 DOI: 10.3217/978-3-85125-528-7-13

Explaining Point Cloud Segments in Terms of Object Models

Manuel Lang[1] and Justus Piater[1]

Intelligent and Interactive Systems
Institute of Computer Science
University of Innsbruck, Austria
{*csae6836,justus.piater*}*@uibk.ac.at*

Abstract
Segmenting the signal of a 3D-sensor represents a core problem in computer vision. Describing segments at the object level is a common requirement for higher-level tasks like action recognition. Non-parametric techniques can provide segmentation without prior model information. However, they are also prone to over- and under-segmentation, especially in case of high occluded scenes. In this paper we propose an approach to segmenting a 3D scene based on a set of known object models. Six-degree-of-freedom (6DOF) model poses result from recognition and pose estimation by exploiting distinct object shapes acquired from a non-parametric segmentation stream. The aligned object models are used in order to resolve over- and under-segmentation by following a bottom-up strategy. Segmentation refinement results from contracting and subdividing input segments in accordance to aligned object models. The proposed algorithm is compared to a trivial model-based segmentation approach that neglects the segmentation stream. Both approaches are evaluated on a set of 24 scenes which are divided into four different complexity categories. The complexity of the scenes ranges from simple to advanced, objects are placed in sparse configurations as well as highly occluded compositions.

1. Introduction

Describing point cloud segments at the object level is of significant importance in the area of computer vision. Having a mechanism that allows to discriminate between individual objects in a captured scene can be useful for higher-level tasks like action recognition, planning and execution [2, 18]. Depth information can provide valuable cues for tasks like segmentation, recognition, pose estimation and tracking [1, 3, 6, 16]. A major challenge is to apply recognition and pose estimation in occluded environments, where scenes are captured by low-resolution RGBD-sensors. This work concentrates on recognition, pose estimation and segmentation of known objects which are part of an assembling task. The objects are placed in table-top scenes that are captured by a Kinect sensor.

The main contribution of this paper can be summarized as follows. Starting from a given model-free[1] segmentation input stream, we propose to execute segment-based object recognition and pose estimation by following a bottom-up strategy. We present a combined recognition, pose estimation and segmentation workflow that exploits geometrical cues delivered by the segments that are computed

[1]In the context of this paper the term *model-free* means that the underlying process does not rely on object models that have to be specified by a supervisor.

by a model-free segmentation process. The complexity of the recognition task is reduced stepwise by handling large segments before small segments. The input segmentation is refined iteratively by exploiting collected 6DOF model pose information. Recognition and pose estimation rely on object models that are specified by 3D meshes as shown in figure 1. Object recognition is bound to certain time constraints, therefore the proposed algorithm does not execute in real-time. The utilized segmentation stream uses color information as its main cue. In contrast to object recognition, which has been restricted to geometrical information. Omitting color information in the latter case has been motivated by the surface characteristics of the evaluated object dataset. The proposed algorithm does not necessarily rely on color cues. In general, it can be applied with any adequate point cloud segmentation input.

Figure 1: The set of object models that are used for recognition and pose estimation.

2. Related Work

Exploiting low-level processing outcomes in higher-level tasks is a fundamental paradigm in computer vision [4, 8, 19]. At present, there exist many segmentation methods that apply to RGBD data [1, 7, 13, 9]. Global surface descriptors are commonly applied to pre-segmented scenes [4]. In this paper we concentrate on local descriptors [5]. The latter type is more suitable for our dataset, since it is more robust against clutter and occlusion. Model information is frequently used for object tracking in videos. The method proposed in [14] uses model information to track 6DOF poses. A RGBD-based segmentation and tracking approach that uses adaptive surface models is proposed in [10]. Our approach concentrates on a combination of object recognition, pose estimation and segmentation in RGBD-images.

3. Background

The following sections provide information about the methods that have been utilized in this paper. Recognition and pose estimation is addressed in the subsequent section 3.1.. Section 3.2. introduces a method that delivers model-free segmentation. The model-based point cloud segmentation that is described in section 3.3. acts as a baseline for the bottom-up segmentation approach proposed in section 4.2..

