusions. Indeed many modelling and manufacturing systems use linear and rotational extrusions in a hierarchy to generate the models used for manufacturing. Uncovering extrusions in partial point clouds can therefore help to generate a complete point cloud from a partial observation. In addition, this knowledge can be used to create a large set of feasible grasps from which a planner can choose suitable candidates for task execution. For example, detecting the axis of symmetry in a rotationally symmetric object allows us to rotate any feasible grasp around this axis.

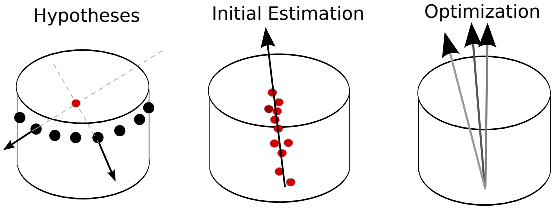

Fig. 3: The three steps used for optimizing the axis of extrusion. First, we generate hypotheses by analysing pairs of points. The resulting estimates are used to produce an initial estimate of the axis of extrusion. Finally, optimization is used to improve the extrusion axis.

In this paper, extrusion detection is performed using a three-step approach, see Fig. 3 for an overview of the approach using rotational extrusions. In the first step, we use points from the partial point clouds to generate hypotheses for the extrusion axis. In the case of rotational extrusions, we randomly sample pairs of points and use the normal of each point to create a line. Each pair of lines is intersected and the resulting point is used as a hypothesis for the axis of extrusion. Fig. 3 shows an example for points sampled from a cylindrical object. To account for noise, we use the midpoint of the line connecting the closest points, in case the two lines do not intersect. The collected hypotheses points are then used to create an initial estimate of the axis of extrusion. To this end, we fit a line into the set of hypotheses using linear least-squares. The RANSAC [7] algorithm is further used to reduce the influence of outliers. Given this initial estimate, we perform optimization to produce a more accurate axis of extrusion. Specifically, we use the dynamic hill climbing algorithm [23] to search for an axis of extrusion which reduces the dispersion of points along the profile of the object. In every iteration, the axis of extrusion is used to rotate all points of the partial point cloud back onto a plane. We then estimate the density of the points using a kernel density estimator [20]. By maximizing the density using the hill climbing algorithm, we can reduce the dispersion of the projected points, thereby recreating the profile of the object. However, performing a kernel density estimation in each step of the optimization process is computationally expensive and does not scale to large point clouds. The following method is, therefore, a discrete approximation of the kernel density, which produced accurate results in practice while at the same time being fast.

We create an approximation of the kernel density estimator by creating a grid over the projected point cloud. The number of cells used in our experiments varied between 5 and 30 cells in each dimension. For each cell $i \in \{1, .., M\}$ we count the number of points c_i that lie within. We then calculate the average of the differences to neighbouring cells $j \in \{1, .., N\}$. The overall objective function of the optimization can be

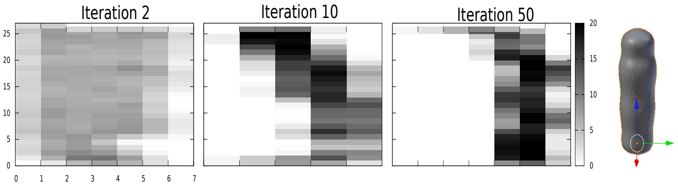

Fig. 4: Density estimation at different stages of optimization. At the beginning of the optimization, the projected points are highly dispersed. The axis of extrusion is then changed to minimize the dispersion, such that the outer profile of the object emerges as can be seen in iteration 50. On the right side we can see the object to which the profile belongs.

written as

$$E = \frac{1}{M}\frac{1}{N}\sum_{i}^{M}\sum_{j}^{N}||c_i - c_j|| \tag{3}$$

where E is the energy to be minimized. Fig. 4 shows three iterations during the optimization of the axis of extrusion. Dark areas correspond to regions of high density of points, while lighter areas correspond to low density regions. In early iterations, the estimate of the axis does not produce a clear profile when points are projected (rotationally) onto a plane. In iteration ten, we can see that high density regions start forming. After fifty iterations, an approximate profile of the object starts to emerge.

