Project IP-2014-09-3155

Advanced 3D Perception for Mobile Robot Manipulators

SURVEY OF STATE-OF-THE-ART POINT CLOUD SEGMENTATION METHODS

Technical Report

ARP3D.TR1

version 1.0

Robert Cupec, Emmanuel Karlo Nyarko, Ratko Grbić

Josip Juraj Strossmayer University of Osijek

Faculty of Electrical Engineering Osijek

Osijek, 2016.

Ovaj rad je financirala/sufinancirala Hrvatska zaklada za znanost projektom IP-2014-09-3155.

Mišljenja, nalazi i zaključci ili preporuke navedene u ovom materijalu odnose se na autora i ne odražavaju nužno stajališta Hrvatske zaklade za znanost.

This work has been fully supported by/supported in part by the Croatian Science Foundation under the project number IP-2014-09-3155.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Croatian Science Foundation.

1. Segmentation to Objects

In **(Richtsfeld et al., 2012; Richtsfeld et al., 2014)**, an approach for segmenting a RGB-D image into object based on SVM classifier is proposed. The input RGB-D image is first segmented into planar segments and smooth surfaces represented by NURBS. The following relations are defined between the segments:

r_{cu} – mean curvature along 2D patch border,
r_{di} – mean distance along 2D patch border,
r_{vdi} – variance of distance along 2D patch border,
r_{cb} – difference of color along 2D patch border,
r_{ch} – difference of patch colour,
r_{tr} – difference of patch texture,
r_{ga} – gabor filter match,
r_{fou} – fourier filter match,
r_{cu3} – curvature along 3D patch border,
r_{di3} – distance along 3D patch border,
r_{nm} – difference of mean surface normals direction,
r_{nv} – difference of variance of normals direction,
r_{ac} – mean angle of normals of nearest contour points,
r_{dn} – mean normal distance of nearest contour points,
r_{md} – minimum distance between patch borders,
r_{rs} – relative patch size difference.

The graph is defined graph, where patches represent nodes and relations represent edges. The parameters describing the aforementioned relations are computed for pairs of segments and vectors composed of these parameters are assigned to the graph edges, where some of these relations are included in vectors assigned to the edges connecting neighborihg segments, and some of them are included in vectors assigned to the edges connecting non-neighborihg segments. The graph is fully connected, as we defined relations between all surface neighbors, as well as all non-neighbouring surface patches. The introduced relation vectors are used in the training phase to train the two SVM using a hand-annotated set of depth images. The applied SVM provides the probability that two segments connected by a particular relation belong to the same object. The graph segmentation approach proposed in (Felzenszwalb and Huttenlocher, 2004) is applied to segment the graph using the probability values from the SVMs as the pairwise energy terms. The approach is evaluated using OSD (See Section 7). The experimental analysis show that the proposed method results in rather high undersegmentation. The source code of the proposed method is made publically available by the authors at http://www.acin.tuwien.ac.at/?id=316.

In **(Stein et al., 2014)**, a method for segmentation of point clouds to locally convex surfaces is proposed. The method is named Locally Convex Connected Patches (LCCP). The algorithm begins by oversegmenting the input point cloud into supervoxels using VCCS (Papon et al., 2013) and creating supervoxel adjacency graph, whose nodes correspond to supervoxels and edges represent neighboring relations between supervoxels. Edges in the graph are then classified as either convex or concave. Region growing is employed to identify locally convex connected subgraphs, which represent the object parts. The method is evaluated using OSD (See Section 7), where it is applied

to segment the scenes into objects. The approach is also tested on the datased created by the authors, where it is applied to segment the scenes to object parts. Although the proposed method is rather simple, the experimental analysis showed that it has a comparable performance to the method proposed in (Richtsfeld et al., 2012). However, a disadvantage of the proposed approach is that it oversegments hollow objects such as bowls, cups etc.

In **(Ückermann, Haschke and Ritter, 2012)**, a real-time algorithm that segments unstructured and highly cluttered scenes is presented. The algorithm robustly separates objects of unknown shape in congested scenes of stacked and partially occluded objects. The model-free approach finds smooth surface patches, using a depth image, which are subsequently combined to form highly probable object hypotheses. The real-time capabilities and the quality of the algorithm are evaluated on a benchmark database. An autonomous grasping experiment with a Shadow Robot Hand, which employs the estimated shape and pose of objects provided by the algorithm, in a task in which it cleans a table is reported.

In a real scene, most pixels at the boundary of an object are at depth discontinuities. But there is a portion of an object boundary that touches the surface it is resting on and does not have any depth discontinuity across it. This portion of the object boundary is referred to as contact boundary. A "simple" object is defined as a compact region enclosed by depth and/or contact boundary in the scene. In **(Mishra, Aloimonos and Fah, 2009)**, a fixation-based segmentation framework is proposed that segments a region given a point anywhere inside it. This framework is used in **(Mishra, Shrivastava and Aloimonos, 2012)** to extract all the "simple" objects in a RGB-D image. The original framework is modified to include a fixation strategy to automatically select points inside the "simple" objects and a post-segmentation process to select only the regions corresponding to the "simple" objects in the scene. A novel characteristic of the approach is the incorporation of border ownership, i.e, the knowledge about the object side of a boundary pixel. The procedure is tested on a publicly available RGB-D dataset (Lai et al., 2011) with results showing that the proposed method successfully extracts 91:4% of all objects in the dataset.

In **(Gupta, Arbeláez and Malik, 2013)** a segmentation algorithm based on Support Vector Machine (SVM) classifier is presented. First, contours are detected using color and depth information. Then, a hierarchical grouping of the initial segments is performed. Contour detection is formulated as a binary pixel classification problem, where the goal is to separate contour from non-contour pixels. A disk centered at each image location is split into two halves at pre-defined orientations and the features in the two disk-halves for each orientation are compared. The features used for this task are color features proposed in (Arbeláez et al., 2011) and the following geometric features: depth, depth gradient, which identifies the presence of a discontinuity in depth, a convex normal gradient, which captures if the surface bends-out at a given point in a given direction, and a concave normal gradient, capturing if the surface bends-in. SVM classifiers are learnt for each orientation channel independently and their final outputs is combined. A combined gradient is computed by considering the average of all local contour cues in each orientation and taking the maximum response across orientations. The watershed transform of the combined gradient is computed and all pixels on the watershed lines are declared as possible contour locations. In the training phase, the labels from ground-truth manual annotations are transferred to the candidate locations for each orientation channel independently. First, the ground-truth contours in a given orientation are identified, and then the candidate contour pixels in the same orientation within a distance tolerance are declared as positives. The remaining candidates in the same orientation are

declared negatives. SVMs with additive kernels (Maji, Berg and Malik, 2013), which allow learning nonlinear decision boundaries with an efficiency close to linear SVMs are used as classifiers, and their probabilistic output is used as the strength of the oriented contour detectors. Finally, generic machinery proposed in (Arbeláez et al., 2011) is used to construct a hierarchy of segmentations, by merging regions of the initial over-segmentation based on the average strength of the oriented contour detectors. Planar surfaces which are not represented by connected sets in the RGB-D image due to occlusion are grouped by a greedy algorithm which merges superpixels into bigger more complete regions, based on the agreement among the parametric geometric fits, and re-estimates the geometric model. The following measures of the agreement are used: (i) orientation (angle between normals to planar approximation to the two superpixels) and (ii) residual error (symmetrized average distance between points on one superpixel from the plane defined by the other superpixel). A linear function of these two features is used to determine which superpixels to merge. As an output of this greedy merging, a set of nonoverlapping regions which consists of both long and short range completions of the base superpixels is obtained. The performance of the proposed segmentation approach is evaluated using the standard benchmarks of the Berkeley Segmentation Dataset (Arbeláez et al., 2011): Precision-Recall on boundaries and Ground truth Covering of regions as well as using NYUD2 dataset. The Matlab code of the proposed algorithm is available at http://www.cs.berkeley.edu/~sgupta/

