Reconstruction and Verification of 3D Object Models for Grasping

Chapter · April 2011

DOI: 10.1007/978-3-642-19457-3_19

4 authors, including:

Zoltan Csaba Marton
German Aerospace Center (DLR)
101 PUBLICATIONS **4,041** CITATIONS

SEE PROFILE

Radu Bogdan Rusu
Open Perception
67 PUBLICATIONS **11,912** CITATIONS

SEE PROFILE

Michael Beetz
Universität Bremen
480 PUBLICATIONS **14,131** CITATIONS

SEE PROFILE

Reconstruction and Verification of 3D Object Models for Grasping

Zoltan-Csaba Marton, Lucian Goron, Radu Bogdan Rusu and Michel Beetz

Abstract In this paper we present a method for approximating complete models of objects with 3D shape primitives, by exploiting common symmetries in objects of daily use. Our proposed approach reconstructs boxes and cylindrical parts of objects from sampled point cloud data, and produces CAD-like surface models needed for generating grasping strategies. To verify the results, we present a set of experimental results using real-world data-sets containing a large number of objects from different views.

1 Introduction

In this paper we discuss a method on how to obtain suitable complete 3D representations for typical objects in a kitchen table cleaning scenario. Our approach differentiates from similar research initiatives in the sense that we do not use databases containing predefined object models in combination with machine learning classifiers to address the object reconstruction problem. Instead we generate CAD-like quality surface models that are extremely smooth and can directly be used to infer

Zoltan-Csaba Marton
Technische Universitaet Muenchen, Boltzmannstr. 3, 85748 Garching b. Muenchen e-mail: marton@cs.tum.edu

Lucian Goron
Technische Universitaet Muenchen, Boltzmannstr. 3, 85748 Garching b. Muenchen e-mail: goron@cs.tum.edu

Radu Bogdan Rusu
Technische Universitaet Muenchen, Boltzmannstr. 3, 85748 Garching b. Muenchen e-mail: rusu@cs.tum.edu

Michael Beetz
Technische Universitaet Muenchen, Boltzmannstr. 3, 85748 Garching b. Muenchen e-mail: beetz@cs.tum.edu

grasping points by exploiting shape symmetries. To show the applicability of our approach, we make use of a mobile manipulation platform (see Figure 1) to acquire 3D point cloud datasets, and show segmentation results for table planes together with sets of unseen objects located on them. The result of our mapping pipeline includes a complete set of 3D representations that can be used to compute grasping points.

Most grasping paradigms, like [11] and [5], require a CAD-like representation of the objects, which is difficult to obtain from sensed data. The two main approaches to produce these representations rely on image or depth information. In the first case, usually a model from a database is matched to the image as in [3, 6], while in the latter, a more flexible combination of shape primitives [17], superquadrics [1, 21] are fit, or a triangulation of the surface [13, 9] is performed.

The most accurate 3D sensing devices usable by robots are laser scanners, which can have an accuracy in the millimeter range for some surface types, but they can provide only partial information about the object (one side, from a single viewpoint), and they have problems when dealing with shiny (eg. metal or ceramic) objects. Unfortunately, these are quite common in every day manipulation tasks, like those a personal assistant robot would encounter in a kitchen, for example. These objects need to be segmented into distinct clusters for grasping, but in some cases an object appears in two separate clusters because of the previously mentioned problems, or occlusions. Thus a mug is typically represented by point clouds in two semi-circular parts and with only a few points on the handle (see Figure 2).

Fig. 1 The mobile manipulation platform used for the experiments presented herein.

Some simplifications have to be made in order to be able to approximate the occluded parts of objects. Our assumption is that most objects have a vertical plane or axis of symmetry (eg. mugs, boxes, bottles, jars, plates, pans, bowl, silverware, etc.), are composed of planar and cylindrical parts, or can be roughly approximated by such, and are representable as groupings of planar patches and cylindrical parts. Because of their vertical symmetries, these models can be obtained by analyzing the footprint of the objects on their supporting plane, and detecting linear and circular segments in it (see Figure 3). We call these objects with sides that are perpendicular to their bottom *standing objects*. Spherical, toroidal and conical parts will be approximated with cylinders, but this would explain the data only partially, so we can recognize the cases when these problems are encountered. This approach provides straightforward ways of correction (like merging two components if they belong together) and verification of the correctness of the fitted model at each surface area unit.

