Hierarchical Scene Segmentation and Classification

Andre Ückermann, Christof Elbrechter, Robert Haschke and Helge Ritter

***Abstract*— The presented real-time scene segmentation approach yields a complete hierarchy of segmentation hypotheses – ranging from loosely coupled groups of objects to strongly coupled individual object parts or surfaces. Using an object classifier that traverses this hierarchy in a top-down manner, we can focus on the correct level of abstraction in a task-specific manner and thus avoid over- and under-segmentation in cases where our previous approach failed due to missing world knowledge. The approach allows for identification of object parts or complete objects (e.g. a mug composed from the handle and its inner and outer surfaces) in a uniform and scalable framework.**

I. INTRODUCTION

Real-time scene segmentation and object tracking are important tasks in real-world human-robot-interaction involving dynamically changing environments. Example tasks include online collision checking, grasping and manipulation of moving objects, or pointing gestures to reference objects. Despite its importance, real-time capable approaches to generic and robust scene segmentation are scarce.

Recent state-of-the-art methods [1], [2], [3] follow a two-step approach: in the first phase, the scene is (over)segmented according to some homogeneity criterion, finding regions of smoothly varying surface normals [1] or planar surface patches [2], [3]. In a subsequent step, these low-level image segments are grouped to form object hypotheses according to some high-level grouping rules, using heuristic rules [1] or convexity of adjacent super-voxels [3], or using a SVM to predict connectivity of pairs of regions [2]. In all cases, the connectivity structure of image patches can be encoded within a graph with edge weights indicating the strength of connectivity between regions. To determine the final segmentation hypothesis, the graph nodes are clustered according to some grouping criterion. While [3] employ region growing and thus only combine locally connected object parts, our previous work was able to recombine aligned object parts separated due to occlusion [1] using a graph-cut algorithm with a fixed cost threshold.

In this paper, we extend this work to create a *full hierarchy* of grouping hypotheses – ranging from spatially neighbored point cloud blobs to object parts and individual object surfaces. Explicitly representing the connectivity structure of the scene, we can postpone the final decision for a grouping hypothesis and exploit the segmentation hierarchy by a higher-level decision process (involving world knowledge) to decide for the proper level of granularity and to find the optimal grouping hypothesis in a task-specific manner. To this end, we propose a simple nearest-neighbour classifier that traverses the hierarchy in a top-down fashion and stops if an object is recognized with high confidence.

This work was supported by the German Collaborative Research Center "CRC 673: Alignment in Communication" and the Center of Excellence Cognitive Interaction Technology (CITEC), both granted by the DFG. The authors are with the Neuroinformatics Group at Bielefeld University, Germany. {aueckerm|celbrech|rhaschke|helge}@techfak.uni-bielefeld.de

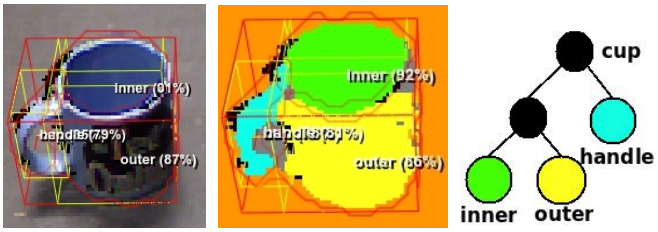

Fig. 1. Segmentation and classification result of a mug (red bounding box) and its sub-parts (yellow b.boxes) obtained from smooth surface patches (middle). The resulting segmentation tree groups loosely coupled image regions first, proceeding to more strongly coupled regions.

II. HIERARCHICAL SEGMENTATION

In the first segmentation phase, we over-segment the scene into regions of smoothly varying surface normals employing the following processing pipeline:

1) three-stage smoothing (median, temporal, Gaussian)
2) calculation of surface normals (from cross product)
3) detection of object edges from angle between normals ($n_1 \cdot n_2 < \Theta$) and Euclidean distance of adjacent points
4) connected component analysis to assign unique IDs to all surface regions
5) assignment of edge points to closest surface region.