After optimization is finished, we regard the projected points as the profile of the object and rotationally extrude them around the axis of extrusion to generate a complete point cloud. Fig. 5 shows a set of household objects, the recorded depth images, as well as the reconstructed complete meshes. Given the completed point cloud, we reconstructed the meshes using Poisson surface reconstruction [13].

For the case of linear extrusions along an axis, a different method for the estimation of the initial axis of extrusion needs to be used. For linear extrusions, we compare the normal vectors of pairs of points and generate a hypothesis if the difference between the normals is below a threshold. The resulting set of hypothesis can then be clustered, such that each cluster represents a possible axis of extrusions. For example, for a box, up to six clusters can be found.

Note, that in our approach we use a point cloud to represent the profile of an extrusion. For revolute objects, the profile defines the outer curve of the object, which can be rotated around the axis of extrusion to generate the complete shape. For linear extrusions, the cloud represents the basic 2D shape which can be extruded to form the object. Fig. 6 shows the extracted object profiles for objects with linear extrusions.

IV. Using Compact Object Representations for Grasp Planning

Grasp planning greatly benefits from the completed point clouds. A complete point cloud can be triangulated and used as an input to existing grasp generation and planning algorithms. In contrast to the partial point cloud, the completed and triangulated mesh can be used to perform collision checks and evaluate grasp quality using existing metrics. In contrast, traditional grasp quality metrics cannot be directly applied to partial point clouds. Similarly, having a complete mesh allows a grasp planner to evaluate a large variety of grasps, which can then be pruned based on task constraints. However, generating many grasps often involves repeated applications of grasp optimization methods which can be computationally demanding, in particular in the presence of many degrees-of-freedom in the robot arm and hand. Extracted shape information from extrusions can be used to improve the efficiency of this process by significantly reducing the number of degrees-of-freedom of the problem.

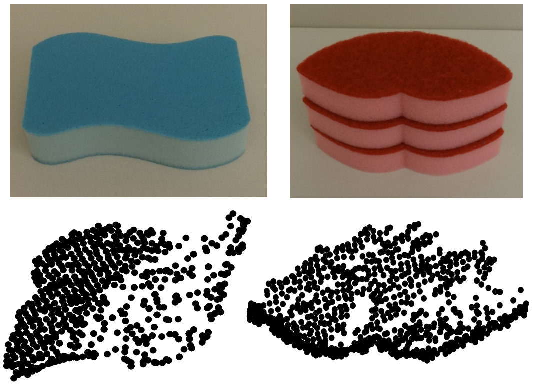

Fig. 6: Extracted object profile for the linearly extruded objects. The extracted profiles are used to create a complete point cloud.

The main insight of this section is that hand shapes during object grasping are invariant to movements along the axis of extrusion. As long as the robot hand moves along the axis of extrusion, no expensive replanning of the hand shape is necessary. In the case of linear extrusions, the robot hand can move up and down the axis of extrusion without having to change the hand shape. Similarly, in the case of rotational extrusions, the hand can be rotated around the axis of extrusion. This knowledge can be exploited during grasp generation in order to turn each single detected grasp into a continuous set of grasps. Subsequently, we present a specific example how information about extrusions can be used to reduce the dimensionality and complexity of a grasp re-planning task.

Fig. 7a shows a scenario, in which a grasp is executed on a rotationally symmetric object. The grasp has a low manipulability index which is not sufficient to achieve the task constraints. Typically, this means that a new grasp and arm pose needs to be planned, which involves (sampling-based) optimization in the high-dimensional space of joint angles.