The main contributions of our approach are:

- exploiting common symmetries to produce suitable completed models for grasping applications from single view scans;
- creating a complete model from a single view for *standing objects*, together with a measure for verifying the approximated reconstruction;
- a way to deal with 3D over-segmentation of objects produced by occlusions and measurement errors.

The remaining of the paper is organized as follows. In the next section an overview is given on the current approaches, followed by the presentation of our method in Section 3. The experimental results are analyzed in Section 4, followed by our conclusions and a discussion on future work in Section 5.

Fig. 2 Typical one-side scan of a mug, where the rim and handle didn't return enough measurement points, thus the mug will be over-segmented into two separate connected components, and the handle is hard to detect.

2 Related Work

A computer vision and machine learning based method is used in [16] to train classifiers that can predict the grasping points in an image. This is then applied to images of unseen objects. To obtain 3D positions of grasping points, the authors use stereo cameras, but their approach works reliably only to the extent provided by the training data. Another issue is the segmentation of objects, since only grasp points are provided with no information about what objects are in the scene and to which of them do the identified points correspond. In [2] an accurate line laser and a camera builds models and identifies grasping points for novel objects with very encouraging results. However the system was tested only on two objects, thus its scalability is not clear. Available models of complex objects are decomposed into superquadric parts in [1, 21], and these models are fit to a point cloud. This however needs a database of models, and moreover, their decomposition into superquadric components, which is often difficult to obtain.

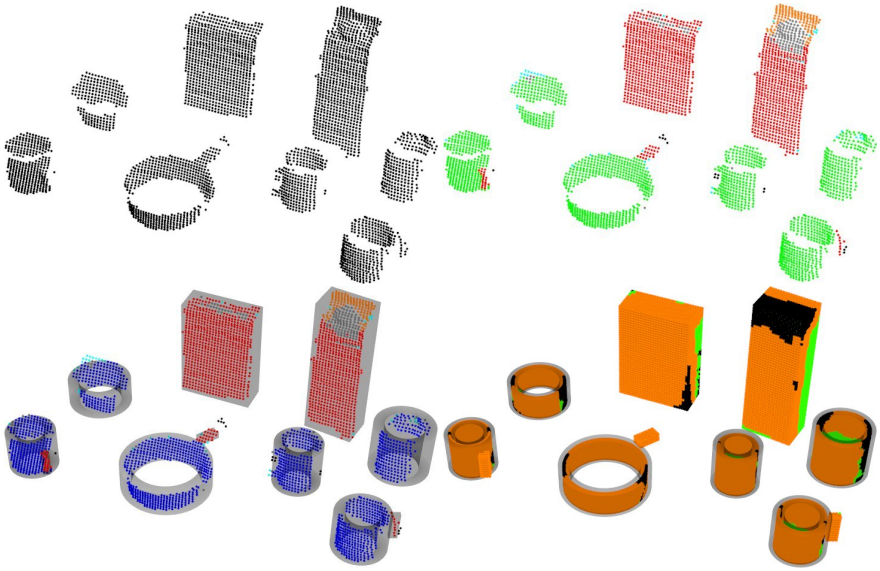

Fig. 3 The most important processing steps of our method are illustrated in the pictures, from left to right, top to bottom: a) measurement points returned from objects on the table are segmented from the complete scan, and sparse outliers are removed; b) connected components are identified and shape primitives are fit to its footprint silhouettes (points belonging to a vertical plane are marked with red and orange, while those belonging to cylinders are green); c) the 3D shapes (boxes and thick cylinders) are obtained from the corrected shape primitives; d) points are generated on the shapes and verified for correspondence to measurements and visibility (green points are invisible from the current viewpoint, orange points are verified by measurements while black points are in visible free space, meaning that the model of the object should not include those parts of the shapes.