Symmetry can be a potentially important cue in determining whether two bodies represent the same object. An approach to detect symmetries is proposed in **(Mitra, Guibas and Pauly, 2006)**.

In **(Karpathy, Miller and Fei-Fei, 2013)**, an algorithm which decomposes the scene into a set of candidate mesh segments and then ranks each segment according to its ”objectness” – a quality that distinguishes objects from clutter. To partition a scene mesh into segments, the mesh is treated as a graph and a graph based segmentation is performed using the algorithm proposed by (Felzenszwalb and Huttenlocher, 2004). Five intrinsic shape measures: compactness, symmetry, smoothness and local and global convexity as well as a recurrence measure (presence of similar objects in other scenes) are proposed as "objectness" measures. Compactness is computed as the ratio of the total surface area of the segment’s mesh to the surface area of its smallest bounding sphere. Symmetry is computed by reflect the segment along a principal axis and measure the overlap between the original segment and its reflection. Smoothness is computed as the average of a smoothness measure computed for each point averaged over all points in a segment. The smoothness measure for a point is computed as entropy of the distribution of the angles of neighboring points projected onto the plane defined by the query point and its normal. Local convexity is computed as the percentage of convex edges between adjacent mesh polygons. Global convexity is computed as average distance from a point on the object to the closest point on the convex hull. Recurrence is computed as the average distance to the top 10 most similar segments in other scenes. To retrieve the most similar segments to a given query segment, all segments within 25% of extent along principal directions in size are considered and the euclidean distance between their normalized shape measures are computed. Each measure is normalized to be zero mean and unit variance during the retrieval. Several options for combining the proposed measures into one objectness score are considered: Simple averaging, Naïve Bayes, Linear SVM, RBF Kernel SVM, and Nearest Neighbor. Ground truth training labels are obtained by manually annotating all extracted segments as being an object or not.

In **(Hickson et al., 2014)**, an efficient and scalable algorithm for segmenting RGBD videos by combining depth, color, and temporal information using a multistage, hierarchical graph-based approach is presented. The authors in their paper introduce the term Toxels, i.e., temporal voxels
The method uses measurements such as color, spatial coordinates, and RGBD optical flow to build a hierarchical region tree for each sequence of n frames (they use $n = 8$), which are then sequentially matched to produce long-term continuity of region identities throughout the sequence. This avoids limitations on the length of the video or on the amount of memory needed due to the high volume of data.
The approach involves four main steps:

1. A graph-based segmentation of 8 consecutive frames using the depth and motion information is performed.
2. An over-segmentation of the frames is done using color and motion information while respecting depth boundaries.
3. Histograms of the resulting regions are used to build a hierarchical segmentation of the spatiotemporal volume represented as a dendrogram, which can then yield a particular segmentation depending on the desired segmentation level output.
4. The final step performs a bipartite graph matching with the 8 previous frames with 4 frames overlapping to enforce the consistency of region identities over time.

The primary contributions are:
(1) A robust and scalable, RBGD video segmentation framework for streaming data.
(2) A streaming method that maintains temporal consistency and can be run on a robot or on videos of indefinite length.
(3) An efficient framework that runs at 0.8 fps but can be downsized to run pairwise in near real-time at 15 fps.
(4) A nonlinear multistage method to segment color and depth that enforces that regions never merge over depth boundaries despite color similarities.
(5) The ability to segment and maintain temporal coherence with camera movement and object movement.
The code and data are publicly available at the paper web page.

While most of work in the field of point cloud segmentation mainly considers decision making and visual processing as two separate tasks, the authors in **(Pajarinen and Kyrki, 2015)** argue that the inherent uncertainty in object segmentation requires an integrated approach that chooses the best decision over all possible segmentations. The proposed approach computed the action a which maximises the expected utility

$$a^* = \arg\max_a \sum_{\mathbf{h}} P(\mathbf{h}) U(\mathbf{h}, a),$$

where utility $U(\mathbf{h}; a)$ is a utility function and $P(\mathbf{h})$ is a probability distribution over object compositions. The probability distribution $P(\mathbf{h})$ is approximated by a particle representation, which is generated by a Markov chain Monte Carlo (MCMC) procedure. The input RGB-D image is oversegmented and possible scene interpretations are restricted to only those hypotheses, where objects consist of the segments obtained by the oversegmentation. Every Markov chain state represents a discrete combination of binary variables, each variable denoting whether two segments are directly connected, i.e. whether they belong to the same object. The Markov chain state are sampled according to the prior probabilities that two segments belong to the same object, which are

computed by the SVM classification proposed in (Richtsfeld et al., 2014). In the experimental evaluation, the proposed approach is applied to the task of moving Lego bricks and to compared to (Richtsfeld et al., 2014).

In **(Gupta et al., 2014)**, an integrated system for scene understanding from RGB-D images is presented. The proposed approach represents a generalization of the system for object detection in RGB images, called R-CNN, proposed in (Girshick et al., 2014), by making effective use of the additional signal – depth. The contour detection is performed by a combination of two approaches: features from (Gupta, Arbeláez and Malik, 2013) and learning framework from (Dollár and Zitnick, 2013). The contour detection approach uses normal gradients and geocentric pose (a per pixel height above ground and angle with gravity) and an RGB edge detector. Object proposal represents a generalization of Multiscale Combinatorial Grouping to RGB-D images which proposes a pool of candidates for potential objects. R-CNN starts with a set of bounding box proposals, computes features on each proposal using CNN and classifies each proposal as being target object class or not with a linear SVM. The CNN is trained in two stages: first, pretraining it on a large set of labeled images with an image classification objective, and then finetuning it on a much smaller detection dataset with a detection objective. The depth image is encoded with three channels at each pixel: horizontal disparity, height above ground, and the angle the pixel's local surface normal makes with the inferred gravity direction. The CNN architecture proposed in (Krizhevsky, Sutskever and Hinton, 2012) is used, which is trained on NYUD2 dataset (see Section 7). The problem was that NYUD2 dataset is about one order of magnitude smaller than the PASCAL VOC dataset, which is used to train CNN in (Krizhevsky, Sutskever and Hinton, 2012). Hence, more data for training hat to be generated and a finetuning of the network had to be performed.

