

Shape Segmentation by Approximate Convexity Analysis

Oliver van Kaick, Noa Fish, Yanir Kleiman, Shmuel Asafi, and Daniel Cohen-Or
Tel Aviv University

We present a shape segmentation method for complete and incomplete shapes. The key idea is to directly optimize the decomposition based on a characterization of the expected geometry of a part in a shape. Rather than setting the number of parts in advance, we search for the smallest number of parts that admit the geometric characterization of the parts. The segmentation is based on an *intermediate-level* analysis, where first the shape is decomposed into approximate convex components, which are then merged into consistent parts based on a non-local geometric signature. Our method is designed to handle incomplete shapes, represented by point clouds. We show segmentation results on shapes acquired by a range scanner, and an analysis of the robustness of our method to missing regions. Moreover, our method yields results that are comparable to state-of-the-art techniques evaluated on complete shapes.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*.

General Terms: shape segmentation, incomplete shapes, point clouds, weakly-convex decomposition.

Additional Key Words and Phrases: missing data, part characterization.

1. INTRODUCTION

Segmentation of shapes into semantic parts is a well researched yet still challenging problem. With the recent technological advances and reduction in the cost of scanning technologies, scanning has become a prominent source for 3D shape acquisition. However, direct segmentation of point clouds has not received much attention by the research community. Instead, a common approach is the reconstruction of a mesh from a scanned point cloud [Schnabel et al. 2009; Li et al. 2011], followed by a state-of-the-art mesh segmentation technique. Segmenting a point cloud at an early stage of the analysis pipeline can greatly simplify the subsequent steps of analysis and reconstruction, as it provides cues on the semantic parts that form the shape (Figure 1).

Common mesh segmentation techniques usually avoid direct definitions of the *geometry* of a part, focusing instead on identifying the *boundaries* between parts. Prominent examples of state-of-the-art methods are segmentation by randomized cuts [Golovinskiy and Funkhouser 2008] and concavity-aware segmentation [Au et al. 2012]. Since these techniques and others [Shamir 2008] do not define an objective function for the parts, they typically search for concavities on the surfaces of triangle meshes. In this paper, we present an algorithm for the segmentation of point clouds and incomplete shapes, based on a definition of the geometry of a part. The algorithm is *non-parametric*; as in statistical non-parametric methods, the user is not required to provide in advance the number of segments into which the shape should be partitioned, although the part definition depends on a set of parameters.

Characterizing the expected geometry of a part is extremely ambitious since semantic parts may have complex geometry and varying features, and in general the meaning of a semantic part is ill-posed. Nevertheless, some intuition for discovering the shape of a

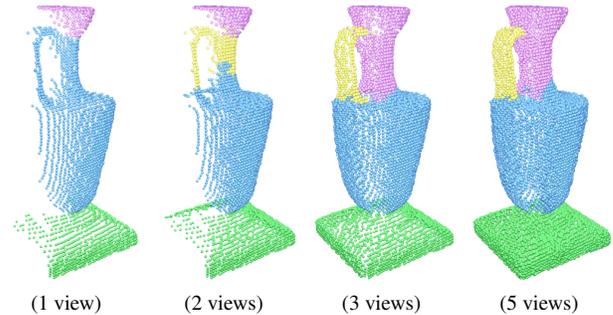


Fig. 1: Results of our segmentation algorithm applied to a progression of incomplete point clouds with an increasing number of scanned views. With only two views, the parts are already meaningful, despite the significant amount of missing data.

semantic part is given by the *minima rule*, which states that humans typically perceive a complex shape as a composition of approximately convex parts, since the part boundaries tend to lie along the concavities of the shape [Hoffman and Richards 1984]. We follow the minima rule by relying on *weakly convex* parts [Asafi et al. 2013], which are approximately convex but not necessarily strictly convex (Figure 2). Many shapes consist of semantic parts which are not necessarily convex, for example the handles of a vase or a curved stand of a lamp, as in Figure 3. However, in many of these cases, a semantic part may be composed of several weakly convex sub-parts which have similar properties. The premise is that identifying similarities between parts is much simpler when the parts are weakly convex, as some meaningful high level properties are revealed. See Figure 3 for an example that illustrates this intuition.

Our method therefore consists of two main steps (illustrated in Figure 2). First, we decompose the point cloud into weakly convex parts with a new method based on the definitions introduced in [Asafi et al. 2013]. This decomposition is robust even when handling point clouds of incomplete shapes. In the second step, we merge neighboring weakly convex parts with similar properties to produce the final segmentation of the shape. This step is likely to create segments that are closer to the semantic parts of the shapes. The main property that we consider when combining the components is the *volumetric profile* of the parts, which goes beyond local signatures by considering the volume of the shape parts.

