

# Object Recognition using constraints from Primitive Shape Matching

Nikhil Somani<sup>‡</sup>, Caixia Cai<sup>†</sup>, Alexander Perzylo<sup>‡</sup>, Markus Rickert<sup>‡</sup>, and Alois Knoll<sup>†</sup> <sup>\*</sup>

Authors Affiliation: <sup>‡</sup>Cyber-Physical Systems, fortiss - An-Institut der Technischen Universität München. <sup>†</sup>Technische Universität München, Fakultät für Informatik.  
Address: <sup>‡</sup>Guerickestr. 25 80805 München (Germany). <sup>†</sup>Boltzmannstrae 3, 85748 Garching bei München (Germany)  
Email: <sup>‡</sup>{somani,perzylo,rickert}@fortiss.org, <sup>†</sup>{caica,knoll}@in.tum.de

**Abstract.** In this paper, an object recognition and pose estimation approach based on constraints from primitive shape matching is presented. Additionally, an approach for primitive shape detection from point clouds using an energy minimization formulation is presented. Each primitive shape in an object adds geometric constraints on the object's pose. An algorithm is proposed to find minimal sets of primitive shapes which are sufficient to determine the complete 3D position and orientation of a rigid object. The pose is estimated using a linear least squares solver over the combination of constraints enforced by the primitive shapes. Experiments illustrating the primitive shape decomposition of object models, detection of these minimal sets, feature vector calculation for sets of shapes and object pose estimation have been presented on simulated and real data.

## 1 Introduction

Object detection, recognition and pose estimation using 3D data is a classic problem in computer vision. There are several types of approaches widely used for this problem. Purely shape based approaches include geometric shape detection [1], primitive shape graphs [2], [3], surflet-pair based approaches [2], [4], [5] and triple-point feature method [6]. Keypoint and descriptor based approaches include local color keypoint [7], [8], local shape keypoint [9] and global descriptors [10]. Each of these approaches have their own advantages and disadvantages. Color based methods work only on textured objects, while purely shape based approaches can not distinguish between objects having identical shape but different texture. Global descriptors such as VFH [10] require a cumbersome training

---

<sup>\*</sup> The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n 287787 in the project SMERobotics, the European Robotics Initiative for Strengthening the Competitiveness of SMEs in Manufacturing by integrating aspects of cognitive systems.

phase, where a large number of object views need to be generated by real experiments. Besides, their accuracy decreases significantly in case of occlusions and partial views. The advantage of these methods lies in their computational speed. On the other hand, methods such as [3], [4], [5] are designed to be robust to partial views, occlusions and noisy data but don't scale well for real-time applications or large point clouds.

This work focuses on object detection using shape information, since the applications we deal with involve industrial parts which are often metallic and texture-less. Another important property of these parts is that they are often composed of simpler geometric parts and hence, their geometries can be approximated accurately using a set of primitive shapes such as planes, cylinders, sphere, etc. Primitive shape detection from 3D data, which is essential for this approach, is also a well-researched topic. Some of the prominent approaches include detection of planes and 3D conics [3], superquadrics [11] and fast plane detection [12]. In this paper, we propose an energy minimization based approach which can handle different kinds of primitive shapes, given that the distance of a candidate point from the shape and an estimate of the complexity of the shape representation can be calculated. Our approach is similar to the one presented in [2]. A key difference in our work is that the iterations for merging/filtering of primitive shapes are now included in the energy minimization approach itself. Hence, heuristics for merging shapes are no longer required.

An important concept presented in this work is an algorithm to detect minimal sets of primitive shapes in an object model. In [2], the authors presented the concept of minimal sets for 3D pose estimation. However, possible combinations such as 3 planes or a plane and a cylinder were defined explicitly. In our approach, we model the constraints that each primitive shape enforces on the object's pose and can detect these minimal sets automatically from a set of shapes. Also, the object recognition approach can handle over-specified constraints and estimate the pose in a least-squares sense. This provides additional tolerance to noise in the estimation of the primitive shape parameters.

Finally, an approach for calculating feature vectors from sets of primitive shapes is presented, that can be used for object recognition. These features contain not only the features from each of the primitive shapes, but also encode geometric properties which arise from combinations of these primitive shapes.