Given that the grasp is performed on a rotationally symmetric object, the grasp generation can be modeled as an inverse kinematics problem where the goal is to determine an arm configuration $\boldsymbol{q}$ that is collision free. The output is constrained by the end-effector position on the object and the corresponding inverse kinematics solution. The end-effector pose $\boldsymbol{x}$ can be parametrized by (1) the rotation around the axis of extrusion ϕ and (2) the distance along the axis of extrusion

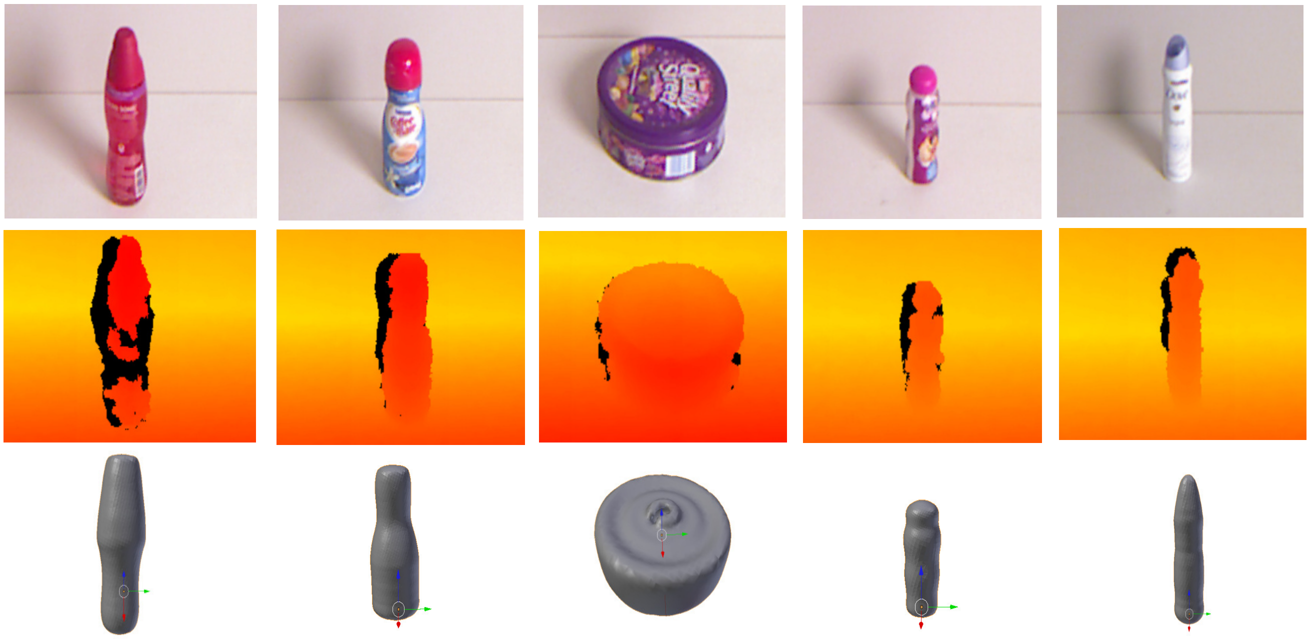

Fig. 5: Reconstruction of rotationally symmetric household objects. The top row shows a photo of the object. The middle row shows the corresponding depth image recorded using a Microsoft Kinect. The bottom row shows the completed mesh. Reconstruction was performed from a single image through the analysis of extrusions.

α, $\boldsymbol{x} = pose(\phi, \alpha)$. The inverse kinematics solution $\boldsymbol{q}$ with a 7-DoF arm for an end-effector pose $\boldsymbol{x}$ can be parametrized by an additional variable θ which represents the angle between the wrist-elbow-shoulder plane and the ground, $\boldsymbol{q} = IK(\boldsymbol{x}, \theta)$.