In purely computer vision based approaches either features like [8] or [7] are used to find matches between parts of a scene and a database of object images. The problem with these kinds of approaches is that they only work for objects that are in the database, and since no knowledge about the 3D information is known, the system can easily make mistakes and return false positives (e.g., a cereal box containing a picture of a beer bottle printed on it might get recognized as a bottle of beer). Another approach to obtain 3D information directly form camera images, is to project CAD models from a database to the image and search for good fits in the edges domain, for example like in [19]. While this is a more direct method, it's still dependent on a database of different CAD models. Acquiring these automatically from the Internet has been explored in [6], but obtaining the models of all the objects in a scene requires selecting all the possible good fits of the models to the image, which takes considerable amounts of time. In our approach, we do not have these problems, since we have access to the 3D information directly, so we can use a bottom up approach for reconstructing the object model from the geometry data, with lower computational constraints than the initiatives mentioned previously.

Hough transforms are sometimes used to detected geometric shapes like cylinders in [12], or planar patches in [20] in point clouds. However they are not as popular as sample consensus based methods like RANSAC [4], since a parameter space has to be constructed for each shape type separately, which complicates things for more complex models. In the sample consensus paradigm, the data is used directly to compute best-fit models. We are using RANSAC because it allows the definition of different models for more complex geometric shapes

A similar sample consensus based approach for model decomposition is presented in [17], where a set of 3D geometric primitives (planes, spheres, cylinders, cones and tori) are fit to noisy point clouds. Since the point clouds presented there are complete, the authors don't need to reconstruct the missing parts. To solve this problems in our case, we are fitting planar and cylindrical shapes, and exploit the vertical symmetries present in most objects to reconstruct their occluded parts. In [18] the authors describe a method for detecting and verifying symmetries in point clouds obtained from a single viewpoint which works very well for nicely segmented objects, however the problem of under- or over-segmented objects remains.

3 Object Model Reconstruction

Our system takes a single view of a table scene as input, and extracts the table together with the points returned from the objects that are on it. The data is cleaned using a statistical analysis of point densities, and clustering is performed to segment the different objects into separate regions.

These regions are then reconstructed, and the fitted models are corrected, evaluated, and used for detecting cases of over-segmentation (i.e., when objects are split into multiple regions).

In the next subsections we present the aforementioned steps in more detail.

3.1 Table Detection

The initial step of our method is to locate the table and the objects on it. This problem falls outside the scope of this paper, and our previous work on Object Maps [15] have already presented the robust localization of important furniture parts in a kitchen in more detail. After the location of the table is obtained, a scan can be made of the area, and a restricted planar search can be performed in order to obtain the model of the table as presented in Figure 4 (see [15] for more details).

3.2 Object Footprints

After the points that are above the table are obtained, the sparse outliers are removed using a statistical analysis of the point densities in each points neighborhood as detailed in [14]. The remaining points are projected along the normal of table plane to obtain 2D clusters, which are then grouped based on a connectivity criterion.

In each cluster a set of boundary points are identified as those which have a maximum angle between the vectors pointing towards their neighbors (from the same region) that matches or exceeds the opening of a what would be a straight line (that is 180°). The neighbors of these points are also marked as boundary points in order to provide a contingent set of boundary points around each cluster. These "footprint" points are then used to match shape primitives to them as it can be seen in Figure 5.

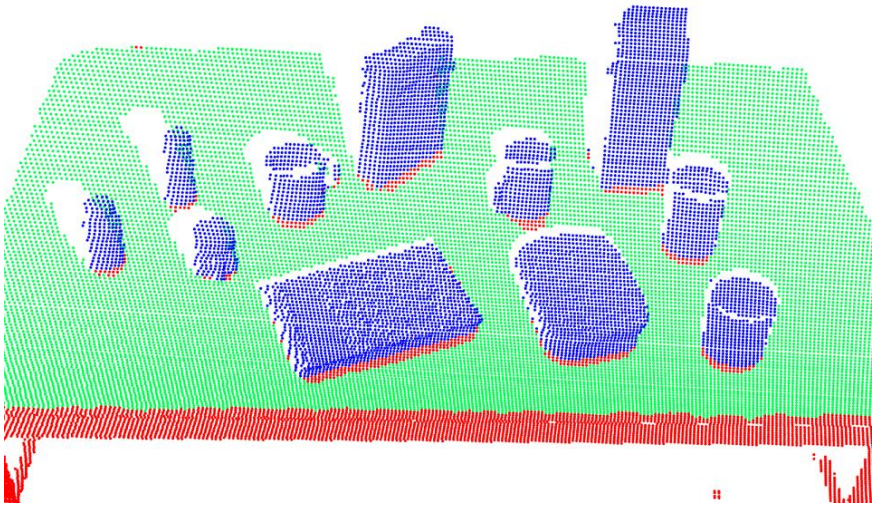

Fig. 4 Detected table and objects on top of it in a partial scan.

