

## COMBINING CONTOUR AND SHAPE PRIMITIVES FOR OBJECT DETECTION AND POSE ESTIMATION OF PREFABRICATED PARTS

Alexander Berner, Jun Li, Dirk Holz, Jörg Stückler, Sven Behnke, Reinhard Klein

Computer Science Institute, University of Bonn, Germany

### ABSTRACT

Man-made objects such as mechanical construction parts can typically be described as a composition of shape primitives like cylinders, planes, cones and spheres. We propose a robust method for the detection and pose estimation of such objects in 3D point clouds. Our main contribution is to enhance a probabilistic graph-matching approach that detects objects using 3D shape primitives with distinct 2D primitives such as circular contours. With this extension, our method copes with difficult occlusion situations and can be applied for object manipulation in complex scenarios such as grasping from a pile or bin-picking. We demonstrate the performance of our approach in a comparison with a state-of-the-art feature-based method for objects of generic shape and a primitive-based approach using only 3D shapes and no contours.

**Index Terms**— shape primitives, contour primitives, object detection, pose estimation

### 1. INTRODUCTION

While much research investigates the recognition of generic and organic objects in intensity images and 3D point clouds, such methods do not exploit properties of man-made objects that are often encountered in industrial and service robotics applications. Typically, man-made machine parts are composed of shape primitives such as cylinders, spheres, cones, and planes (see, for example, the objects in Fig. 1).

In this approach, we follow a popular line of work matching graph models [1] for object recognition. Both the object being searched for and the scanned scene are represented as compositions of geometric primitives in graphs. Object hypotheses are generated by identifying parts of a search graph in the graph of a captured scene. Using the established graph correspondences, one is able to determine the pose of the object in the scene. Finally, hypotheses are verified with the original 3D model in the scene point cloud.

Since the detection of 3D shape primitives requires areal measurements on the primitive's surfaces, occlusions by other objects and the object itself may hinder reliable detection. Only detecting a part of the primitives forming an object can



**Fig. 1.** Left: Robot grasping a found pipe object out of a box. Right: we detect 2D contour and 3D shape primitives in 3D point clouds and perform graph-matching to detect objects and to estimate their pose.

lead to ambiguities in the object's pose or not detecting the object at all. For many object types, however, contours at occlusion boundaries and at sharp object edges are unique features that remove ambiguities and support pose estimation. In this paper, we propose to complement 3D shape primitives with 2D contour primitives such as circles. The novel combination of 2D contour and 3D primitive features tremendously improves detection rate in difficult occlusion settings.

The remainder of this paper is structured as follows: after a discussion of related work in section 2, we describe our approach to detecting occlusion boundaries and contour primitives as well as the extension of our graph based object detection framework in section 3. In section 4 we present experiments where we compare our approach to a state-of-the-art object detection method [2] and our former approach only using shape primitives (as used in [3]).

### 2. RELATED WORK

A common approach for object detection and pose estimation is to use intensity images and various invariant features to find correspondences between a query object and the scene. Construction parts as used here are usually textureless making visual features such as SIFT [4] and SURF [5] less suitable. In 3D data, various invariant features have been proposed that

This research has been partially funded by the FP7 ICT-2007.2.2 project ECHORD (grant agreement 231143) experiment ActReMa.

catch the distribution and arrangement of points and local surface normals to describe an object or a part of it. Prominent examples are spherical harmonic invariants [6], spin images [7], curvature maps [8], and—more recently—point feature histograms (PFH) [9, 10], FPFH [11], VFH [12], SHOT [13]. Point-pair feature (PPF) voting methods find locally consistent arrangements of surfel pairs between model and scene through either Hough voting [14] or RANSAC [2]. However, these approaches are designed for organic shapes that provide high variability in shape throughout the object. Man-made objects often consist of few simple geometric primitives, i.e., less unique local regions. Another line of work finds compositions of edges in depth images [15]. These approaches are particularly well suited for objects in piles, but may require special sensing to reliably find edges [16]. In the presence of occlusions erroneous edge detections can hinder reliable detection.