In **(Beale, Iravani and Hall, 2011)**, a segmentation method is proposed, which segments a 2D image sequence into objects using information about the motion of the robot which moves objects and observes their visual feedback.

2. Segmentation into Primitives

In **(Schnabel, Wahl and Klein, 2007)**, an approach for segmentation of point clouds into planes, spheres, cylinders, cones and tori is presented. The method is based on RANSAC. The proposed algorithm is an iterative procedure, where in each iteration the primitive with maximal score is searched using the RANSAC paradigm. New shape candidates are generated by randomly sampling minimal subsets of the input point cloud using a novel hierarchically structured sampling strategy. Candidates of all considered shape types are generated for every minimal set and all candidates are collected in a candidate set. After new candidates have been generated, the one with the highest score is identified using a novel lazy score evaluation scheme, which is designed to reduce the computational cost. The best candidate is only accepted if, given the size of the selected candidate (number of points) and the number of drawn candidates, the estimated probability that no better candidate was overlooked during sampling is high enough. If a candidate is accepted, the points, which are inliers for that candidate, are removed from the point cloud and all candidates which have common points with the selected candidate are deleted from the candidate set. The algorithm terminates as soon as the estimated probability that all relevant primitives are already detected becomes large enough.

In **(Schmitt and Chen, 1991)**, an approach for creating a triangular mesh from depth image is proposed. The algorithm consists of a recursive Delaunay triangulation method followed by region merging. The algorithm starts by representing the entire depth image by two triangles whose vertices are the corners of the image. For each triangle, the point with the maximum distance to the plane defined by the triangle vertices is identified. If this distance is greater than a predefined threshold the triangle is split into smaller triangles whose vertices are the vertices of the original triangle and the maximum distance point. After that, the triangulation is modified in order to obtain a Delaunay triangulation. This procedure is recursively repeated for all triangles until all a triangular mesh is obtained, such that all points of the input depth image lie on a mesh triangle within the user specified threshold. Finally, a merging procedure based on region growing is applied to merge the triangles in planar segments.

In **(Cupec, Nyarko and Filko, 2011)**, a mesh segmentation approach is proposed, which segments a triangular mesh created from a depth image by the method proposed in (Schmitt and Chen, 1991) into approximately convex surfaces. Analogously to (Schmitt and Chen, 1991), the method performs grouping of mesh triangles by a region growing procedure, but using convexity criterion instead of planarity criterion.

The approach proposed in **(Garland, Willmott and Heckbert, 2001)** performs a hierarchical segmentation of a triangular mesh according to a planarity criterion. The proposed algorithm produces a binary tree where each node, except the leaf nodes, represents a segment which is the union of two segments corresponding to its child nodes. The root node is the entire mesh. The procedure starts with a triangular mesh, where each triangle represents an initial segment. The algorithm proceeds by merging two neighboring segments with the minimum merging cost. The merging cost is the integral L^2 distance of the vertices of the triangles grouped in the merged segments from the total least squares plane.

In **(Attene et al., 2008)**, a mesh segmentation approach is proposed, which performs hierarchical segmentation analogously to (Garland, Willmott and Heckbert, 2001), but using as the cost a measure of concavity of the segment obtained by merging of neighboring segments.

In **(Attene, Falcidieno and Spagnuolo, 2006)**, a similar approach is used to segment a mesh into planes, spheres and cylinders.

In **(Reisner-Kollmann and Maierhofer, 2012)**, a method for segmentation of multiple registered depth images into planes, cylinders, spheres and cones is proposed. The input to the algorithm are multiple depth images aligned to each other, e.g. with Iterative Closest Points (ICP) algorithm (Besl and McKay, 1992). The central data structure for the proposed algorithm is a sample point graph which is formed from a subset of points sampled from all input frames. Whenever a new frame is added, the sample graph has to be updated to include newly captured areas. The point cloud is sampled such that the average distance between neighboring points is approximately equal to a user defined parameter, which adjusts the number of node in the graph and provides a tradeoff between accuracy and computational time. The algorithm performs an optimization procedure consisting of three parts: (i) detecting new shapes, (ii) optimizing assignment of points to shapes and (iii) optimizing shapes, which are repeated until convergence. Shape detection is performed by the RANSAC algorithm proposed in (Schnabel, Wahl and Klein, 2007). Assignment of points to shapes is accomplished by graph cuts (Boykov and Jolly, 2001) which optimizes the cost function

consisting of a term which penalizes the points and the shapes and a term which favors assigning neighboring points to the same shape. After the set of assigned points has changed, a refitting step is applied. The geometric error of the shapes is optimized with weighted least-squares. Shapes which have no or only very few points assigned are deleted.

In **(Holz and Behnke, 2014)**, a method for segmentation of depth images into planes, spheres and cylinders is proposed. The method is based on region growing. A seed point is randomly selected from the input depth image. The region growing procedure is initialized by creating a region consisting of the selected seed point and growing this region by adding neighboring points which satisfy a particular criterion. For planar segments, this criterion is that the point must lie on the plane defined by the seed point and its normal within a predefined threshold. In addition to planar segments, smooth surfaces are also detected by region growing, where the growing criterion is that the candidate point must be similar to the seed point normal. For every locally smooth segment, the proposed algorithm tries to find the geometric primitive that best explains the underlying point set. RANSAC (Fischler and Bolles, 1981) is used to find the best sphere and cylinder model. The search for sphere and cylinder model is performed only if the computed planar model does not fully explain the segment.

In **(Rabbani, van den Heuvel and Vosselman, 2006)**, a method for segmentation of point clouds to smooth surfaces is proposed. The method relies on the residual in the plane fitting as a measure for surface curvature. The plane is fitted to a local neighborhood of every point in the input point cloud and the residual w.r.t. this plane is assigned to each point. The point cloud is then segmented by a region growing procedure. This procedure starts by selecting the point with the minimum residual as the current seed. The region is grown by adding the neighboring points of the current seed which satisfy the smoothness condition and whose residuals are less than a predefined threshold. The smoothness condition is that the angle between the normal of the candidate point and the neighboring seed point is smaller than a predefined threshold.

In **(Rusu et al., 2009)**, a method for point cloud segmentation into primitives and scene completion is proposed. It is assumed that the direction of the gravity axis is known. The subset of the input point cloud containing points whose normals are approximatively parallel with the gravity axis, is segmented using Euclidean clustering. For every cluster, a sample consensus estimator is used to find the best planar model. For every set of point inliers corresponding to the planar model, a bounding polygon is computed representing the estimated table bounds. All Euclidean point clusters supported by the table model (i.e. sitting on the table, and within the polygonal bounds) are extracted. For each segmented point cluster search for 3D primitive geometric surfaces is performed using Randomized M-Estimator Sample Consensus – RMSAC (available within the Point Cloud Library (PCL)) is applied with a focus on 3D primitive geometric shapes such as planes, cylinders, spheres and cones. Each candidate primitive shape is scored according to the number of inliers. The efficient inlier counting is performed by using octree structure representing the input point cloud. A branch of the octree is represented by a box and efficient methods for detection of box-plane, box-cylinder, box-sphere and box-cone intersection are applied. If a box representing a octree branch does not intersect with the candidate primitive, all points contained in this box are excluded from the further inlier search. For the best primitive shape candidate, non-linear optimization of the shape parameters is performed using the Levenberg-Marquard algorithm. The geometric shape coefficients are then used to reconstruct missing data. Residual points are resampled and

triangulated, to create smooth decoupled surfaces that can be manipulated. Finally, the resultant hybrid shape-modelis built by concatenating the shape model with the surface model.