The search radius used for identifying the neighbors was chosen so that it compensates the variations caused by noise, but still includes points on thick handles such that a robust fit can be performed.

3.3 Hierarchical Model Fitting

Having the 2D boundary points for each cluster, we fit shape primitives to them, lines and circles more specifically, in order to locate the vertical planes and cylinders in the object.

Initially a line and a circle is fit to the footprint, and whichever has the most inliers gets accepted. Before accepting a circle however, two conditions have to be met.

Since a single circle can approximate a rectangle much easier than a single line, special care has to be taken to correctly recognize rectangles. For this, an oriented bounding rectangle is computed around each region using principle component analysis, and the average of normalized distances to the closest boundary points is computed as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\frac{dist(p_i, obr)}{width(p_i, obr)}, \tag{1}$$

where N is the number of boundary points p_i in a region, obr is the oriented bounding rectangle of the region, and the functions $dist()$ and $width()$ return the minimum distance of the point p_i to the sides of obr and the width of the bounding rectangle along the measurement direction respectively. Thus a fraction is computed between the distance of the point to a side of the obr along a principle component, and the width of the obr along that principle component.

Naturally, μ will have smaller values for rectangles than for circles or half-circles. The average normalized distance was found to be around 3% for rectangular footprints, and above 5% for non-rectangular ones, thus allowing us to decide when to neglect a first circular fit to the region (please see Figure 6).

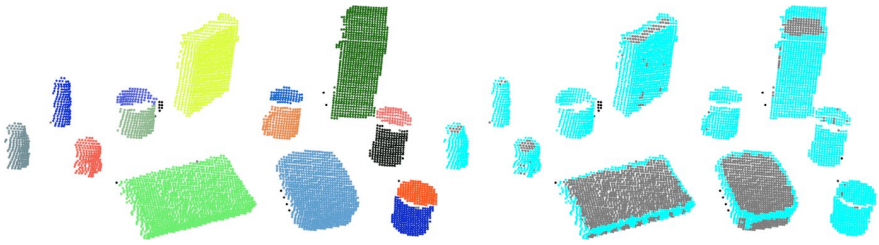

Fig. 5 Left: the identified objects on the table based on the clustering of their projections shown in random colors. Right: points that lie on the boundaries of the clusters are highlighted.

In some cases checking the oriented bounding box is not enough, so circles have to be checked also against a set of parallel and perpendicular lines fit to the data. To do this, lines are fit to the cluster until there are not enough points left, and a subset of these lines is selected that are pairwise parallel or perpendicular, and have the most number of inliers. This exhaustive search has a low computational complexity, since it only has to be performed for points in a single cluster.

Please note that the orange points in Figure 3b are inliers to a line that was grouped to the previous parallel line found in the cluster.

In the subsequent iterations, the steps are repeated (except the check with the oriented bounding rectangle, of course) for the remaining points, until their number drops below a minimum threshold for which a robust fit can not be ensured anymore. Please see the results in Figure 7.

3.4 Model Correction and Merging

After a set of shape primitives were fit to the region, these have to be corrected and possibly merged to improve the quality of the reconstruction.

Inliers of 2D circles which have a high overlap, are likely to belong to the same 3D cylinders, so merging them simplifies and also corrects the reconstruction. The criterion for merging circles, is that one of them includes the center of the other, since small measurement errors, or the presence of a handle for example can already modify the position and size of the best fit considerably.