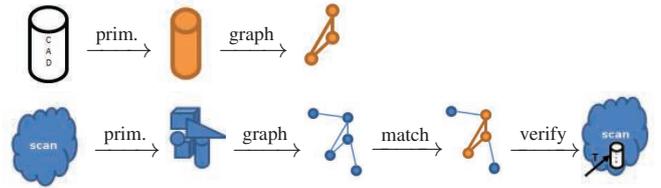
We address the problem of object detection based on detecting geometric shape and contour primitives. A widely used and adopted method for detecting basic shapes in unorganized point clouds is by Schnabel et al. [17] (section 3.1). It decomposes the point cloud into inherent shapes such as planes, spheres, cylinders, cones, and tori. In [18] the work is extended by modeling compositions of primitives in a graph and finding certain arrangements in building models. Previous work [3] built on top of this approach to detect objects using graphs of connected primitives. In this paper we evaluate this method on various real world scans and show how results can be drastically improved by our combination with 2D contour primitives.

Related to this approach is the work of Li et al. [19] who build a similar graph and use it to iteratively refine orientations and positions of detected shapes. Also related is fitting superquadrics to represent individual parts of objects [20]. Superquadrics generalize better on object parts that cannot be described by a single simple shape primitive and can adopt to a larger class of shapes. The technical construction parts we’re aiming to detect reliably are, however, composed of simple shapes only. Hence, restricting the representations of parts in the object part graphs to simple shape primitives not only suits perfectly the class of objects of interest, but also drastically limits the dimensionality of the parameter space. Detecting individual parts is thus more stable and reliable.

### 3. APPROACH

In our approach, objects are represented by compositions of 2D contour and 3D shape primitives. We model such compositions by a graph of spatial relations between primitives. Detected primitives, and the model parameters describing them, form the vertices of the graph. Spatial relations between (neighboring) primitives are encoded in the edges.

Given the graph of a query object, we convert an input point cloud into a graph modeling the captured scene and find



**Fig. 2.** Object learning (top) and detection (bottom): In CAD models or 3D point clouds, we detect contour and shape primitives and construct a graph encoding their composition. Object hypotheses are obtained from constrained sub-graph matching and verified by checking the overlap between the transformed model and the points in the original point cloud.

sub-graphs that match with the graph of the query object (see Fig. 2).

#### 3.1. Shape primitive detection and graph construction

We assume man-made objects that are composed of simple geometric shape primitives such as planes, cylinders, spheres, cones and tori. In order to detect such primitives we employ the algorithm by Schnabel et al. [17] based on random sampling. It decomposes a point cloud  $P = p_1, \dots, p_N$  into a set of inherent geometric shape primitives  $\phi_i$  with support points  $S_{\phi_i}$  and a set of remaining points  $R$ :

$$P = S_{\phi_1} \cup \dots \cup S_{\phi_A} \cup R. \quad (1)$$

Each support set  $S_{\phi_i}$  is a connected component of points that are 1) close to the primitive (distance  $< \epsilon$ ) and 2) compatible w.r.t. the angle (angle  $< \alpha$ ) between surface normals at the point  $n_s$  and on the primitive at the closest point on the primitive  $n(\phi_i, s)$ :

$$s \in S_{\phi_i} \Rightarrow \|s, \phi_i\| < \epsilon \wedge \angle(n_s, n(\phi_i, s)) < \alpha. \quad (2)$$

As an important extension to [17], we distinguish two different phases of detecting (shape) primitives: off-line learning of new objects where we aim at finding an optimal decomposition into arbitrary shape primitives (open model parameters) and on-line detection of known objects (see Fig. 2). We save computations in the on-line phase by constraining the primitive detection to only find primitives existing in the query object (fixed model parameters). In case of (off-line) learning of new objects given CAD models, we first sample the model uniformly to obtain a 3D point cloud.