In **(Lakani et al., 2014)**, a method for segmenting of 3D objects into primitives: cone, cylinder, cube and sphere is presented. The primitives are detected in the low-curvature regions in the object. The proposed algorithm has two steps. In the first step, low-curvature regions are detected. Then, in each region, primitive shapes are estimated. The input point cloud is oversegmented into supervoxels using the method proposed in (Papon et al., 2013). The local surface curvature shape is identified by HK algorithm, which categorizes the surface shape into five categories; flat, convex elliptic, concave elliptic, convex cylindrical, concave cylindrical. This algorithm is applied on the estimated curvature of border voxels between two adjacent supervoxels. In the case that the estimated surface shape based on two adjacent supervoxels is non-flat, it is considered as disconnected. From the low-curvature regions obtained as above, the likelihood of a specific region given each particular primitive shape is estimated. The likelihood is estimated according to the similarity between the point normals and the corresponding primitive normals.

3. Oversegmentation as a Preprocessing Step

Many segmentation algorithm perform oversegmentation to superpixels or supervoxels as a preprocessing step (Gupta, Arbeláez and Malik, 2013). The introduction of a low-level preprocessing step to oversegment images into superpixels – relatively small regions whose boundaries agree with those of the semantic entities in the scene – has enabled advances in segmentation by reducing the number of elements to be labeled from hundreds of thousands, or millions, to a just few hundred (Simari, Picciau and De Floriani, 2014).

In **(Papon et al., 2013)**, a method for segmentation of point clouds into supervoxels is proposed. The method is named Voxel Cloud Connectivity Segmentation (VCCS). The method is based on k-means clustering, which is the idea originally used in SLIC (Achanta et al., 2012) – one of the most popular superpixel methods for RGB images. The space of the input point cloud is represented by voxels of size R_{voxel}. The algorithm begins by selecting a number of seed points which will be used to initialize the supervoxels. In order to do this, the space is divided into a voxelized grid, where the size R_{seed} of the cells in this grid is significantly larger than R_{voxel}. Initial candidates for seeding are chosen by selecting the voxel in the cloud nearest to the center of each occupied seeding voxel. The seeds are shifted to the connected voxel within the search volume which has the smallest gradient in the search volume, where the gradient is computed as the average absolute difference between the query voxel and the neighboring voxels in CIELAB color space. Once the seed voxels have been selected, the supervoxel feature vector is initialized by finding the center of the seed voxel in feature space and connected neighbors within 2 voxels. VCCS supervoxels are then detected as clusters in a 39 dimensional feature space, where feature vector is composed of the following elements: spatial coordinates *x*, *y*, and *z*, color in CIELab space and 33 elements of FPFH (See ARP3D TR2). The distance in this feature space is defined by

$$D = \sqrt{\frac{\lambda D_c^2}{m^2} + \frac{\mu D_s^2}{3R_{seed}^2} + \varepsilon D_f^2}$$

where D_s is the Euclidean distance in the point cloud 3D space, D_c is the Euclidean distance in CIELab space D_f is calculated using the Histogram Intersection Kernel and m is a normalization constant. Constants λ, μ and ε control the influence of color, spatial distance and geometric similarity, respectively, in the clustering. The clustering is performed using a local k-means clustering, as proposed in (Achanta et al., 2012), with the significant difference that connectivity and flow are considered when assigning pixels to a cluster. The process is as follows: beginning at the voxel nearest the cluster center, for all its adjacent voxels the distance D to the supervoxel center is computed. If the distance is the smallest this voxel has seen, its label is set and its neighbors which are further from the center are added to a search queue for this label. Then the next supervoxel is processed in the same way, so that each level outwards from the center is considered at the same time for all supervoxels. The process continues iteratively outwards until the edge of the search volume is reached for each supervoxel, or there are no more neighbors to check. Once the search of all supervoxel adjacency graphs has concluded, the centers of each supervoxel cluster are updated by taking the mean of all its constituents. This is done iteratively; either until the cluster centers stabilize, or for a fixed number of iterations. The source code of the proposed method is made available by the authors within PCL.

A similar approach is proposed in **(Yang et al., 2015)**. This approach is a more direct adaptation of the SLIC algorithm for RGB-D images. In contrast to the algorithm proposed in (Papon et al., 2013), which can be applied to general point clouds, the approach proposed in (Yang et al., 2015) requires organized point cloud i.e. RGB-D images. Nothing substantially new regarding superpixel detection is achieved in comparison to (Papon et al., 2013).

In **(Simari, Picciau and De Floriani, 2014)**, an oversegmentation algorithm for triangular (not colored) meshes is proposed. The segments this algorithm produces are called by the authors superfacets. Analogously to other k-means style algorithms, like SLIC, the proposed algorithm can be subdivided into three high-level steps: (i) initialization, (ii) update of segment centers and (iii) classification of triangles, where steps (ii) and (iii) are alternately repeated until convergence. Three initialization methods are proposed. The first is an iterative farthest point strategy. The first region center is placed at the triangle whose centroid is closest to the centroid of the whole mesh. Then, each subsequent center is added at the triangle with maximum Euclidean distance to the nearest already placed center. Another initialization method is the following: starting from the triangle closest to the mesh centroid the classification step is performed until the triangle being processed is at a distance from the initial seed greater than a user defined parameters. When this distance is exceeded, this triangle is used to begin a new region. The third initialization method is to subdivide the 3D space in which the mesh is embedded into a regular 3D grid based on the desired radius. When all the triangles have been assigned to some superfacet, the position of each superfacet center is computed as follows. For each superfacet, the Euclidean area-weighted mean of all triangle centroids belonging to that superfacet is computed and then the new center is designated to be the triangle closest to said mean. If no center has changed, the algorithm terminates. Otherwise, the classification step is performed. The classification step proceeds as follows. For each triangle, its shortest-path distance along the face graph of the mesh to the nearest superfacet center is computed. The distance metric used in this process is a weighted sum of an approximate geodesic distance and the normalized dihedral angle at the shared edge of two neighboring triangles weighted by a factor which increases the distance in the case of concavity.

An adaptation of the SLIC algorithm to 3D scalar fields defined over the vertices of a tetrahedral mesh considering tetrahedra rather than pixels is proposed in **(Picciau et al., 2015)**. The source code of the proposed method is made publically available by the authors.