## 2 Organization/Overview

In Section 3, the primitive shape decomposition approach is presented. In Section 4, the modeling of constraints from primitive shapes is presented. In Section 4.1, an algorithm for object recognition using feature vectors from sets of primitive shapes is presented. In Section 4.2, an algorithm for detection of minimal sets of primitive shapes is presented. In Section 4.3, a pose estimation approach using a least squares optimization over shape constraints is presented. Finally, in Section 5 an evaluation of the mentioned approaches and some applications are provided.

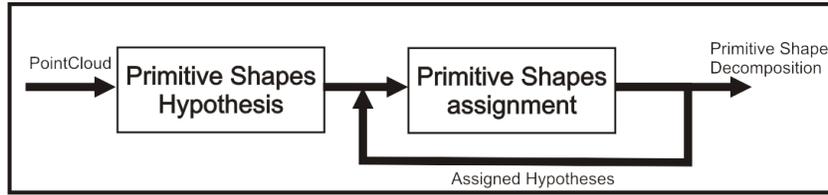


Fig. 1. Pipeline for Primitive Shape Decomposition.

### 3 Primitive Shape Detection

The primitive shape detection pipeline is shown in Fig. 1. The point cloud  $P$  is represented as a set of primitive shapes  $s_i$  containing points  $p_i \subseteq P$  such that  $\cup p_i \subseteq P$ . In this work, the primitive shapes  $s_i$  can be planes, cylinders and spheres. An example of such a decomposition is shown in Fig. 2, where the original scene cloud is shown in Fig. 2 (a) and its decomposition into primitive shapes is shown in Fig. 2 (b).

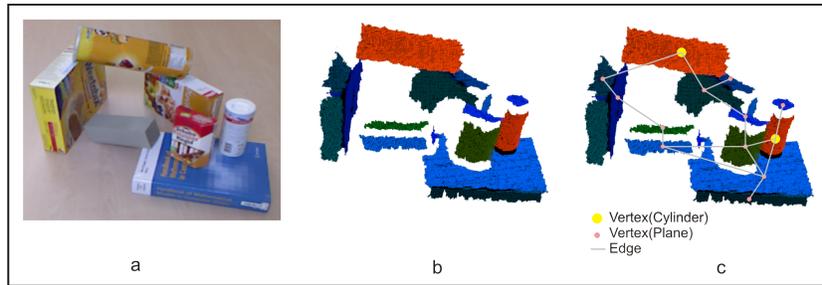


Fig. 2. Primitive Shape Decomposition example : (a) RGB channel of original Point Cloud (b) result of Primitive Shape Decomposition (c) Primitive Shape Graph representation.

#### 3.1 Primitive Shape Hypothesis

Hypothesis for primitive shapes are generated by randomly sampling point sets in the point cloud. Once the hypotheses have been generated, each point in the cloud is checked to determine whether it satisfies the hypotheses. The method used for generating a hypothesis and determining its inliers depends on the type of primitive shape.

- **Planes:** A plane hypothesis can be generated using a single point  $(X_0)$  with its normal direction  $(\hat{n})$ . To test if a point  $X$  lies on the plane  $(X - X_0) \cdot \hat{n} = 0$ , the distance of the point from the plane  $|(X - X_0) \cdot \hat{n}|$  is used.

- **Cylinders:** A cylinder hypothesis can be generated using 2 points  $(X_0, X_1)$  with their normal directions  $(\hat{n}_0, \hat{n}_1)$ . The principal axis of the cylinder is selected as the minimum distance line between the normal directions  $\hat{n}_0$  and  $\hat{n}_1$ . The radius  $r$  is the distance of either point to this line. To test if a point  $X$  with normal direction  $n$  lies on the cylinder, the minimum distance point on cylinder’s axis is calculated  $X_{min}$ . The vector  $X - X_{min}$  should have length  $r$  and direction  $n$ .
- **Spheres:** A sphere hypothesis can be generated using 2 points  $(X_0, X_1)$  with their normal directions  $(\hat{n}_0, \hat{n}_1)$ . The intersection of the lines along the normal directions of each point is the center of the sphere  $X_c$ . The radius  $r$  is the distance of either point to the center. To test if a point  $X$  with normal direction  $n$  lies on the sphere, vector having length  $r$  and direction  $X - X_c$  is compared with the vector having length  $|X - X_c|$  and direction  $n$ .