For each detected primitive  $\phi_i$  a vertex is added to the topology graph  $G(\Phi, E)$ , i.e.,  $\Phi = \phi_1, \dots, \phi_a$ . An edge  $e = (\phi_i, \phi_j)$  is added if the support sets of the primitives  $\phi_i$  and  $\phi_j$  are neighboring (as opposed to the distance between the actual primitives having indefinite extent, i.e.:

$$\exists p \in S_{\phi_i}, q \in S_{\phi_j} : \|p - q\| < t \quad (3)$$

Three types of constraints are encoded in the graph: *node constraints* for the similarity of primitives (model parameters such as type and size), *edge constraints* for the similarity of spatial relations between incident primitives (e. g., the angle between two planes), and *graph constraints* given only implicitly by the topology in the graph (e. g., parallelism of disconnected planes).

### 3.2. Boundary estimation and contour detection

For being able to detect contour primitives we first extract all points in an input point cloud belonging to sharp edges and occlusion boundaries. We employ the algorithm by Bendels et al. [21] that computes, for every point, a contour probability using a local  $k$ -neighborhood around the query point and various criteria combined in weighted sum: 1) the angle between the query point’s normal and the normals of the neighbors, 2) the relative positions of the neighbors to the query points and 3) the shape of the underlying surface at the query point (encoded in the Eigenvalues of the covariance matrix).

Given the set of points lying on sharp edges and occlusion boundaries in a point cloud, we aim at finding contour primitives that help in detecting objects and resolving ambiguities. To this end, we only consider circular contours since our objects of interest either contain cylinders or drill holes that result in circular contours. Moreover, circles have only one shape parameter, besides the position and orientation, for the radius. Therefore, we are able to detect them very robustly in noisy and occluded data. As for the shape primitive detection, we use a RANSAC-based approach and distinguish between on-line and off-line phase. In the off-line phase, we fit circle hypotheses by sampling three points, estimating the common plane, and determining the center of the circle through the points. In the on-line phase, the points are chosen according to the searched radii computed of the queried object and efficiently tested using a fast octree implementation that is also used in the shape primitive detection. We only accept hypotheses with a sufficient number of supporting inliers.

Detected contour primitives are added to the graph  $G(\Phi, E)$  already containing detected shape primitives. False positives in the contour detection arising in the on-line phase will be pruned in our graph-matching approach, since their relative pose is inconsistent with the object model graph.

### 3.3. Graph matching and pose estimation

Since we cannot expect to always find the full query graph of the searched object, we are looking for a maximal partial match. The matching procedure takes advantage of the annotations at the nodes and edges given by translation- and rotation-invariant shape properties like radii or relative poses. We employ the recursive constrained sub-graph matching algorithm by Schnabel et al. [18].

**Graph matching:** Input to our method is the graph of 2D/3D primitives of a query object to be searched for in the



**Fig. 3.** Objects, features and primitives. Left to right: wood, cross clamping piece, and pipe. Top to bottom: typical point pair features (PPF), contour primitives, and shape primitives.

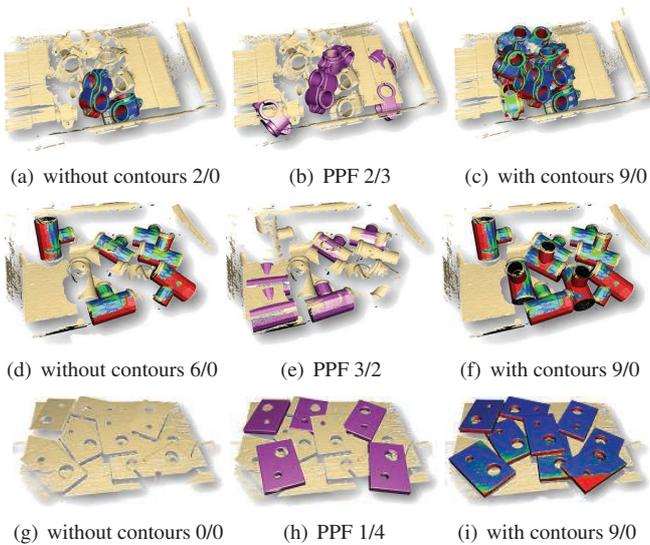
scene. We start with a random edge in the query graph and find similar edges in the scene graph. We compare edges by their relative pose and nodes by their shape properties, and compute a score for each match. This score is zero if the types of primitives do not match and increases with the similarity in relative pose and shape parameters. For each matching edge, we expand the match to adjacent edges (and nodes) in query and scene graph, if they also match in relative pose and shape properties. The search is not exhaustive since expansion is stopped if either the whole query graph matches or no further corresponding edges can be found in the scene graph. We take into account deviations in the object parts by allowing tolerances specified by hand. The process is repeated in order to find multiple objects in the scene.