The approach proposed in **(Stückler and Behnke, 2014)** represents a point cloud, obtained by registration of multiple colored point clouds, by a multi-resolution surfel map. This map is based on octree. In each node of the tree across all resolutions (voxel sizes), statistics on the joint spatial and color distribution of the points within its volume is stored. This data related to each octree node is named a surfel. The distribution is approximated with sample mean and covariance of the data, i.e., the data is modeled as normally distributed in a node's volume. Since the method is intended for building of maps of scenes and objects from all perspectives, multiple distinct surfaces may be contained within a node's volume. Hence, multiple surfels are maintained in a node that are visible from several view directions. Up to six orthogonal view directions aligned with the axes of the map reference frame are considered. When adding a new point to the map, the view direction onto the point is determined and associated with the surfel belonging to the most similar view direction. The source code of the proposed method is made publically available by the authors at https://code.google.com/archive/p/mrsmap/.

4. Semantic Segmentation

Segmenting of RGB-D images into regions classified in one of the following four structural classes:

- ground,
- permanent structures (such as walls, ceilings and columns),
- large furniture (such as tables, dressers, and counters) and
- props (easily movable objects).

is proposed in **(Silberman et al., 2012)**, where an approach based on identification of support relationships is presented. The classification process starts by aligning point cloud to principal directions (vanishing points) detected using straight line segments in the image. Then, potential wall, floor, support, and ceiling planes are detected using a RANSAC procedure. To determine which image pixels correspond to each plane, the graph cuts segmentation with alpha expansion is applied. An initial segmentation is performed by the watershed algorithm applied to Pb boundaries, proposed in (Arbeleaz, 2006). This oversegmentation is forced to be consistent with the 3D plane regions previously detected using RANSAC, which primarily helps to avoid regions that span wall boundaries with faint intensity edges. The segments obtained by oversegmentation are then segmented into objects and dominant surfaces by a hierarchical segmentation algorithm proposed in (Hoiem, Efros and Hebert, 2011). Regions with minimum boundary strength are iteratively merged until the minimum cost reaches a given threshold. Boundary strengths are predicted by a trained boosted decision tree classifier which uses RGB and 3D features. These proposed 3D features encode regions corresponding to different planes or having different surface orientations or depth differences are likely to belong to different objects. Given an image split into regions representing objects or surfaces, the support relationship between these regions are determined. The basic assumption made the proposed model is that every region is either (a) supported by a region visible in the image plane, (b) supported by an object not visible in the image plane or (c) requires no support indicating that the region is the ground itself. Two types of support relations between two

regions are considered: (i) one region is supported by the other from below or (ii) the region is supported by the other region from behind. The support relations are determined together with the membership of regions to one of the four aforementioned structural classes are determined simultaneously by a maximum a posteriori (MAP) process, where likelihood that one region supports another and the likelihood that a particular region belongs to a particular structural class are computed by two logistic regression classifiers. The source code of the proposed method is made publically available by the authors at http://cs.nyu.edu/~silberman/code.html.

In **(Gupta, Arbeláez and Malik, 2013)**, the problem of semantic segmentation into ground, structure, furniture and props is also considered. The segments obtained by the approach described in Section 1 are classified to the aforementioned classes by a SVM and a random forest classifier. The features which are used as classification cues are the angle with respect to gravity, absolute orientation in space, fraction of a segment that is vertical, fraction of the segment that is horizontal, the minimum and maximum height above ground, mean and median height of the horizontal part of the segment, the size of the 3D bounding rectangle, the surface area - total area, vertical area, horizontal area facing up, horizontal area facing down, if the segment is clipped by the image and what fraction of the convex hull is occluded, planarity of the segment (estimated by the error in the plane fitting), average strength of local geometric gradients inside the region, on the boundary of the region and outside the region, average orientation of patches in the regions around the segment. In addition to these generic features, the following category specific features are also used: histograms of vector quantized color SIFT (van de Sande, Gevers and Snoek, 2010) as the appearance features, histograms of geocentric textons (vector quantized words in the joint 2-dimensional space of height from the ground and local angle with the gravity direction) as shape features.

The same segmentation problem is addressed in **(Cadena and Kosecka, 2013; Cadena and Kosecka, 2015)**, where a solution based on conditional random fields and minimum spanning tree is proposed. The proposed approach is evaluated using NYUD2 dataset (see Section 7).

In **(Schulz, Nico and Behnke, 2015)** the problem of semantic segmentation into the four classes introduced by (Silberman et al., 2012) using CNN is proposed. The problem with missing scale invariance in CNN is addressed. The neural network is trained on image patches with a size chosen proportional to the depth of the patch center. Since training is scale invariant, smaller models can be used and the training data can be used more efficiently. A sampling scheme which covers the image with overlapping patches of depth-adjusted size is proposed. Thus, closer image regions are processed at a large scale, while far-away regions are processed at a small scale. This automatic adjustment is more efficient than sliding windows, because the scale is chosen automatically. In contrast to a multi-scale sliding window, the scale adjustments in the proposed approach are continuous. Height above ground is used as an additional input to the CNN. The proposed approach is evaluated using NYUD2 dataset (see Section 7).

5. Segmentation into Parts

In **(Schoeler, Papon and Wörgötter, 2015)**, a bottom-up method for segmenting 3D point clouds into functional parts is proposed. The method uses local concavities as an indicator for inter-part boundaries. The algorithm begins by oversegmenting the input point cloud into supervoxels using

VCCS (Papon et al., 2013) and creating supervoxel adjacency graph, whose nodes correspond to supervoxels and edges represent neighboring relations between supervoxels. The classification proposed for the LCCP-algorithm (Stein et al., 2014) is used to label edges in the graph as either convex or concave. The adjacency graph is converted into a Euclidean Edge Cloud (EEC), where each point represents an edge in the adjacency graph. The EEC is then partitioned by a greedy procedure, starting from the entire point cloud, representing the initial segment, which is then recursively segmented by partitioning each segment using the locally constrained directional weighted RANSAC. This RANSAC-based procedure performs random sampling of an EEC segment searching for the optimal partitioning plane, which maximizes the following score

$$S_m^n = \frac{1}{\left|P_m^n\right|} \sum_{i \in P_m^n} \omega_i t_i \ ,$$

where P_m^n denotes a cluster of points lying on the partitioning plane within a particular tolerance, $|\cdot|$ denotes the cardinality of a set and ω_i and t_i are weights of the point i, which are computed as explained in the following. The weight ω_i is computed by

$$\omega(\alpha) = \mathcal{H}(\alpha - \beta_{thresh}) \ ,$$

where α is the angle between the normals of the neighboring supervoxels connected by the edge represented by the point i in ECC, β_{thresh} is a user defined parameter set to 10° and $\mathcal{H}$ is the Heaviside step function. The weight t_i is computed by

$$t_i = \begin{cases} \left|d_i^T s_m\right| & \text{if } i \text{ is concave} \\ 1 & \text{if } i \text{ is convex} \end{cases}$$

where d_i is the unit vector connecting the centroid of the neighboring supervoxels and s_m is the normal of the partitioning plane. Cluster P_m^n is created by Euclidean clustering of all points lying on the partitioning plane within a user defined tolerance. This whole cutting procedure is repeated recursively on the newly generated segments and terminates if no cuts can be found which exceed the minimum score S_{min} or if the segment consists of less than N_{min} supervoxels. The proposed approach is evaluated using the Princeton Object Segmentation Benchmark (Chen, Golovinskiy and Funkhouser, 2009). The method's source code will be freely distrib1uted as part of the Point Cloud Library (PCL)

The segmentation method proposed in **(Stein et al., 2014)**, described in Section 1, is applied for segmentation of objects into body and handle. The handle detection is useful for robotic grasping.