### 3.2 Primitive Shape Assignment

The hypotheses associated with each point in the cloud can be considered as labels for point. There may be multiple labels associated with each point and the labeling may be spatially incoherent. To resolve such issues and generate a smooth labeling, a multi-label optimization using graph-cuts is performed. In this setting, the nodes in the graph comprise all possible assignment of labels to the points. The different terms of the energy functional are explained below:

- The data term indicating the cost of a label assignment to a point is proportional to the distance of the point from the primitive shape.
- The smoothness term penalizes neighboring points having different labels and the penalty is inversely proportional to the distance between the neighboring points.
- Label swap energies are used for neighboring primitive shapes in a way that only neighboring primitive shapes labels can be swapped.
- Label energies are added according to the complexity of the primitive shape (e.g. planes require 3 parameters while cylinders require 7), thereby favoring simpler primitives and penalizes over-fitting of primitives.

The convex energy functional thus created is then solved using the  $\alpha$  - expansion,  $\beta$  -swap algorithms [13], [14], [15], [16] which gives the label assignment for each point in the cloud, such that the total energy is minimized.

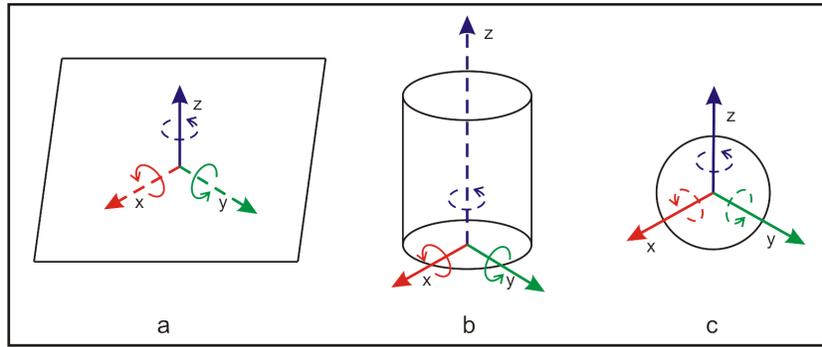
### 3.3 Merging and Pruning of Primitive Shapes

Due to the addition of label energies, some hypotheses might not be assigned any points. Such hypotheses are removed from the hypothesis set and the optimization process is repeated until no more hypotheses can be removed.

### 3.4 Primitive Shape Graph(PSG) Representation

The primitive shapes detected in the previous step are now used to create a graphical representation of the point cloud. In this graph  $G = (V, E)$ , each primitive shape is a node  $v \in V$  and neighboring primitive shapes are connected by an edge  $e \in E$ . An example of such a graph is shown in Fig. 2 (c).

## 4 Constraints from Primitive Shapes



**Fig. 3.** Geometric Constraints enforced by each primitive shape. Constrained/un-constrained axes are indicated by solid/dotted arrows respectively. (a) Infinite Planes: The position along z-axis (normal direction) is fixed, while positions along x and y is free. Rotation is allowed only around z-axis. (b) Infinite Cylinder: The position along x and y axes is fixed, while position along z-axis (principal axis of cylinder) is not fixed. Rotation is allowed only around the z-axis. (c) Sphere: The positions in all axes are fixed. Rotation is allowed around all axes.

Each primitive shape  $P_i$  enforces a set of constraints  $(C_{pi}, C_{ni})$  on the position and orientation of the object respectively. Each row of  $C_{pi}$  and  $C_{ni}$  contains a direction along which the constraint has been set. Examples of constraints set by each primitive shape are shown in Fig. 3 and explained below:

- Infinite plane:  $C_{pi} = [0, 0, 1]$  and  $C_{ni} = [1, 0, 0; 0, 1, 0]$
- Infinite cylinder:  $C_{pi} = [1, 0, 0; 0, 1, 0]$  and  $C_{ni} = [1, 0, 0; 0, 1, 0]$
- Sphere:  $C_{pi} = [1, 0, 0; 0, 1, 0; 0, 0, 1]$  and  $C_{ni} = \phi$

A *complete set* of primitive shapes is defined as a set where the constraints fully specify the 3D position and orientation of the object. A *minimal set* of primitive shapes is defined as a set which is complete but removing any primitive shape from the set would make it not complete.