At each iteration i, the new arm position is computed using an updated grasp position from the parameter space $\{\phi_{i-1} \pm \delta_\phi, \alpha_{i-1} \pm \delta_\alpha\}$ and the corresponding inverse kinematics parametrized by $\{\theta_{i-1} - \delta_\theta, \theta_{i-1}, \theta_{i-1} + \delta_\theta\}$. Let $\mathbb{P}$ represent the full space of the variables ϕ, α, and θ. The algorithm iteratively updates these parameters by determining which tuple leads to the maximum manipulability [19]. This is realized by solving for the following objective

$$\boldsymbol{q}_i = \underset{\{\phi,\alpha,\theta\}\in\mathbb{P}}{\operatorname{argmax}} \sqrt{det(J(\boldsymbol{q})J^T(\boldsymbol{q}))} \tag{4}$$

where $\boldsymbol{q} = IK(pose(\phi, \alpha), \theta)$. The sequence in Fig. 7 shows several snapshots during this optimization. In this scenario, the robot grasps a rotationally symmetric bottle. The initial random grasp sample in Fig. 7a yields a manipulability of 0.268 which is then improved in Fig. 7d leading to a value of 0.540. To optimize the manipulability, the planner iteratively changes the grasp position on the robot with the ϕ and α parameters, and the inverse kinematics parameter θ. This optimization can be performed efficiently since, the high-dimensional configuration space of the hand does not need to be represented thanks to the extracted symmetries. Instead, a three-dimensional space of parameters $\{\phi, \alpha, \theta\} \in \mathbb{P}$ is used.

V. Experimental Results

In this section, we present a set of experiments which we conducted to evaluate the proposed approach. The first set of experiments focuses on the complexity and accuracy of point cloud completion when generating compact object representations. The second set of experiments shows the application of the approach to grasp planning on a humanoid robot. The used humanoid robot is based on Schunk LWA3 arms with 7 DoF. A Schunk gripper with a maximum aperture of 7cm was used. Partial point clouds were recorded using a Microsoft Kinect camera.

A. Accuracy of Fit

We first analyzed the accuracy of fit of the two presented compact object representations. For extrusions, we collected a set of rotationally symmetric meshes from internet databases from which we generated partial point clouds. We then cut out a partial point cloud representing 30% of the data and simulated Kinect-like noise by adding holes and noise to the dataset. The partial cloud was then completed using the extrusion detection methods from Sec. III-B. To measure the accuracy, we compared the completed clouds to the original mesh of the object. On average, the approach produced an error (distance of points to mesh) of 2mm, where objects had a diameter between $10 - 20$cm. Analysis of the extrusions required on average 200ms.

For superquadric fitting we conducted a similar experiment. However, in this case we noticed larger variations in the reconstructed shapes depending on the perspective of the camera to the object. We therefore placed each object at one of five different locations in front of the camera and measured the run time of the algorithm including symmetry analysis and without it. As depicted in Tab. I, the fitting time is shorter when additional points are added via symmetry analysis. While this

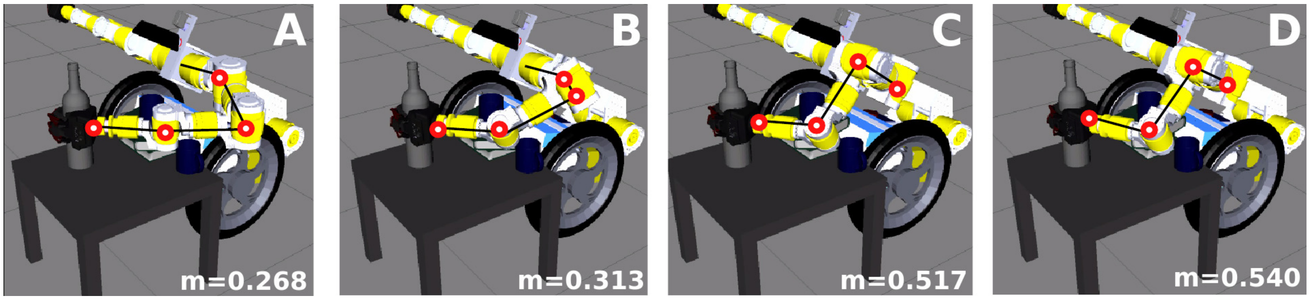

Fig. 7: Grasp manipulability optimization along the axis of extrusion. Since the object is symmetric, the same hand configuration can be rotated around the object (A-B, C-D). At the same time, the extra DOF in the inverse kinematics solution is also utilized to maximize manipulability (B-C).

may seem unintuitive, we found that the superquadric shape has more constraints when considering mirrored points. As a result, the optimization process required for fitting quickly settles on a good solution.