To represent the potential connectivity structure between low-level image regions we determine a connectivity graph according to three heuristic rules, namely (i) cut-free adjacency, (ii) co-planarity and (iii) similar curvature of surface pairs. Each criterion contributes a subset of edges within the graph. For more details we refer to [1]. In contrast to previous work, also remaining edge points – representing small objects or protruding parts – will be considered as individual image regions and become nodes within the connectivity graph. To do so, they are segmented using 3D Euclidean distance and local curvature as homogeneity criterions for region growing.

The hierarchical segmentation algorithm aims to find a hierarchical clustering of graph nodes into groups of increasing connectivity strength. To this end, we first assign connection weights to all graph edges indicating the connectivity strength or cutting costs. Subsequently we iteratively apply the minimum cut algorithm to split nontrivial, connected subgraphs into pairs of subgraphs such that the sum of removed edge weights is minimal at each step. If the initial graph consists of several disconnected subgraphs (cf. Fig. 2),

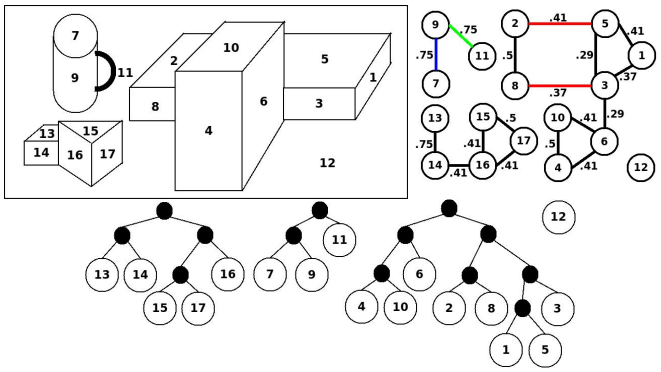

Fig. 2. Connectivity-Graph with initial edge weights and edges from different criteria: cut-free adjacency (black), co-planarity (red), curvature (blue), remaining edge points (green) and the resulting segmentation tree.

each of them will be treated independently and we will arrive at a forest of composition trees, each associated with grouping hypotheses for a sub-scene.

A. Assigning edge weights

The naive approach to compute cutting costs, counting the number of removed edges (corresponding to weights equal one), would correctly separate loosely connected 3-cliques typically arising from boxes (e.g. 4-6-10 and 15-16-17 in Fig. 2). However, this approach will fail to prefer separation of subgraph 13-14 from 15-16-17 to the separation of node 13, because there is only a single connecting edge in both cases, i.e. resulting in identical costs.

Our solution is to assign edge weights $w_{ij} = 1/n$ to all edges (i, j) originating from node i, where n denotes the number of outgoing edges of node i and to average the weights of incoming and outgoing edges to achieve a symmetric cost distribution. This normalizes the costs of any graph cut to the range $[0..1]$ and causes the costs to separate a single node to be equal (or close to) one. In this context, please notice, that we consider only cuts into *two* subgraphs. On the other hand, smaller cliques gain stronger internal connectivity compared to their bridging links to adjacent cliques: Although individual edge weights within the cliques 1-3-5, 4-6-10 and 15-16-17 are small, each cut dividing them would have costs close to one. In contrast, the bridges between cliques have even lower weights and thus become first candidates for cutting. Nodes 13 and 14 become more strongly connected than 14 and 16, because there is only a single edge from node 13.

Figure 2 illustrates an example comprising edges introduced due to different heuristic criteria: cut-free adjacency (black), co-planarity (red), curvature (blue), and remaining edge points (green). Analyzing the resulting weights, we observe two problems for a proper hierarchy deduction:

(1) The subgraph 2-5-1-3-8 (after separation from 4-6-10) has two minimum cuts with identical costs: one splitting off node 1 and the other (preferred one) splitting off 2-8.

(2) The subgraph 7-9-11 also offers two possible cuts: removing the edges originating from the curvature or the remaining points rules resp.