3.1. Point-based Object Recognition and Pose Estimation

At present, there exists a large variety of different object recognition and pose estimation approaches. An appropriate method should be robust against noise which is introduced by the sensor and it should provide reliable results even in the case of occluded scenes. Scenes are captured by a depth-sensor and object models are represented as point clouds that are sampled from 3D meshes. The method used in this paper estimates 6DOF poses by applying a point-based recognition pipeline [3]. The pipeline is publicly available as part of the Point Cloud Library (PCL) [16]. Figure 2 shows the single steps that are executed in order to recognize the objects in the scene. The first stage extracts keypoints from

model and scene point clouds. In general, keypoints are defined by detecting characteristic surface points. A simple and efficient alternative is to sample keypoints uniformly from the surface. The local geometry of each keypoint is described by the *Signature of Histograms of Orientation* (SHOT) descriptor [17], which delivers favorable results for the evaluated dataset. PCL provides a variety of different descriptor implementations. A comprehensive comparison can be found in [5]. Correspondences are generated by matching scene descriptors against a database of offline computed model descriptors. The next step clusters geometrically consistent correspondences into groups. Starting from a seed correspondence $c_i = \{p_i^m, p_i^s\}$ (p_i^m and p_i^s denote corresponding key points of model and scene), geometrical consistency follows from the following relation

$$|||p_i^m - p_j^m||_2 - ||p_i^s - p_j^s||_2| < \varepsilon \tag{1}$$

where ε defines a distance threshold between the keypoints. A minimum of three correspondences is required to estimate a 6DOF pose. The absolute orientation step eliminates correspondences that are not consistent with a unique 6DOF pose. The utilized recognition pipeline provides an optional iterative closest point (ICP) refinement step, which can be applied on the recognized hypotheses. The number of ICP iterations has been set to a low value. Running more than 5 ICP iterations on the given dataset does not result in significant recognition improvements. The final hypothesis verification step determines a set of non-conflicting model hypothesis that are in accordance with the scene point cloud. Hypothesis that result from unexpected objects within the scene have to withstand the following quality measurement. An acceptance function evaluates the number of supported model points that are close to scene points, as well as the number of unsupported model points (visible model points that have no counterpart in the scene). A detailed description of the hypothesis verification algorithm that has been utilized in this paper is given in [12].

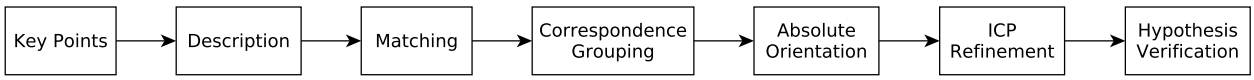


Figure 2: Recognition pipeline used in this paper.

3.2. Model-Free Point Cloud Segmentation

Segmentation results from summarizing interesting and distinguishable image properties. Higher-level visual tasks like object recognition and pose estimation can benefit from such condensed image representations. A method that segments the signal of a RGBD-sensor, without explicit object model information has been presented in [1]. Homogeneous regions (segments) are generated by using color information. In addition, the method exploits depth information in order to support the segmentation and tracking process. Figure 3 shows two example scenes that have been segmented by this method. The segmentation result depends on several factors like scene density, degree of occlusion, object geometry, light conditions, etc.

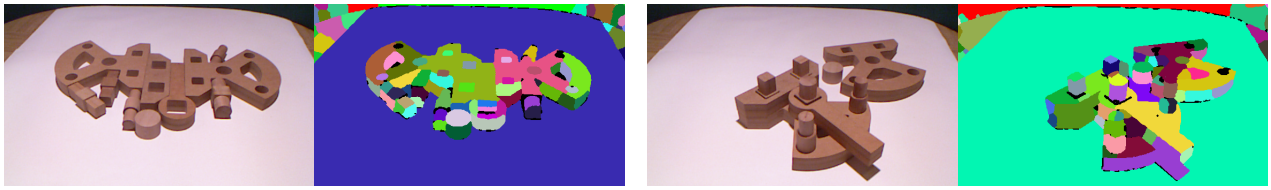


Figure 3: Point cloud segmentation generated by a color-based model-free method.