TABLE I: Comparison of fitting times

Object	Input	P1	P2	P3	P4	P5	Avg. Time
Apple	Symmetry	0.02	0.13	0.01	0.06	0.07	**0.05s**
	Plain	0.14	0.17	0.06	0.06	0.06	**0.098s**
Milk	Symmetry	0.20	0.07	0.03	0.05	0.04	**0.078s**
	Plain	0.42	0.56	0.27	0.53	0.06	**0.368s**
Jam	Symmetry	0.06	0.11	0.13	0.08	0.21	**0.118s**
	Plain	0.08	0.29	0.10	0.08	0.11	**0.132s**
Raisins	Symmetry	0.29	0.25	0.31	0.14	0.27	**0.252s**
	Plain	0.36	0.40	0.43	0.43	0.32	**0.388s**
Creamer	Symmetry	0.15	0.15	0.22	0.13	0.14	**0.158s**
	Plain	0.65	0.09	0.39	0.26	0.29	**0.336s**

B. Robot Grasping Experiments

Next, we conducted an experiment in which a humanoid robot was used to grasp household objects located in front of it. We also placed several other objects as clutter on the table. Given the depth image all objects were reconstructed using compact object representations. After that, the robot planned and executed grasps using the normal at a point as an approach direction and the method described in Sec. IV for ensuring manipulability and obstacle avoidance. We conducted trials with 4 objects which were placed at 4 different locations on the table. Each trial was repeated three times. A grasp was regarded successful if the robot was able to lift the object.

Tab. II summarizes the results of the experiment. We can see that the approach using superquadrics performs well on most objects with the exeption of the roll. In contrast, the extrusion-based approach seems to have difficulties with a specific location (C3). Analyzing the robot executions, we found that superquadric approach typically leads to approximate shapes which are slightly larger than the original object. Hence, the executed grasp includes a ”buffer” zone that allows it to succeed in the presence of sensor and calibration noise. Grasps planned for the shapes generated by the symmetry detection, however, are tighly fit to the object. This often lead to premature contact with the object during grasp execution. In Tab. II we also see the number of different grasps found using the two approaches. We can see that the symmetry based approach leads to a larger number of different grasps, due to the invariance along the axis of extrusion. Images of the executed grasp and the experimental setup can be found in Fig. 8.

TABLE II: Experimental results, 3 trials per object per location

		Location	Creamer	Dove	Roll	Micro
Success	**Extrusion**	**B4**	100%	100%	0%	100%
		C3	0%	0%	0%	0%
		C4	100%	100%	100%	100%
	SQ	**B4**	100%	100%	0%	100%
		C3	100%	100%	66%	100%
		C4	100%	100%	0%	100%
Grasps	**Extrusion**	**B4**	1040	900	1200	640
		C3	800	400	2200	800
		C4	1270	320	800	1020
	SQ	**B4**	11	11	7	11
		C3	7	3	1	7
		C4	10	5	5	13

VI. Discussion and Conclusion

In this paper we introduced methods for generating compact and complete object representations that are particularly useful for robot grasping applications. The approach exploits natural patterns found in many shapes, e.g., symmetries, linear extrusions, and rotational extrusions to generate a complete mesh from a single depth image. We also showed that the extraction of this information can be used to improve the efficiency and quality of the grasp planning step. The work presented in this paper can be seen as a first step towards shape priors that can be used by a robot to generate hypotheses about the shape of an object in invisible regions. Other cues, such as curvature and texture may also be helpful in predicting the complete shape from partial observations. At the moment the introduced approach is limited to household objects, which are often based on linear and rotational extrusions. However, it can also be extended to work in a hierarchy to complete more complex objects. In future work, we hope to investigate this aspect in more detail.