To resolve these ambiguities and to prefer correct groupings we propose to employ a-priori confidence weights for each edge type: Edges originating from the cut-free adjacency criterion have highest confidence and thus should be least preferrable for cuts, when compared with other edges. Edges that were added to link regions classified as "remaining edge points" should have highest cutting preference, since those only correspond to extensions of the main object body. Therefore we propose to assign weights $\omega_a > \omega_p > \omega_c > \omega_r$ to the different criteria cut-free adjacency, co-planarity, curvature, and remaining points respectively.

By multiplying the basic edge weights with these feature-related weights, the relative influence of a single feature is increased or decreased and ambiguities are resolved.

B. Hierarchical Segmentation Tree

The segmentation forest resulting from processing all separated sub-graphs is shown in Fig. 2. For example, the sub-graph 7-9-11 is first divided along the remaining-points edge, separating the mug handle from its body. The two sub-graphs of each cut are added as children of a new virtual parent node that replaces the original graph. The sub-graphs are in turn processed with minimum-cut, now separating the inner and outer surface of the mug that were linked due to the curvature matching criterion. This process is repeated until a single node (corresponding to a single surface patch) remains at the leaf of the tree. These leaf nodes contain the actual point cloud data, while all branching nodes correspond to potential grouping hypotheses. Due to the *minimum* cut strategy, branches at the root have weakest connectivity, while leaf nodes have strongest connectivity.

III. HIERARCHICAL CLASSIFICATION

The obtained region hierarchy plays the central role to realize a very flexible interplay between bottom-up region segmentation and top-down application of world knowledge about the appearance of semantically relevant entities: we use the segmentation hierarchy to control the spatial attention of a region-based classifier that comprises all world knowledge about the appearance of semantically relevant scene entities. Combining the efficient binary splittings guided by the tree structure with a fast (e.g. NN-based) classifier implementation allows the classifier to be applied *at all segmentation scales* without undue computational load. Peaks of classification confidence then indicate salient entities in the scene. Typically, we expect these to correspond to objects. However, the hierarchical application of the classifier admits *simultaneous confidence peaks at several levels* that reflect the detection of salient object parts (at lower levels – e.g. a handle), as well as the detection of salient object groupings (at higher levels – e.g. a heap of apples). Thereby, the region hierarchy connects two complementary representations of regularities in the world very efficiently: *(i)* highly generic, low-level regularities about homogeneity structures that govern segmentation, and *(ii)* specialized, high-level feature correlations that are indicative of objects, object parts or special object configurations. An illustrative example is depicted in Fig. 1, where the mug's handle and body as well its inner and outer surface could be recognized individually.

A. Bag-of-features NN-classifier

For classification we employ a standard nearest-neighbor (NN) classifier that internally works on a bag-of-features model, namely using size, elongation, and color features. The *size* feature approximates the volume of a segment's 3D bounding sphere by performing PCA on the points. The resulting matching function is very efficient as only the scalar volume values of two segments have to be compared. The *elongation* feature computes the ratio of the two largest PCA eigenvalues. The *color* feature uses a binned and smoothed hue-saturation histogram. The matching functions of all individual features are normalized to provide values from zero to one facilitating weighted comparison of individual matching results. The overall matching distance is determined from the maximum of all individual feature distances ensuring good matching for all features.

For efficiency reasons, features are computed in bottom-up fashion, starting at the leaves of the segmentation tree and then propagating to the root by exploiting the pre-computed feature values of a node's children. The performance bottleneck of (k)NN-classification is usually the feature matching step, which is why a lot of research has been focused on its optimization , trying to find highly descriptive, but low-dimensional feature vectors that can be matched efficiently. Our internal implementation allows each feature type to provide its own accelerated matching function.

IV. RESULTS

Table I summarizes segmentation results of recent approaches on the Object Segmentation Database (OSD) [4] showing that our previous approach [1] already performed best considering the ratio of accuracy and processing time.

We now present qualitative results that illustrate the improvements of our new approach over our previous work.