**Table 1.** Feature Vectors for Primitive Shape sets

Primitive Shape	FeatureVector ( <i>fv</i> )
Inf. Plane	$\phi$
Inf. Cylinder	<i>radius</i>
Sphere	<i>radius</i>
Plane+Plane	$fv(plane1), fv(plane2),$ $angle(plane1\_normal, plane2\_normal),$ $min\_distance(plane1, plane2)$
Cylinder+Cylinder	$fv(cylinder1), fv(cylinder2),$ $angle(cylinder1\_axis, cylinder2\_axis),$ $min\_distance(cylinder1, cylinder2)$
Plane+Cylinder	$fv(cylinder), fv(plane),$ $angle(plane\_normal, cylinder\_axis)$
Plane+Plane+Cylinder	$fv(plane1, cylinder), fv(plane2, cylinder)$

#### 4.1 Feature Vectors for Sets of Primitive Shapes

Feature vectors for sets of primitive shapes can be defined using geometric properties of the primitive shapes. These features contain not only the properties of each primitive shape, but also the geometric properties relating different primitive shapes together.

Some of the feature vectors used for primitive shapes their combinations are defined in Table 1. It can be seen that the feature vector consists of (a) the intrinsic features of each shape and (b) the geometric relations between the shapes. Hence, feature vectors for more complicated sets can be constructed using this approach. It should be noted that these sets need not be complete.

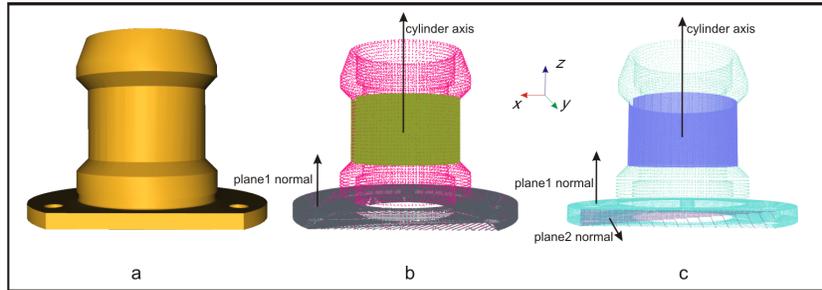
Given a model with a set of primitive shapes  $[P]_m$ , and sets of complete shapes  $[[P]_i]_c$  (where  $[P]_i \subset [P]_m$ ), the set of primitive shapes  $[P]_s$  detected in the scene can be matched with each of the complete sets in the model, using the corresponding feature vectors. The distance between the feature vectors provides a metric for recognition of objects.

#### 4.2 Constraint Processing for Complete and Minimal Primitive Shape Set Detection

The constraints  $(C_{pi}, C_{ni})$  enforced by each primitive shape  $P_i$  are stacked into two matrices  $C_p$  and  $C_n$  (each having 3 columns), where  $C_p$  represents the combination of constraints for the position and  $C_n$  represents the combination of constraints on the orientation. The constraints are complete if the matrices  $C_p$  and  $C_n$  both have rank 3. Fig. 4 shows an example of a complete set of primitive shapes. An algorithm for detecting minimal sets is presented in Algorithm 1.

#### 4.3 Constraint Solving for Pose Estimation

Let us consider the object's frame to be specified at position  $t$  and orientation (rotation matrix)  $R$ . This can also be represented in the form of a homogeneous



**Fig. 4.** *Primitive shapes groups in a sample object.* (a) Sample object. The cylinder fixes the position of the object in x and y axes, the plane1 fixes the position along the z-axis, and plane2 fixes the position in y-axis. The cylinder and plane1 both fix the rotation of the object in x and y axes, and plane2 fixes the rotation in z and x axes. (b) The pose of the object is fully defined in 3D. However, the cylinder axis direction and the normal direction of plane1 are the same. Hence, rotation of the object along the z-axis is not fixed. (c) Position and orientation of the object are both fully defined and these primitive shapes form a minimal set.

transformation matrix  $T$ . From the object models, the relative pose (partially defined)  $T_i$  of each primitive shape  $P_i$  w.r.t. the object's frame is available. To obtain the pose of the object, we need to solve linear equations of the form  $A_{(p,n)} \times x_{o(p,n)} = b_{o(p,n)}$  and where  $A$  is the matrix containing the constraints, and  $x_{o(p,n)}$  is the position and orientation respectively of the object's frame of reference.