3.3. Model-based Point Cloud Segmentation

A trivial model-based point cloud segmentation results from evaluating the point vicinity of recognized object models. Object models are aligned with the scene point cloud by applying point-based methods, as described in section 3.1.. It is reasonable to assume that a model point that is close to a scene point indicates a *model-explained* segment membership of this point. Spatial decomposition techniques such as kd-trees provide an efficient structure to determine the k closest points of a query point [15]. The set of scene points that are explained by an aligned object model results as follows. Each point that has been sampled from the model point cloud defines k nearest neighbors (kNN) in the scene point cloud. The nearest neighbor search is carried out in a kd-tree, which represents the scene point cloud. Choosing the value for k results in a trade-off between segment density and sharpness of the segment edges.

4. Segment-based Object Recognition and Pose Estimation

Rising the degree of occlusion in a scene inevitably complicates the segmentation process. Nevertheless, the set of regions that result from the model-free segmentation method described in section 3.2. can preserve a certain amount of object characteristics, even in the occluded case. This motivates a segment-based recognition and pose estimation approach where the model-free segmentation acts as main input. Single segments like the one shown in figure 3 are often not expressive enough to apply recognition and pose estimation on them. Many of them show less variation in surface-normal orientation. We propose to generate larger surface patches in order to increase the recognition output. Surface patches are created by clustering a set of adjacent segments together. Figure 4 provides an overview of how segment-based model poses are generated iteratively in order to refine model-free segmentation in a bottom-up way. In the rest of this paper, the terms surface patch and segment are interchangeable, since single segments can also act as simple surface patches.

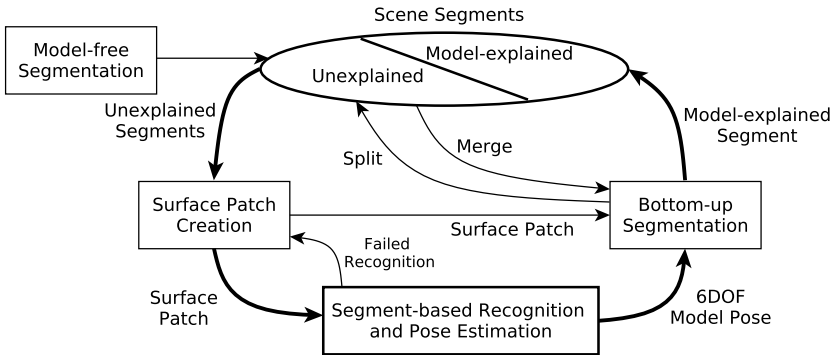


Figure 4: Iterative application of segment-based object recognition and pose estimation.

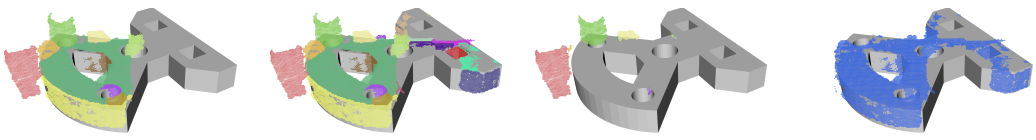
4.1. Adaptive Correspondence Grouping

The correspondence-based recognition and pose estimation method that has been introduced in section 3.1. searches for a set of non-conflicting hypotheses that describe the whole scene at once. In contrast, we propose a segment-based bottom-up strategy. This approach is motivated by two considerations. Firstly, restricting recognition and pose estimation to a surface patch, that preserves certain object characteristics, could reduce the number of wrong hypotheses. Secondly, following a bottom-up strategy that handles large surface patches early, reduces the complexity of the recognition task for smaller segments. The latter consideration is gaining relevance if the scene is a composition of