6. Performance Measures

Four performance measures for determining the quality of (2D) segmentation results are mentioned in (Arbelaez et al., 2011): Variation of Information (*VI*); Probabilistic Rand Index (*PRI*), Segmentation Covering (*C*) and Precision-Recall on Boundaries.

1. **Variation of Information** (Meilă, 2005)

It measures the distance between two segmentations in terms of the information difference between them.

$$VI(S,S') = H(S) + H(S') - 2I(S,S') \quad (1)$$

where H and I represent respectively the entropies and mutual information between the two data clusterings S and S' (or test and ground truth segmentations). S and S' represent cluster sets of the same data set D, where $S = \{S_1, S_2, S_3, \ldots S_C\}$ with cluster sizes n_k and $S' = \{S'_1, S'_2, S'_3 \ldots S'_{C'}\}$ with cluster sizes $n'_{k'}$. The total number of points in D is n_{tot}. H and I are defined by:

$$H(S) = -\sum_{k=1}^{C} \frac{n_k}{n_{tot}} \cdot \log\left(\frac{n_k}{n_{tot}}\right) \quad (2)$$

$$I(S,S') = \sum_{k=1}^{C}\sum_{k'=1}^{C'} \frac{n_{k,k'}}{n_{tot}} \cdot \log\left(\frac{n_{k,k'}}{n_{tot}}\frac{n_k}{n_{tot}}\frac{n_{k'}}{n_{tot}}\right) \quad (3)$$

where $n_{k,k'}$ represents the number of points in the intersection of clusters S_k of S and $S'_{k'}$ of S'

$$n_{k,k'} = |S_k \cap S'_k| \quad (4)$$

The perceptual meaning and applicability of *VI* in the presence of several ground-truth segmentations is unclear.

2. Rand Index (Rand, 1971)

The Rand index compares the compatibility of assignments between pairs of elements in the clusters. The Rand Index between test and ground truth segmentations S and G is given by the sum of the number of pairs of pixels that have the same label in S and G and those that have different labels in both segmentations, divided by the total number of pairs of pixels (Rand, 1971; Unnikrishnan and Hebert, 2005). This gives a measure of similarity with value ranging from 0 when the two segmentations have no similarities (i.e. when one consists of a single cluster and the other consists only of clusters containing single points) to 1 when the segmentations are identical.

Variants of the Rand Index have been proposed for dealing with the case of multiple ground-truth segmentations (Pantofaru and Hebert, 2005; Unnikrishnan and Hebert, 2005; Unnikrishnan et al., 2007)

Given a set of ground-truth segmentations $\{G_k\}$, the Probabilistic Rand Index is defined as:

$$PRI(S,\{G_k\}) = \frac{1}{T}\sum_{i<j}\left[c_{ij}p_{ij} + \left(1 - c_{ij}\right)\left(1 - p_{ij}\right)\right] \quad (5)$$

where c_{ij} is the event that pixels i and j have the same label and p_{ij} its probability. T is the total number of pixel pairs. Using the sample mean to estimate p_{ij}, *PRI* amounts to averaging the Rand Index among different ground-truth segmentations. The *PRI* has been reported to suffer from a small dynamic range (Pantofaru, and Hebert, 2005; Unnikrishnan et al., 2007) and it often has similar values across images and algorithms. The *PRI* index is on a scale of 0 to 1, but there is no expected value for a given segmentation. The *PRI* score may be compared to the score of another segmentation of the same image, but there is no information as to whether or not the difference between the two scores is relevant or not. In (Pantofaru, and Hebert, 2005; Unnikrishnan et al., 2007), this drawback is addressed by normalization with an empirical estimation of its expected value (ie. Normalized *PRI*).

3. Segmentation Covering

In order to evaluate the pixel-wise classification task in recognition, the overlap between two regions *R* and *R*' is used (Everingham et al., 2005; Everingham et al., 2015; Malisiewicz and Efros, 2007) and is defined as:

$$O(R, R') = \frac{|R \cap R'|}{|R \cup R'|} \tag{6}$$

The *covering* of a segmentation *S* by a segmentation *S'* is defined as (Arbelaez et al. 2011):

$$C(S' \to S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} O(R, R') \tag{7}$$

where *N* is the total number of pixels in the image. In a similar manner, the covering of a segmentation *S* by a family of ground-truth segmentations $\{G_i\}$ is defined by first covering *S* separately with each ground-truth G_i, and then averaging over the different ground truths.

4. Precision-Recall on Boundaries (Martin et al. 2004)

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

where

- *P* – precision,
- *R* – recall,
- *TP* – true positives,
- *FP* – false positives,
- *FN* – false negatives.

In probabilistic terms, precision is the probability that the detector's signal is valid, and recall is the probability that the ground truth data was detected. The precision and recall measures can be particularly meaningful in the context of boundary detection when applications that make use of boundary maps, such as stereo or object recognition, are considered. It is reasonable to characterize higher level processing in terms of how much true signal is required to succeed *R* (recall), and how much noise can be tolerated *P* (precision). A particular application can define a relative cost α between these quantities, which focuses attention at a specific point on the precision-recall curve.

The *F-measure* defined as

$$F = \frac{P \cdot R}{(\alpha R + (1-\alpha) P)} \tag{10}$$

captures this trade off as the weighted harmonic mean of *P* and *R*. The location of the maximum *F-measure* along the curve provides the optimal detector threshold for the application given α, which is set to 0.5 in their experiments.

If a set of ground truth boundary maps exist, the obtained boundary map is compared separately with each ground truth map in turn. Only the pixels in the obtained boundary map that match no ground truth boundary are counted as false positives. The hit rate is simply averaged over the

different ground truth maps. Hence, to achieve perfect recall the obtained boundary map must explain all of the ground truth data. In this way, estimating precision and recall matches as closely as possible the intuitions one would have if scoring the outputs visually.

The precision-recall curve is a rich descriptor of performance. If a single performance measure or summary score is required, two measures can be used: (i) precision and recall can be combined with the *F-measure* to provide the best/maximal/global *F-measure* (with $\alpha = 0.5$); (ii) the average precision (AP) on the full recall range (equivalent to the area under the *P- R* curve).

In addition to the aforementioned four measures, **over-segmentation** and **under-segmentation score** are proposed in (Richtsfeld et al., 2012).