So far the constraints have been defined in the primitive shape's frame of reference  $x_{(p,n)}$  and provide linear equations of the form  $C_{(p,n)} \times x_{(p,n)} = b_{(p,n)}$ . The values of the vectors  $b_p$  and  $b_n$  are 0 and 1 respectively. Since we are now trying to estimate the pose of the object's frame, these  $b_{(p,n)}$  values will be modified to  $b_{o(p,n)}$  in the following way:

- Position:  $x_p = R_i \times x_{op} + t_i$ . Hence,  $C \times x_p = b_p \iff (C \times R_i) \times x_{op} = b_p - C \times t_i$ .
- Orientation:  $x_n = R_i \times x_{on}$ . Hence,  $C \times x_n = b_n \iff (C \times R_i) \times x_{on} = b_n$ .

In this way, the constraints defined by the primitive shapes can be solved to obtain the complete pose of the object. If the constraints are complete, the pose is uniquely defined. Otherwise, the constraint solver returns one possible solution.

## 5 Evaluation and Applications

Fig. 5 shows some results obtained by applying the proposed primitive shape decomposition approach to the Object Segmentation Database [17].

**Algorithm 1** Detecting minimal and complete primitive shape sets

---

**Input** :  $[P_i]$  (set of primitive shapes)  
**Output** :  $[[P_i]_{min}], [[P_i]_{complete}]$  (sets of minimal and complete primitive shapes)  
**forall**  $P_i$   
     $min\_flag \leftarrow true$   
     $[P]_{candidate} \leftarrow P_i$   
    **forall**  $P_j, j > i$   
         $[P]_{candidate} \leftarrow [P]_{candidate} \cup P_j$   
        **if**  $check\_complete([P]_{candidate})$   
             $[[P_i]_{complete}] \leftarrow [[P_i]_{complete}] \cup [P]_{candidate}$   
            **if**  $min\_flag$   
                 $min\_flag \leftarrow false$   
                 $[[P_i]_{min}] \leftarrow [[P_i]_{min}] \cup [P]_{candidate}$   
            **endif**  
        **endif**  
    **endforall**  
**endforall**

---

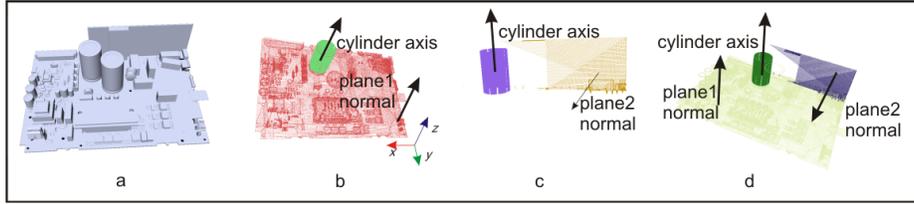


**Fig. 5.** Some results of the primitive shape decomposition algorithm on the Object Shape Database

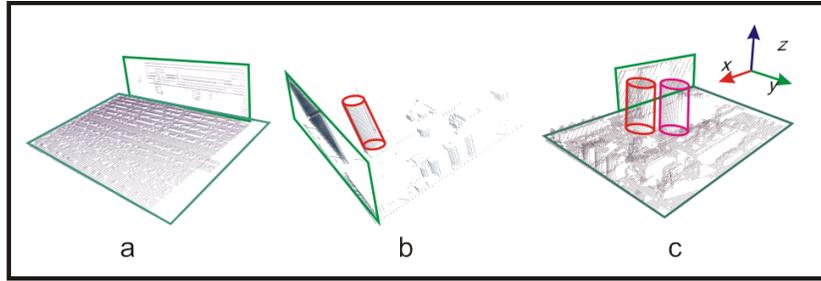
Fig. 6 shows examples of (b) not complete, (c) minimal and (d) complete sets of primitive shapes for the same object.