large and small objects. The proposed approach utilizes the recognition pipeline shown in figure 2. Model hypothesis are computed from consistent correspondence groups, as described in section 3.1.. However, in this case the proposed algorithm adapts the number of correspondences that are required to form a consistent group. According to [3] the correspondence grouping threshold trade-offs the number of correct recognition for the number of wrong recognitions. In general the size of the group can range between three (the minimum required to compute a 6DOF pose) and the number of correspondences that are found in total. A high threshold generates few hypotheses whereas a low threshold leads to many hypotheses. An optimal threshold is influenced by many factors like surface patch size, level of over- and under-segmentation, object similarity, object geometry and also the noise-level of the 3D-sensor. We propose to adapt the correspondence grouping threshold in accordance to the hypothesis verification process which is the last stage in the recognition pipeline shown in figure 2. Starting from a large value the correspondence grouping threshold is reduced stepwise until at least one hypothesis survives the verification process. If the threshold falls below the absolute minimum of three, recognition fails. The acceptance function of the hypothesis verification process also offers opportunities for a segment-based parameter tuning. The thresholds for the number of supported and unsupported scene points, as described in section 3.1., can be weaken if the surface patch size exceeds a certain threshold. This adjustment is justifiable since large surface patches commonly generate fewer hypothesis that are more discriminable.

4.2. Bottom-up Segmentation

The basis for the bottom-up segmentation process is a 6DOF model pose that results from segment-based object recognition and pose estimation. In contrast to the trivial model-based segmentation process that has been described in section 3.3., we propose a recycling of the model-free segmentation stream. According to figure 4, model-free (unexplained) segments are merged and splitted in accordance to the recognized object model that has been placed at the estimated pose. The segmentation process can be described as follows: If recognition fails surface patch creation restarts with the next largest segment. In case of successful recognition, the initial surface patch is extended with parts of unexplained segments that are covered by the aligned object model. Covered segment parts are determined by applying a segment-based radius search in a kd-tree, similar to the approach described in section 3.3.. The search radius is set to a fraction of the object model size. Surface patch parts that are not covered by the recognized object model are separated from the current surface patch and fed back into the recognition process. The process restarts until each unexplained segment becomes part of a model-explained segment or gets labeled as unrecognizable. The single steps of the segmentation process are shown in figure 5. Figure 5a shows the recognized object model that has been aligned with the initial surface patch. Figure 5b shows the extension of the initial surface patch. The separation of non-covered segment parts is shown in 5c. The final result of the model-explained segment can be seen in figure 5d.



(a) (b) (c) (d)

Figure 5: Bottom-up segmentation. (a) Object model aligned with surface patch. (b) Merging of covered segments. (c) Splitting of non-covered (unexplained) segments. (d) Final segmentation result.

5. Results

The proposed algorithms have been evaluated on 24 different scenes which are divided into four complexity categories. The first category contains simple object compositions where objects are widely spread over the field of view. Category two consists of dense scenes that are commonly under-segmented. The third category contains disordered scenes, showing a high degree of clutter. The last and most challenging category contains objects that are assembled together, which results in a high degree of occlusion. Figure 7 shows an instance of each category. The proposed segment-based bottom-up approach is compared to the point-based method that has been described in section 3.3.. The algorithms have been tested on an Intel(R) core(TM) i5 2.53GHz CPU (multiple cores) with 7.7GB RAM. The average scene execution time[2] of the segment-based approach is 154.38 seconds. The point-based approach executes in 129.37 seconds.