**Pose estimation:** We determine object poses from partial matches between model and scene graph. Depending on the type of shape primitive, each correspondence determines some of the six degrees of freedom of the pose. A circle-to-circle correspondence, for example, completely determines the translation between the circle centers and two rotational degrees of freedom by the alignment of the circle planes. It does not, however, determine the rotation around the axis perpendicular to the circle plane through its center. Hence, we require several correspondences between shapes until the pose of the object is fully retrieved. Computed object poses are refined by registration using the ICP variant of Mitra et al. [22].

**Hypotheses verification:** Since the previous steps only consider the matching between shapes but not their consistency with the overall scan false-positives may be generated. In a verification step, we check the overlap of object hypotheses with the actual point cloud and sort out those with insufficient overlap (15% in our experiments). If several hypotheses overlap with the same scan points, we remove the hypotheses with lower overlap and only keep the best.

## 4. EXPERIMENTS

The scenario for our tests is a construction task, where a mobile robot picks prefabricated parts out of bins (see Figure 1). Each bin contains an unorganized pile of objects of the same type. We have chosen three types of objects for our experi-



**Fig. 4.** Example detections (true positives / false positives) for cross clamping pieces (10 visible, 9 pose estimable), pipe objects (10 visible, 9 pose estimable), and wood objects (9 visible, 9 pose estimable).

ments (see Figure 3) that are similar to parts found in real-world construction applications—a rectangular wooden plate containing two drilled holes (manually created CAD model), a cross clamping piece (CCP), a typical construction part to connect poles (CAD models are freely available), and a drain pipe connector (scanned from different perspectives and registered to obtain a 3D model). For every object we have first computed a query graph using the off-line variant of our approach and then detected objects in, respectively, scans and scene graphs in five differently piled heaps of objects.

We compare the proposed approach combining contour and shape primitives with our previous approach using shape primitives only [3], and an implementation of the state-of-the-art approach by Papazov and Burschka [2] based on point pair features (PPF) with parameters as recommended by the authors. In Fig. 4 we show one scan per object with the detection results of the evaluated approaches for qualitative analysis and visual inspection. Quantitative results are summarized in Table 1. Since the PPF-based approach is randomized, it is run for ten times on every input scan to determine the average detection rate. For the class of objects in our problem setting our method clearly outperforms the other two methods.

Compared to the shape-only approach (as in used [3]), adding contour primitives improves detection results by resolving object pose ambiguities. In case of the CCP, we often find only a single cylinder that does not suffice for determining the object pose as it leaves two degrees of freedom undetermined. Adding circular contour primitives yields unique pose estimate and successful detections. The worst case for the shape-only approach is shown in Fig. 4, where the plates are lying nearly flat on the table. The objects cannot be de-

**Table 1.** Average accuracy of the detection

Average accuracy	Ours	Without 2D	PPF
CCP	0.81	0.23	0.29
Pipe	0.89	0.47	0.27
Wood	0.82	0.21	0.08
Overall average	0.84	0.30	0.22

true positives / (true positives + false negatives + false positives)

tected as the only found primitives are (indefinitely extending) planes (with three open degrees of freedom). By combining the planes with the contour primitives, all objects with estimable pose can be detected.