### 5.1 Calculating detectability of objects from viewpoints

Using the primitive shape sets approach, it can be calculated whether an object's pose can be fully estimated using a set of visible primitive shapes. This knowledge is important in object recognition and pose estimation problems, where primitive shape based approaches might fail due to insufficient data available from a certain viewpoint. The approach presented in this work can detect such problems. Fig. 7 shows an example of different sets of primitives detected from different viewpoints.



**Fig. 6.** Primitive shapes groups in a sample object. (a). The sample object. The cylinder fixes the position of the object in x and y axes, plane1 fixes the position along z-axis, and plane2 fixes the position in y-axis. The cylinder and plane1 both fix rotation of the object in x and y axes, and plane2 fixes rotation in z and x axes. (b) Position of the object is fully defined in 3D. However, the cylinder axis direction and normal direction of plane1 are the same. Hence, rotation of the object along z-axis is not fixed. (c) Position and orientation of the object are fixed and these primitive shapes form a minimal set. (d) Position and orientation of the object are fully defined. These primitive shapes form a complete set but not a minimal set.



**Fig. 7.** Primitive shapes groups for different views of an object. Using primitive shape sets, it can be calculated whether an object's pose can be fully estimated from a viewpoint. In (a) the position of the object along y axis is not determined. In (b) and (c) the complete pose of the object can be estimated. (Note: The detected primitive shapes are highlighted manually to make the image clearer.)

## 6 Conclusion and Future Work

In this paper, an algorithm for object recognition and pose estimation that is based on geometric constraints provided by the set of primitive shapes which the object model consists of has been presented. Detection of minimal and complete sets of primitives is an important feature of this formulation, which enables several applications such as determining whether and with what restrictions an object's pose can be estimated from a certain viewpoint. Future work includes incorporation of more shape primitives into the framework. The current approach for determining minimal and complete sets is designed to be a complete approach, which could be sped up using heuristic approaches.

## References

1. Hu, G.: 3-d object matching in the hough space. In: Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on. Volume 3. (1995) 2718–2723 vol.3
2. Somani, N., Dean, E., Cai, C., Knoll, A.: Scene perception and recognition in industrial environments for human-robot interaction. Proceedings of the 9th International Symposium on Visual Computing (2013)
3. Schnabel, R., Wessel, R., Wahl, R., Klein, R.: Shape recognition in 3d point-clouds. In Skala, V., ed.: The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008, UNION Agency-Science Press (2008)
4. Papazov, C., Haddadin, S., Parusel, S., Krieger, K., Burschka, D.: Rigid 3D geometry matching for grasping of known objects in cluttered scenes. International Journal of Robotic Research **31** (2012) 538–553
5. Papazov, C., Burschka, D.: An efficient ransac for 3d object recognition in noisy and occluded scenes. Proceedings of the 10th Asian conference on Computer vision - Volume Part I (2011) 135–148
6. Thomas, U.: Stable pose estimation using ransac with triple point feature hash maps and symmetry exploration. In: MVA. (2013) 109–112
7. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th international conference on Multimedia. MULTIMEDIA '07, New York, NY, USA, ACM (2007) 357–360
8. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. Vis. Comput. **27** (2011) 963–976
9. Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: Computer Vision Workshops (ICCV Workshops), 2009. (2009) 689–696
10. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the viewpoint feature histogram. In: Intelligent Robots and Systems (IROS), 2010, IEEE (2010) 2155–2162
11. Biegelbauer, G., Vincze, M.: Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation. In: Robotics and Automation, 2007 IEEE International Conference on. (2007) 1086–1091
12. Holz, D., Holzer, S., Rusu, R.B., Behnke, S.: Real-time plane segmentation using rgb-d cameras. In: RoboCup 2011: Robot Soccer World Cup XV. Springer (2012) 306–317
13. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23** (2001) 1222–1239
14. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26** (2004) 1124–1137
15. Delong, A., Osokin, A., Isack, H., Boykov, Y.: Fast approximate energy minimization with label costs. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. (2010) 2173–2180
16. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. Int. J. Comput. Vision **96** (2012) 1–27
17. Richtsfeld, A., Morwald, T., Prankl, J., Zillich, M., Vincze, M.: Segmentation of unknown objects in indoor environments. In: Intelligent Robots and Systems (IROS), 2012. (2012) 4791–4796