5.1. Recognition Rate

Table 1 summarizes the recognition results of the object models shown in figure 1. As shown in the table, segment-based adaptive correspondence grouping (CG) outperforms the point-based method for almost all models. Figure 6 shows a more detailed comparison between all four evaluated complexity categories. The low *bolt sensor* rating is caused by the segment-based parameter tuning of the hypothesis verification process that has been discussed in section 4.1.. In this case, the verification process eliminates too many reasonable hypotheses, which finally leads to confusion with similar looking *bolt angular* and *shaft* objects.

method / **model**	point-based CG	segment-based CG
faceplate	91.67	97.92
separator	83.33	100.00
pendulum	75.00	87.50
shaft	95.83	100.00
bolt angular	79.17	85.42
bolt sensor	75.00	60.42
pendulum head	83.33	95.83
average	82.92	**87.08**

Table 1: Recognition rate comparison of the point-based baseline method and the proposed segment-based method. CG - Correspondence Grouping

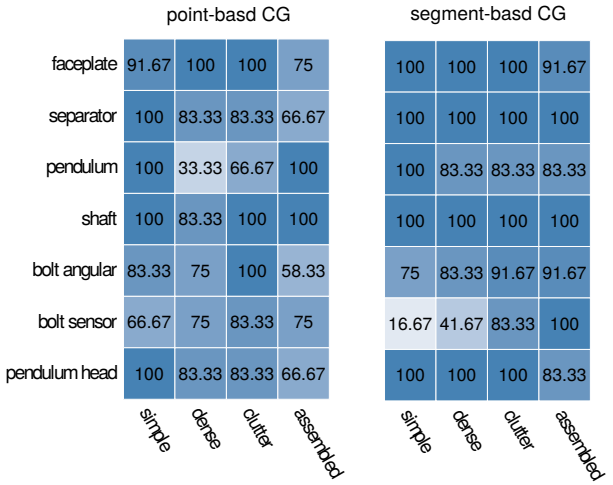


Figure 6: Recognition rate comparison between four evaluted scene complexities.

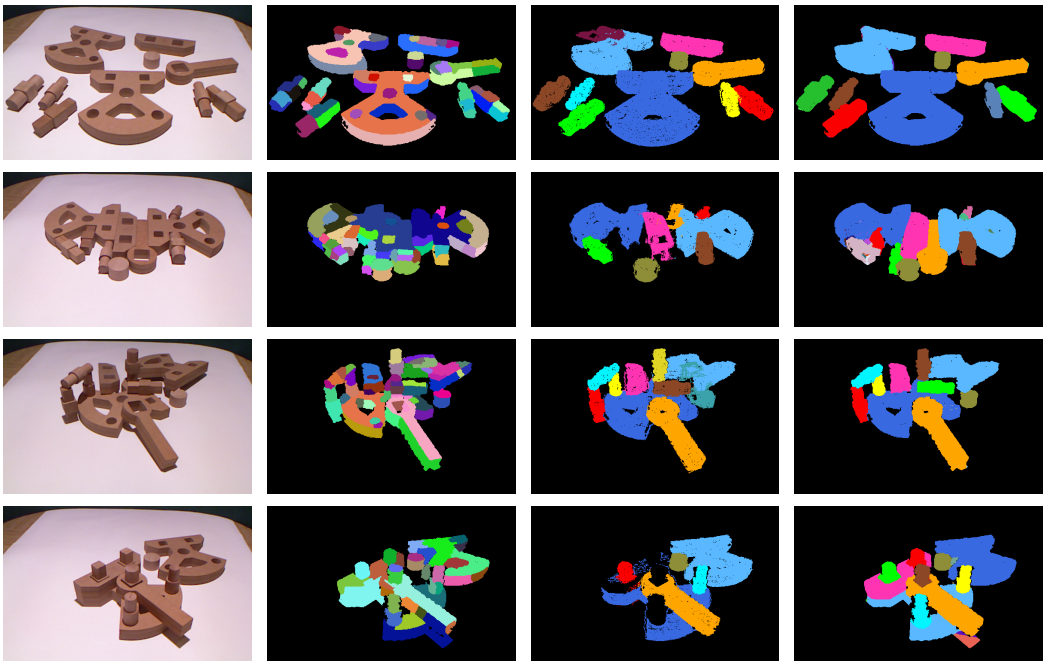
5.2. Segmentation

Figure 7 shows a segmentation comparison of four selected scenes. As shown in the image, the segmentation quality strongly depends on the accuracy of the estimated model poses. Object confusion impairs the segmentation result. The bottom-up segmentation benefits from the recycling of model-free segments. The segment recycling results in sharper edges when compared to the trivial model-based segmentation method. The destructive characteristic of the model-based segmentation results from an inherently trade-off between sharp segment margins and segment density.