Compared to the PPF-based approach (showing outstanding performance for objects of generic shape) we also achieve better detection rates. Man-made objects as addressed in our work are usually formed by compositions of few simple geometric primitives. These are easy to detect for our approach but also have the property of less varying normals which is disadvantageous for approaches based on geometric features. PPFs computed in these areas do not provide unique transformations and can cause false positives. Our primitive-based approach does not generate hypotheses in these areas, because it requires only a minimal number of shape matches. Hence, it does not produce any false positives in our experiments. A restriction of our solution is that it is only suitable for objects that can be described by a composition of shape primitives. It cannot be applied to arbitrary organic objects.

## 5. CONCLUSIONS

In this paper, we introduced the combination of 2D contour primitives and 3D shape primitives for detecting objects in point clouds through shape-graph matching. We provide qualitative and quantitative experiments on the object types we focused on and compare the results to a state of the art method of point pair feature matching. Our combination of primitive types tremendously improves the robustness of the simple shape primitive object detection approach, compared to using 3D shape primitives alone. Our method is ideally suited for man-made construction objects that can be described by compositions of shape primitives. In experiments, our method shows very good detection results and clearly outperforms the former method and a recent general approach to object detection based on point pair features when evaluated on mechanical construction parts.

In future work, we will incorporate further 3D surface and 2D contour primitives such as spline-based contour descriptions into our method. In order to further improve run-time we will consider parallel implementation on GPUs. In the current implementation, model tolerances are specified beforehand. It is a matter of future work to learn these automatically together with the object model.

## 6. REFERENCES

- [1] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, Jan. 2005.
- [2] Chavdar Papazov and Darius Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2011, pp. 135–148.
- [3] Matthias Nieuwenhuisen, Jörg Stückler, Alexander Berner, Reinhard Klein, and Sven Behnke, "Shape-primitive based object recognition and grasping," in *Proc. of the German Conf. on Robotics*, 2012.
- [4] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.
- [6] G. Burel and H. Henocq, "Three-dimensional invariants and their application to object recognition," *Signal Processing*, vol. 45, no. 1, pp. 1–22, 1995.
- [7] A.E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.
- [8] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature maps for local shape comparison," in *Proc. of the Int. Conf. on Shape Modeling and Applications (SMI)*, 2005, pp. 244–253.
- [9] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification," in *Proc. of the Int. Conf. on 3-D Digital Imaging and Modeling (3DIM)*, 2003, pp. 474–481.
- [10] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz, "Learning informative point classes for the acquisition of object model maps," in *Proc. of the Int. Conf. Control, Automation, Robotics and Vision (ICARCV)*, 2008, pp. 643–650.
- [11] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz, "Fast point feature histograms (FPFH) for 3d registration," in *Proc. of the IEEE international Conf. on Robotics and Automation (ICRA)*, 2009, pp. 1848–1853.
- [12] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 2155–2162.
- [13] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2010, pp. 356–369.
- [14] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 998–1005.
- [15] Darius M. Gavrilă, "A Bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1408–1421, 2007.
- [16] Amit Agrawal, Yu Sun, John Barnwell, and Ramesh Raskar, "Vision-guided robot system for picking objects by casting shadows," *Int. J. Rob. Res.*, vol. 29, no. 2-3, pp. 155–173, 2010.
- [17] Ruwen Schnabel, Roland Wahl, and Reinhard Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [18] R. Schnabel, R. Wessel, R. Wahl, and R. Klein, "Shape recognition in 3D point-clouds," in *Proc. of the Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2008.
- [19] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra, "GlobFit: consistently fitting primitives by discovering global relations," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, vol. 30, pp. 52:1–52:12, 2011.
- [20] Georg Biegelbauer and Markus Vincze, "Efficient 3D object detection by fitting superquadrics to range image data for robot's object manipulation," in *Proc. of the IEEE international Conf. on Robotics and Automation (ICRA)*, 2007, pp. 1086–1091.
- [21] Gerhard H. Bendels, Ruwen Schnabel, and Reinhard Klein, "Detecting holes in point set surfaces," *Journal of WSCG*, vol. 14, no. 1–3, Feb. 2006.
- [22] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proc. of the Eurographics Symposium on Geometry Processing (SGP)*, 2004, pp. 23–31.