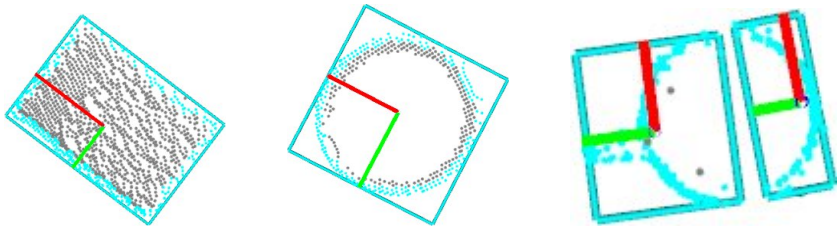

Fig. 6 Left to right: PCA analysis performed on clusters of a book, pan and the two parts of a mug.

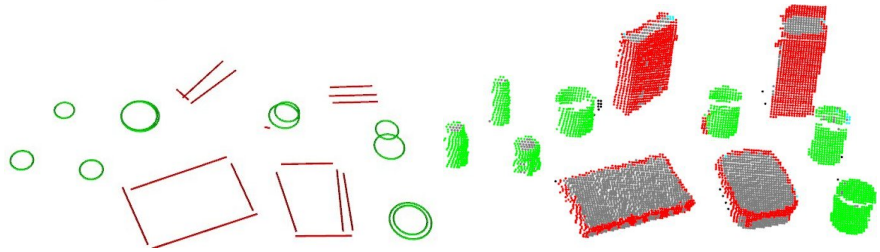

Fig. 7 Left: the 2D shapes that were fit to the cluster. Right: the inliers of those shapes in 3D.

When two circles from the same region are merged, a weighted average is computed for them, where the weights are the number of inliers of each circle. A re-fit using RANSAC would be redundant, because a search for circles was already performed in the region, and yielded the two circles separately. For circles from different regions, a complete re-fit is possible since their inliers were not considered already by the sample consensus method, so this gives accurate results even for small overlaps. An example for a merge between circles from different regions is illustrated in Figure 8. In these cases, the two regions are merged into one, since the two parts of a circle indicate a strong evidence that over-segmentation occurred.

Lines that form parallel and/or perpendicular groups in a region are also merged, since they are most probably part of a box for which the boundaries were identified (see Figure 9 for results).

3.5 Model Reconstruction and Verification

The corrected lines and circles are transformed into boxes and thick cylinders respectively, using the minimum and maximum heights of their inliers, together with the distances of the inliers to the model as thickness (as shown in the right part of Figure 9).

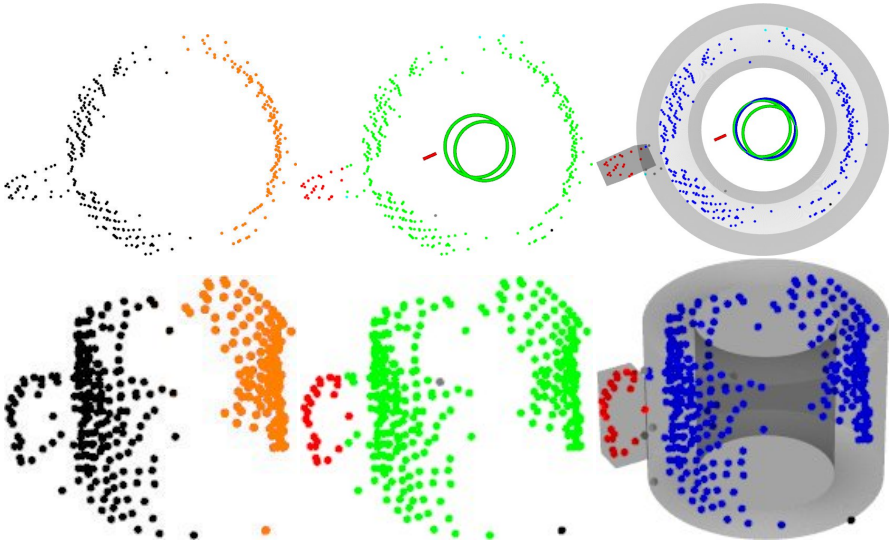

Fig. 8 From left to right (top row is viewed from above, while the bottom from the side): a) original measurement points returned from a mug, clustered in two disconnected regions; b) the inliers to the primitives identified in the regions (the shapes themselves can be seen in the background of the top row pictures) c) the two circles are merged and the model is re-fitted.