[2]The real-time model-free segmentation process, which is not part of this evaluation, relies on a GPU-based system.

6. Conclusion

We have presented a segment-based object recognition and pose estimation approach. The proposed bottom-up segmentation strategy reduces the complexity of the recognition task in an iterative way. The geometrical cues of the model-free input segmentation can successfully be exploited in order to improve recognition rates in occluded scenes. The proposed segment-based recognition and pose estimation approach relies on correspondence-based recognition. False hypothesis are suppressed by adapting the cardinality of consistent correspondence groups. The estimated 6DOF pose information can effectively be used in order to resolve over- and under-segmentation of the model-free input stream. The suitability of our approach was demonstrated on 24 scenes. The complexity of the evaluated dataset reaches its maximum in assembled object compositions. The efficiency of the segment-based object recognition and pose estimation is bound to the amount of under-segmentation in the surface patch.



(a) (b) (c) (d)

Figure 7: Segmentation comparison. (a) RGB input. (b) Model-free segmentation. (c) Model-based segmentation (baseline). Unrecognizable objects are colored black. (d) Bottom-up segmentation.

References

[1] A. Abramov, J. Papon, K. Pauwels, F. Wörgötter, and B. Dellen. Depth-supported real-time video segmentation with the kinect. In *IEEE workshop on the Applications of Computer Vision WACV*, 2012.

[2] Eren Erdal Aksoy, Alexey Abramov, Johannes Dörr, KeJun Ning, Babette Dellen, and Florentin Wörgötter. Learning the semantics of object-action relations by observation. *I. J. Robotic Res.*, 30:1229–1249, 2011.

[3] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Tutorial: Point cloud

library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robot. Automat. Mag.*, 19:80–91, 2012.

[4] Aitor Aldoma, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, and Gary R. Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In *ICCVW*, 2011.

[5] Luís A. Alexandre. 3d descriptors for object and category recognition: a comparative evaluation. In *in Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[6] Renaud Detry and Justus H. Piater. Continuous surface-point distributions for 3d object pose estimation and recognition. In *ACCV*, 2010.

[7] Aleksey Golovinskiy, Thomas Funkhouser, and Traac Light Car. Min-cut based segmentation of point clouds. 2009.

[8] Aleksey Golovinskiy, Vladimir G. Kim, and Thomas A. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *ICCV*, 2009.

[9] Steven Hickson, Stan Birchfield, Irfan A. Essa, and Henrik I. Christensen. Efficient hierarchical graph-based segmentation of rgbd videos. In *CVPR*, 2014.

[10] Farzad Husain, Babette Dellen, and Carme Torras. Consistent depth video segmentation using adaptive surface models. *IEEE T. Cybernetics*, 45:266–278, 2015.

[11] Ajmal S. Mian, Mohammed Bennamoun, and Robyn A. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1584–1601, 2006.

[12] Chavdar Papazov and Darius Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *ACCV*, 2010.

[13] Jeremie Papon, Tomas Kulvicius, Eren Erdal Aksoy, and Florentin Wörgötter. Point cloud video object segmentation using a persistent supervoxel world-model. In *IROS*, 2013.

[14] Karl Pauwels, Leonardo Rubio, Javier Díaz, and Eduardo Ros. Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues. In *CVPR*, 2013.

[15] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. In *DE*, 2009.

[16] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *ICRA*, 2011.

[17] Samuele Salti, Federico Tombari, and Luigi di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[18] Emre Ugur and Justus H. Piater. Refining discovered symbols with multi-step interaction experience. In *HUMANOIDS*, 2015.

[19] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.