7. Benchmark Datasets

Probably the most popular benchmark dataset for RGB-D image segmentation is **NYUD2** dataset presented in (Silberman et al., 2012). The dataset is comprised of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. It features:

- 1449 densely labeled pairs of aligned RGB and depth images (hand selected from 435 103 video frames, to ensure diverse scene content and lack of similarity to other frames), gathered from a wide range of commercial and residential buildings in three different US cities;
- 464 different indoor scenes across 26 scene classes;
- 35 064 distinct objects, spanning 894 different classes;
- Each object is labeled with a class and an instance number (cup1, cup2, cup3, etc);
- For each of the 1449 images, support annotations were manually added. Each image's support annotations consists of a set of 3-tuples: [R_i, R_j, type] where R_i is the region ID of the supported object, R_j is the region ID of the supporting object and type indicates whether the support is from below (e.g. cup on a table) or from behind (e.g. picture on a wall);
- 407 024 new unlabeled frames.

The dataset is publically available at http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html.

Recently, a larger dataset Scene Understanding Benchmark Suite – **SUN RGB-D** (Song et al., 2015) is made available to the scientific community. It consists of:

- 1449 images from the NYUD2;
- 554 manually selected realistic scene images from the Berkeley B3DO Dataset (Janoch et al., 2011) (the images were captured using Kinect v1);
- manually selected 3,389 distinguished frames without significant motion blur from the SUN3D videos (Xiao et al., 2013) (the images were captured using Asus Xtion);
- 3784 images captured using Kinect v2;
- 1159 images captured using Intel RealSense.

The whole dataset is densely annotated and includes 146,617 2D polygons and 64,595 3D bounding boxes with accurate object orientations, as well as a 3D room layout and scene category for each image. In total, there are 47 scene categories and about 800 object categories.

Since the data is obtained using different sensors, a proposed robust algorithm for depth map integration from multiple RGB-D frames is used to improve the depth maps.

Any of the 6 following tasks can be tested using the benchmark:

- Scene Categorization
- Semantic Segmentation
- Object Detection (2D and 3D)
- Object Orientation
- Room Layout Estimation

The dataset is publically available at http://rgbd.cs.princeton.edu/.

Berkeley 3-D Object Dataset – **B3DO** (Janoch et al., 2011) is publically available at http://kinectdata.com/. The size of the dataset is not fixed and will continue growing with crowd-sourced submissions. The first release of the dataset contains:

- 849 images taken in 75 different scenes;
- over 50 different object classes are represented.

SUN3D database (Xiao et al., 2013) is a large-scale RGB-D video database with camera poses and object labels, capturing the full 3D extent of many places. It consists of RGB-D images, camera poses, object segmentations, and point clouds registered into a global coordinate frame. It is publically available at http://sun3d.cs.princeton.edu/.

The Object Segmentation Database – **OSD** (Richtsfeld et al., 2012) provides 111 RGBD data in 6 subcategories to enable evaluation of object segmentation approaches. It is publically available at http://users.acin.tuwien.ac.at/arichtsfeld/?site=4.

Willow Garage provides a RGB-D dataset for an Object Recognition Challenge. The dataset consists mainly of simple freestanding objects for evaluation of object segmentation approaches as well as parametric surface fitting and provide exact ground truth for all objects in this dataset. It is publically available at http://www.acin.tuwien.ac.at/forschung/v4r/software-tools/willow/.

RGB-D Object Dataset (Lai et al., 2011) is a large-scale, hierarchical multi-view object dataset collected using an RGB-D camera. The dataset contains 300 objects organized into 51 categories. It is publically available at http://rgbd-dataset.cs.washington.edu/dataset.html.

Cornell-RGBD-Dataset has 24 labeled office scene point clouds and 28 labeled home scene point clouds. It is publically available at http://pr.cs.cornell.edu/sceneunderstanding/data/data.php.

In **(Chen, Golovinskiy and Funkhouser, 2009)**, a benchmark dataset for segmentation of 3D meshes into parts, named Princeton Object Segmentation Benchmark, is presented.

References

Achanta R, Shaji A, Smith K, Lucchi A, Fua P and Süsstrunk S (2012) SLIC Superpixels Compared to State-of-the-Art Superpixel Methods, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281

Arbelaez P (2006) Boundary extraction in natural images using ultrametric contour maps, *IEEE Workshop on Perceptual Organization in Computer Vision (POCV)*, New York, USA.

Arbeláez P, Maire M, Fowlkes C and Malik J (2011) Contour detection and hierarchical image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*

Attene M, Falcidieno B and Spagnuolo M (2006) Hierarchical mesh segmentation based on fitting primitives, The Visual Computer, vol. 22, no. 3, pp. 181-193.

Attene M, Mortara M, Spagnuolo M and Falcidieno B (2008) Hierarchical Convex Approximation of 3D Shapes for Fast Region Selection, *Computer Graphics Forum*, vol. 27, no. 5, pp. 1323–1332

Beale D, Iravani P and Hall P (2011) Probabilistic Models for Robot-Based Object Segmentation, *Robotics and Autonomous Systems*, vol. 59, issue 12, pp. 1080–1089.

Besl P and McKay N (1992) A Method for Registration of 3-D Shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2.

Boykov YY and Jolly MP (2001) Interactive graph cuts for optimal boundary & region segmentation of objects in ND images, *International Conference on Computer Vision (ICCV)*, pp. 105–112.

Cadena C and Kosecka J (2013) Semantic Parsing for Priming Object Detection in RGB-D Scenes, *Semantic Perception Mapping and Exploration (SPME)*, Karlsruhe, Germany.

Cadena C and Kosecka J (2015) Semantic Parsing for Priming Object Detection in Indoors RGB-D Scenes, *International Journal on Robotics Research*, vol. 34, no. 4-5, pp. 582–597.

Chen X, Golovinskiy A and Funkhouser T (2009) A Benchmark for 3D Mesh Segmentation, *ACM Transactions on Graphics (SIGGRAPH)*, vol. 28, no. 3

Cupec R, Nyarko EK and Filko D (2011) Fast 2.5D Mesh Segmentation to Approximately Convex Surfaces, *Proceedings of the European Conference on Mobile Robots (ECMR)*, Örebro, Sweden pp. 127–132

Dollár P and Zitnick CL (2013) Structured forests for fast edge detection. *International Conference on Computer Vision (ICCV)*, pp. 1841–1848.

Everingham M, Eslami SMA, van Gool L, Williams CKI, Winn J and Zisserman A (2005) The Pascal Visual Object Classes Challenge, http://host.robots.ox.ac.uk/pascal/VOC/index.html.

Everingham M, Eslami SMA, van Gool L, Williams CKI, Winn J and Zisserman A (2015) The Pascal Visual Object Classes Challenge: A Retrospective, *International Journal of Computer Vision (IJCV)*, vol. 111, no. 1, pp. 98–136.

Felzenszwalb PF and Huttenlocher DP (2004) Efficient graph-based image segmentation, *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181.

Fischler MA and Bolles RC (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Graphics and Image Processing, vol. 24, no. 6, pp. 381–395.

Garland M, Willmott A and Heckbert PS (2001) Hierarchical Face Clustering on Polygonal Surfaces, *ACM Symposium on Interactive 3D Graphics*.

Girshick R, Donahue J, Darrell T and Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587.