In order to verify these models, we generate points on a grid on the sides of the boxes and on the surface of the cylinders defined by the initial circles. These points can fall into 3 categories:

1. points that are *invisible* from the current viewpoint;
2. points that are *verified* by measurements;
3. points that are *void*, meaning that they are in visible free space, thus the model of the object should not include those parts of the shapes.

To check which points are verified, we verify if there are measurement points in their neighborhoods, within a maximum distance d_{th}. Those points that are not *verified* are checked for visibility by verifying the points that lie along the vector that connects the point to the viewpoint, or at most at distance d_{th} from it. If the point is occluded by measurement points, it is marked as *invisible* and as *void* otherwise.

An example can be seen in Figure 10. This way we can form an image about the measure of these misalignments. Generally we can say that the error of the assumption that the sides of the objects are perpendicular to the estimated normal of the table, lies well below 5 degrees, as does the error in approximating the rotation of the object around the normal.

The resulting point categories give valuable feedback about the probability of a successful fit for that particular shape, but also on how well the object respected our assumptions. For their interpretation please see Section 5.

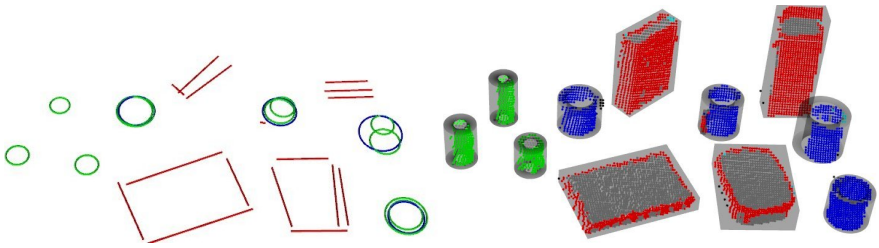

Fig. 9 Left: the corrected circles are marked with blue. Right: the reconstructed 3D models from the merged 2D shapes.

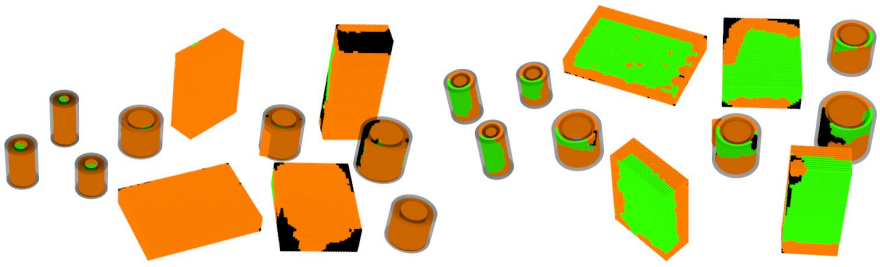

Fig. 10 Verification of object models, viewed from the front (left) and from the back (right). Orange points are marked as *verified*, green ones as *invisible* and black points as *void*.

4 Experimental Results

We applied our method to several views of tables containing objects of every day use on them (e.g., boxes, tetra packs, mugs, jars, small containers, plates and pans) at different distances and orientations, and obtained fairly robust results, as it can be seen in Figures 3, 9 and 11.

The small variations in the results for the same dataset, are due to the random element in the sample consensus approach, but the method gives consistent approximations. There is a small ambiguity for very thin, small containers, which are sometimes reconstructed as boxes instead of cylinders, but the small inaccuracies of the laser scanner make it very hard to distinguish circles with small radii from a box. Please see Figure 12 for an example.

In some cases an incorrect fit of a cylinder is accepted by the method over a part of a box,but they get rejected whenever they are verifiable, as presented in Figure 13.