As can be seen from Fig. 1 showing classification results for a mug, our approach can correctly identify all sub parts of an object as well as the object itself. The identification of handles or knobs and of concavities is particularly important for task-specific grasping and for pouring tasks resp. In previous work such object parts were either absorbed within the overall object point cloud or were segmented as separate regions without the link to the main object.

Figure 3 shows situations that can only be segmented and classified correctly by combining bottom-up and top-down processing streams: In both images, a box is separated into two parts due to the occluding cylinder. Although having different color distributions, both parts in the left image belong to the same object (Kinect box). On the other hand, in the right image, both parts actually belong to different, but

TABLE I

SEGMANTATION RESULTS OF RECENT METHODS ON OSD [4].

	true-pos.	false-pos.	false-neg.	time
Ückermann [1]	96.3±4.1	**2.5**±4.5	3.7±4.1	**30-40ms**
Richtsfeld [2]	**97.2**±4.8	5.8±10.3	**2.8**±4.8	1-8s
Wörgötter [3]	90.7±8.7	4.3±2.5	9.3±8.7	550ms

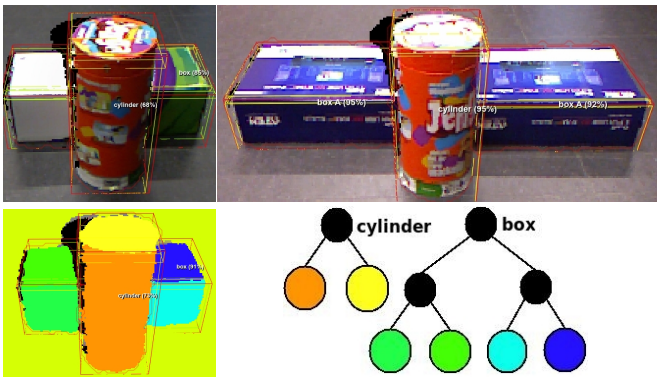

Fig. 3. Example illustrating the selection of the proper grouping level from classification: The box behind the cylinder could be a single object (left: Kinect box) or two separate ones (right). As the classifier knows the appearance of those boxes, it can correctly classify them either on the root level or on the first sub-level of the classification tree.

perfectly aligned boxes. A proper grouping decision can be achieved in this case only if learned appearance models of the objects are exploited. In the corresponding segmentation tree, whose leaf nodes are colored according to the found surface patches of the pre-segmentation phase, the grouping will either combine both branches or split them – depending on the confidence level of the classifier.

A video of the algorithm applied in a complex robotics demonstrator for human-robot-cooperation is shown at youtu.be/gI7c9RC7gKg.

V. SUMMARY AND OUTLOOK

This paper proposed a generic method to yield a hierarchical segmentation tree by iteratively applying minimum graph-cut to a weighted connectivity graph defined on a meaningful over-segmentation of RGB-D images. Combining the bottom-up segmentation algorithm with a top-down classification we were able to improve our segmentation results in complex scenes by autonomously finding the correct grouping level. Furthermore, the approach allows for a task-dependent focus on the grouping hierarchy in order to identify task-relevant object parts, like handles or knobs for grasping and concavities for pouring.

Obviously, the choice of specific rules to create graph connections and their weightings is crucial for the resulting segmentation hierarchy. In this work we relied on heuristic rules already introduced in our previous work [1]. However, several alternatives are possible.

Our method can also be extended to handle both, known and unknown objects at the same time. To do this we can simply resort to our previously presented, fixed threshold-based grouping approach to generate grouping hypotheses for individual trees that cannot correctly be classified.

REFERENCES

[1] A. Ückermann, R. Haschke, and H. Ritter, "Realtime 3D segmentation for human-robot interaction," in *Proc. IROS*, 2013, pp. 2136–2143.

[2] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Learning of perceptual grouping for object segmentation on RGB-D data," *J. Vis. Comun. Image Represent.*, vol. 25, no. 1, 2014.

[3] Stein, Wörgötter, Schöler, Papon, and Kulvicius, "Convexity based object partitioning for robot applications," in *Proc. ICRA*, 2014.

[4] "Object Segmentation Database," http://www.acin.tuwien.ac.at/?id=289.