Gupta S, Arbeláez P, and Malik J (2013) Perceptual Organization and Recognition of Indoor Scenes from RGB-D Images, *Computer Vision and Pattern Recognition (CVPR)*

Gupta S, Girshick R, Arbeláez P and Malik J (2014) Learning Rich Features from RGB-D Images for Object Detection and Segmentation, *Lecture Notes Computeer Science*, vol. 8695, pp. 345–360.

Hickson S, Birchfield S, Essa I and Christensen H (2014) Efficient hierarchical graph-based segmentation of RGBD videos, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 344–351.

Hoiem D, Efros AA and Hebert M (2011) Recovering occlusion boundaries from an image, *International Journal of Computer Vision*, vol. 91, pp. 328–346.

Holz D and Behnke S (2014) Approximate Triangulation and Region Growing for Efficient Segmentation and Smoothing of Range Images, *Robotics and Autonomous Systems*, vol. 62, no. 9, pp. 1282–1293

Janoch A, Karayev S, Yangqing J, Barron,JT, Fritz,M, Saenko,K and Darrell T (2011) A category-level 3-D object dataset: Putting the Kinect to work, *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1168–1174.

Karpathy A, Miller S and Fei-Fei L (2013) Object discovery in 3d scenes via shape analysis, *IEEE International Conference on Robotics and Automation (ICRA)*

Koppula H, Anand A, Joachims T and Saxena A (2011) Semantic labeling of 3d point clouds for indoor scenes, *Annual Conference on Neural Information Processing Systems (NIPS)*

Krizhevsky A, Sutskever I and Hinton GE (2012) ImageNet classification with deep convolutional neural networks. *Annual Conference on Neural Information Processing Systems (NIPS)*

Lai K, Bo L, Ren X and Fox D (2011) A large-scale hierarchical multi-view RGB-D object dataset, *IEEE International Conference on Robotics and Automation (ICRA)*

Lakani SR, Popa M, Rodríguez-Sánchez A and Piater J (2014) Scale-Invariant, Unsupervised Part Decomposition of 3D Objects, *Parts and Attributes, (Workshop at ECCV)*

Lakani SR, Popa M, Rodríguez-Sánchez A and Piater J (2015) CPS: 3D Compositional Part Segmentation through Grasping, *Conference on Computer and Robot Vision*, pp. 117–124

Lowe DG (2004) Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110.

Maji S, Berg AC and Malik J (2013) Efficient classification for additive kernel SVMs, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 35, no. 1, pp. 66–77.

Malisiewicz T and Efros AA (2007) Improving Spatial Support for Objects via Multiple Segmentations, *British Machine Vision Conference (BMVC)*, pp. 55.1–55.10.

Martin DR, Fowlkes CC and Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 26, no. 5, pp. 530–549.

Meilă, M (2005) Comparing clusterings - An Axiomatic View, *International Conference on Machine learning (ICML)*, pp. 577–584.

Mishra AK and Aloimonos Y (2011) Visual segmentation of simple objects for robots, *Robotics: Science and Systems*, VII, pp. 1–8

Mishra AK, Aloimonos Y and Fah CL (2009) Active Segmentation With Fixation, *International Conference on Computer Vision (ICCV)*

Mishra AK, Shrivastava A and Aloimonos Y (2012) Segmenting "Simple" Objects Using RGB-D, *IEEE International Conference on Robotics and Automation (ICRA)*

Mitra NJ, Guibas LJ, and Pauly M (2006) Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 560–568

Pajarinen J and Kyrki V (2015) Decision Making Under Uncertain Segmentations. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1303–1309.

Pantofaru C and Hebert M (2005) A Comparison of Image Segmentation Algorithms, *Technical Report, Robotics Institute, Carnegie Mellon University*.

Papon J, Abramov A, Schoeler M and Wörgötter F (2013) Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2027–2034.

Picciau G, Simari P, Iuricich F and De Floriani L (2015) Supertetras: A Superpixel Analog for Tetrahedral Mesh Segmentation, *International Conference on Image Analysis and Processing (ICIAP)*, Genoa, Italy

Rabbani T, van den Heuvel F and Vosselman G (2006) Segmentation of point clouds using smoothness constraint, *Proceedings of the ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, Dresden, Germany, pp. 248–253.

Rand WM. (1971) Objective Criteria for the Evaluation of Clustering Methods, *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850.

Reisner-Kollmann I and Maierhofer S (2012) Segmenting multiple range images with primitive shapes, *Proceedings of 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp.320-323

Richtsfeld A, Mörwald T, Prankl J, Zillich M, and Vincze M. (2014) Learning of perceptual grouping for object segmentation on rgb-d data. *Journal of Visual Communication and Image Representation*. vol. 25, pp. 64–73.

Richtsfeld A, Mörwald T, Prankl J, Zillich M, and Vincze M (2012) Segmentation of unknown objects in indoor environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4791–4796.

Rusu RB, Blodow N, Marton ZC and Beetz M (2009) Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp 1–6.

van de Sande KEA, Gevers T and Snoek CGM (2010) Evaluating color descriptors for object and scene recognition, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 32, issue 9, pp. 1582–1596.

Schmitt F and Chen X (1991) Fast segmentation of range images into planar regions, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 710–711.

Schnabel R., Wahl R and Klein R (2007) Efficient RANSAC for point-cloudshape detection. *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226

Schoeler M, Papon J and Wörgötter F (2015) Constrained Planar Cuts - Object Partitioning for Point Clouds, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5207–5215.

Schulz H, Nico H and Behnke S (2015) Depth and Height Aware Semantic RGB-D Perception with Convolutional Neural Networks. European Symposium on Artificial Neural Networks (ESANN).

Silberman N, Hoiem D, Kohli P and Fergus R (2012) Indoor Segmentation and Support Inference from RGBD Images, *European Conference on Computer Vision (ECCV)*

Simari P, Picciau G and De Floriani (2014) Fast and scalable mesh superfacets. *Computer Graphics Forum*, vol. 33, no. 7, pp. 181–190.

Song S, Lichtenberg SP and Xiao J (2015), SUN RGB-D : A RGB-D Scene Understanding Benchmark Suite, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, no. Figure 1, pp. 567–576

Stein SC, Wörgötter F, Schoeler M, Papon J and Kulvicius T (2014) Convexity based object partitioning for robot applications, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3213–3220

Stückler J and Behnke S (2014) Multi-Resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking, *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 137–147

Ückermann A, Haschke R and Ritter H (2012) Real-time 3D segmentation of cluttered scenes for robot grasping, *IEEE-RAS International Conference on Humanoid Robots*

Unnikrishnan R and Hebert M (2005) Measures of similarity, *IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 394–400.

Unnikrishnan R, Pantofaru C and Hebert M (2007) Toward objective evaluation of image segmentation algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944.

Xiao J, Owens A and Torralba A (2013) SUN3D: A database of big spaces reconstructed using SfM and object labels, *IEEE International Conference on Computer Vision (ICCV)*, pp. 1625–1632.

Yang J, Gan Z, Li K and Hou C (2015) Graph-Based Segmentation for RGB-D Data Using 3-D Geometry Enhanced Superpixels, *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 913–926