The performed robot experiments showed that the approach can be used to create a variety of grasps. In particular, we can generate grasps that extend to parts of the object that are not seen. This is in contrast to other methods which limit the approach direction of the robot to the visible part of the object. We have shown in the experiments that the method can be used to reconstruct objects in a cluttered scene without prior

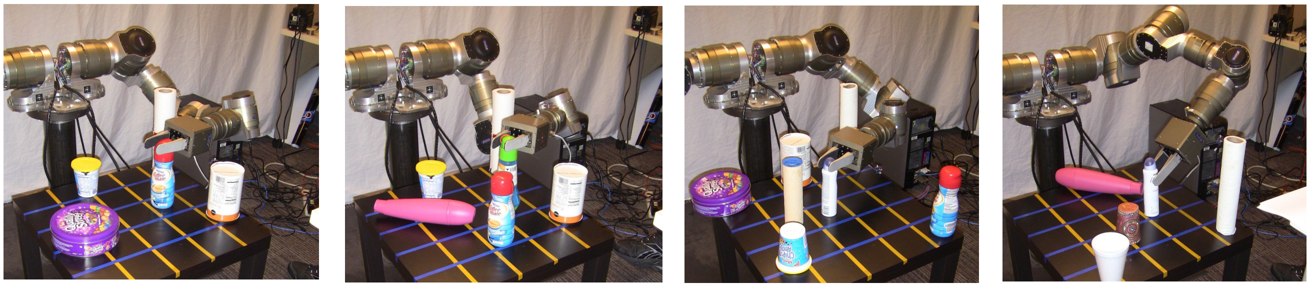

Fig. 8: Grasps on household objects generated via grasp planning on compact object representations. All objects on the table were reconstructed. Objects that were not grasps were regarded as obstacles to be avoided during the manipulation task.

information. Yet, the additional information gained by creating complete meshes also imposes additional requirements on the accuracy of the robot controller. Planning grasps with more accurate reconstructions of the observed object means that the robot needs to be very precise in the task execution. So far, we do not have a model of the inherent sensor and actuation noise. We hope to investigate Bayesian approaches to object fitting, which would allow us to use information about the uncertainty during task execution.

Acknowledgments

This work is dedicated to the memory of Mike Stilman, whose enthusiasm for making robots do cool things will always be remembered.

References

[1] R. Balasubramanian, L. Xu, P. D. Brook, J.R. Smith, and Y. Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[2] G. Biegelbauer and M. Vincze. Efficient "3d" object detection by fitting superquadrics to range image data for robot's object manipulation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.

[3] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales. Mind the gap: Robotic grasping under incomplete observation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[4] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2010.

[5] K. Duncan, S. Sarkar, R. Alqasemi, and R. Dubey. Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[6] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1992.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[8] C. Goldfeder and P. Allen. Data-driven grasping. *Autonomous Robots*, 2011.

[9] C. Goldfeder, P. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.

[10] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[11] K. Huebner and D. Kragic. Selection of robot pre-grasps using box-based shape approximation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.

[12] Y. Jiang, S. Moseson, and A. Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[13] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. of the Fourth Eurographics Symposium on Geometry Processing*, 2006.

[14] U. Klank, D. Pangercic, R.B. Rusu, and M. Beetz. Real-time cad model matching for mobile manipulation and grasping. In *9th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2009.

[15] D. Kragic, A. Miller, and P. Allen. Real-time tracking meets online grasp planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2001.

[16] O. Kroemer, H. Ben Amor, M. Ewerton, and J. Peters. Point cloud completion using extrusions. In *Int. Conf. on Humanoid Robots(Humanoids)*, 2012.

[17] A. Miller, S. Knoop, H. I. Christensen, and P. Allen. Automatic grasp planning using shape primitives. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[18] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 560–568, New York, NY, USA, 2006. ACM.

[19] Y. Nakamura and H. Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of dynamic systems, measurement and control*, 1986.

[20] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 1962.

[21] M. Przybylski, T. Asfour, and R. Dillmann. Unions of balls for shape approximation in robot grasping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[22] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.

[23] D. Yuret and M. de la Maza. Dynamic hill climbing: Overcoming the limitations of optimization techniques. In *Second Turkish Symposium on Artificial Intelligence and Neural Networks*, 1993.