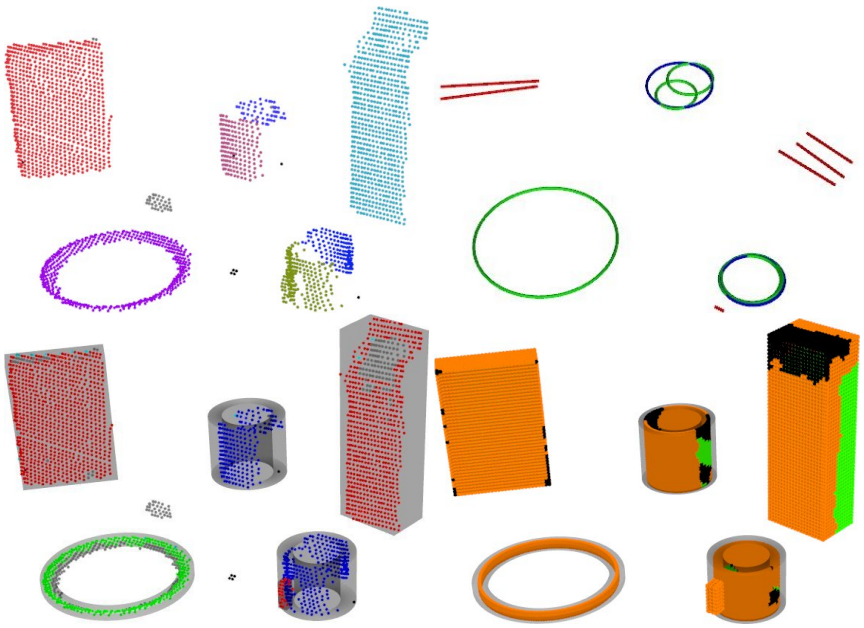

Fig. 11 From top to bottom, left to right: a) connected components, b) shape primitives and their corrections, c) the obtained boxes and thick cylinders, and d) the labels of the areas on the models. For the interpretation of the colors we kindly refer the reader to the explanations of the previous pictures.

5 Conclusions

In this paper, we presented a method for producing approximations of complete object models from a single partial view, by fitting planar patches (and combining them into boxes) together with cylindrical areas to 3D point cloud data. The processing steps include an analysis of the silhouettes of the object's 2D projections, and their decomposition into shape primitives. These are then used to recompute the parameters of 3D shapes that model the real objects well enough for estimiating grasping points. The approach can easily be extended to model different layers of objects separately for increased accuracy.

A method for merging parts of over-segmented objects is inherently embedded in our approach, as is the verification and correction of the models. The labels set by the model verification step provide means of checking the correctness of the model at each surface unit, refit the model if the current one is implausible (see Figure 13), and remove parts of the model if necessary. This way concavities can be recognized and since the model extends to the limits of the inliers, unexpected collisions can be avoided.

While the exact grasping strategy is not the scope of this paper (please see the approaches mentioned earlier for example), this information can also be used to optimize grasping (since poses for grabbing boxes and cylinders are relatively easy to generate) and improve its accuracy. In the case of pre-generated grasps for boxes and cylinders, apart from the collision checks with the other parts of the object and the surroundings, this can be achieved by excluding the generated end-effector

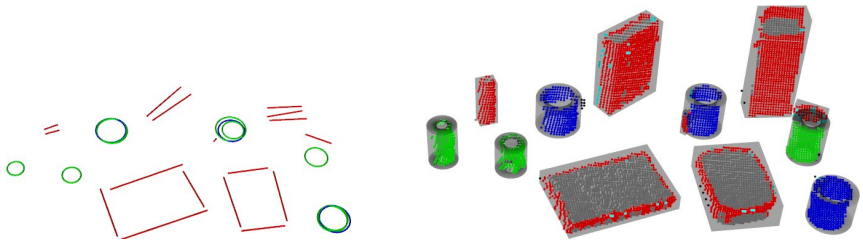

Fig. 12 A slightly different reconstruction of the scene presented previously in Figure 9. Left: the 2D shapes that were fit to the cluster. Right: the inliers of those shapes in 3D.

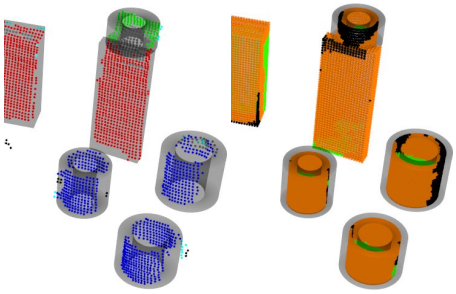

Fig. 13 Incorrect fit of the cylinder on top of the box, detected on the basis of the high amount of *void* parts.

poses that would require it to have contact in the *void* points for a successful grip. If grasps are to be generated after the completed model is obtained, good grasps can be obtained by adjusting the friction coefficients on the model based on these labels.

The points remaining after fitting the shape models might hold some information, so grouping them and fitting them into boxes might be useful, but this will have to be limited in order to avoid the unnecessary complication of the object models because of measurement noise. The points which can not be explained by the models might be introduced as triangular meshes to form hybrid object models, but the primary problem here is the resolution and accuracy of the measurements, which still has room for improvement, for example by accepting multiple return pulses instead of averaging them in the case when the laser beam hits an edge and the object behind it as well.

Our next steps will be to work on the problem of separating objects which are located close to each other, and to extract features from the fitted model for using them in training a a classifier that differentiates between plausible and less plausible configurations, i.e. to find out how probable is that a combinations of fitted models to a cluster is approximating the true 3D structure correctly (employing similar techniques as in [10]). For this, a large number of hand-labeled training sets of correct and incorrect fits is needed, for which the slight variations for the same data introduced by the random sample consensus method works to our advantage.

Acknowledgements This work is supported by the CoTeSys (Cognition for Technical Systems) cluster of excellence.

References

1. Biegelbauer, G., Vincze, M.: Efficient 3D Object Detection by Fitting Superquadrics to Range Image Data for Robot's Object Manipulation. In: IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy (2007)
2. Bone, G., Lambert, A., Edwards, M.: Automated Modeling and Robotic Grasping of Unknown Three-Dimensional Objects. In: Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, USA (2008)
3. Collet, A., Berenson, D., Srinivasa, S.S., Ferguson, D.: Object Recognition and Full Pose Registration from a Single Image for Robotic Manipulation. In: IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan (2009)
4. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: Comm. of the ACM, Vol 24 (1981)
5. Harada, K., Kaneko, K., Kanehiro, F.: Fast grasp planning for hand/arm systems based on convex model. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, pp. 1162–1168 (2009)
6. Klank, U., Zia, M.Z., Beetz, M.: 3D Model Selection from an Internet Database for Robotic Vision. In: International Conference on Robotics and Automation (ICRA), Kobe, Japan (2009)
7. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(9), 1465–1479 (2006)

8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004). DOI http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94
9. Marton, Z.C., Rusu, R.B., Beetz, M.: On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan (2009)
10. Marton, Z.C., Rusu, R.B., Jain, D., Klank, U., Beetz, M.: Probabilistic Categorization of Kitchen Objects in Table Settings with a Composite Sensor. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). St. Louis, MO, USA (2009)
11. Miller, A., Allen, P.K.: Graspit!: A Versatile Simulator for Robotic Grasping. IEEE Robotics and Automation Magazine **11**(4), 110–122 (2004)
12. Rabbani, T., Heuvel, F.: Efficient hough transform for automatic detection of cylinder in point clouds. In: ISPRS WG III/3, III/4, V/3 Workshop, Laser scanning 2005, Enschede, The Netherlands (2005)
13. Richtsfeld, M., Vincze, M.: Grasping of Unknown Objects from a Table Top. In: Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments (2008)
14. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3D Point Cloud Based Object Maps for Household Environments. Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge) (2008)
15. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M.E., Beetz, M.: Functional Object Mapping of Kitchen Environments. In: Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, September 22-26 (2008)
16. Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic Grasping of Novel Objects using Vision. The International Journal of Robotics Research **27**(2), 157–173 (2008)
17. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for Point-Cloud Shape Detection. Computer Graphics Forum **26**(2), 214–226 (2007)
18. Thrun, S., Wegbreit, B.: Shape from symmetry. In: Proceedings of the International Conference on Computer Vision (ICCV). IEEE, Bejing, China (2005)
19. Ulrich, M., Wiedemann, C., Steger, C.: Cad-based recognition of 3d objects in monocular images. In: International Conference on Robotics and Automation, pp. 1191–1198 (2009)
20. Vosselman, G., Dijkman, S.: 3d building model reconstruction from point clouds and ground plans. In: International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4 pages 37-43,Annapolis, MD, 22-24 Oct. 2001 (2005)
21. Zhang, Y., Koschan, A., Abidi, M.: Superquadric Representation of Automotive Parts Applying Part Decomposition. Journal of Electronic Imaging, Special Issue on Quality Control by Artificial Vision, Vol. 13, No. 3 pp. 411–417 (2004)