These two steps result in an *intermediate-level* analysis of the shape, implying that we do not only consider low-level properties (such as boundaries) in the segmentation, but take into account an explicit characterization of the shape parts. We show that our method is robust when segmenting incomplete shapes, bypassing the need for non-trivial reconstruction to a complete shape, and yet, it yields results that are comparable to those of state-of-the-art *parametric* and *non-parametric* mesh segmentation techniques.

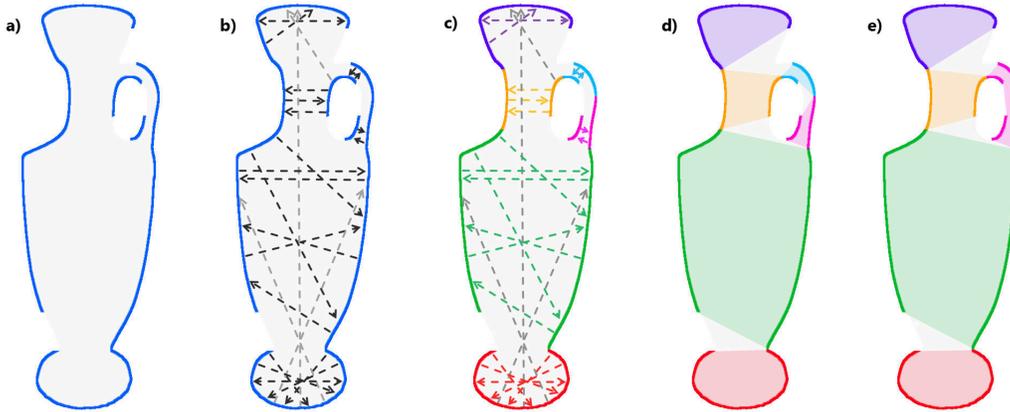


Fig. 2: Part characterization: (a) A point cloud with missing parts is given. (b) We compute the visibility between points in the shape. (c) The visibility information is used to decompose the shape into weakly-convex parts. Here, lines-of-sight within components are colored according to their component colors, while lines-of-sight between different components are grey. (d) The weakly-convex parts are analyzed to find adjacent parts with similar geometric properties. (e) Adjacent similar parts are merged to produce the final segmentation of the shape.

2. RELATED WORKS

Mesh segmentation. Much work has been done on mesh segmentation [Attene et al. 2006] which is surveyed in detail by Shamir [2008]; a set of representative algorithms is also compared in the benchmark of Chen et al. [2009]. Many of the proposed algorithms incorporate some form of the minima rule to segment the surface of models, such as the approaches based on randomized cuts [Golovinskiy and Funkhouser 2008], variational decomposition [Zhang et al. 2012] or concavity-aware fields [Au et al. 2012]. The randomized cuts algorithm was also applied to point clouds, but in the context of separating foreground from background, rather than segmenting an object into semantic components [Golovinskiy and Funkhouser 2009].

A few works also make direct use of volumetric information for segmentation, such as the shape diameter function [Shapira et al. 2008] or the part-aware metric [Liu et al. 2009]. In these works, the visibility between points or distance between points inside of the volume implies the separation between parts. In our work, the visibility between points also plays an important role, and we use it in a way that allows us to segment incomplete shapes. Lu et al. [2007] also present a method using volumetric information to compute a bounding volume of a shape. Their variational method resembles our convex decomposition step, however, unlike our method, they do not proceed beyond the decomposition towards a more semantic segmentation of the shapes.

The maturity of research on segmentation has culminated in the development of a benchmark which enabled the comparison of different segmentation algorithms [Chen et al. 2009]. The study concluded that the randomized cuts method [Golovinskiy and Funkhouser 2008] could be considered the state-of-the-art in *parametric* segmentation. More recently, Au et al. [2012] presented a *non-parametric* method based on concavity-aware fields, which outperformed the randomized cuts algorithm and is currently the state-of-the-art. We show comparable results to these methods.

Lastly, there has also been research on co-segmentation [Sidi et al. 2011; Hu et al. 2012; Meng et al. 2013] and joint segmentation [Huang et al. 2011] of sets of shapes, as well as on supervised segmentation [Simari et al. 2009; Kalogerakis et al. 2010]. These methods either require a training set of labeled shapes, user input,

or a set of shapes from the same category. Differently, we aim at an automatic segmentation of individual shapes, which is of importance when the category of the shapes is unknown in advance, cannot be clearly categorized or no training set is available.

Convex decomposition. An exact decomposition of a shape into convex components is costly and too strict for segmentation, since it can generate a large number of small parts. Thus, researchers have focused on obtaining approximately convex decompositions. Lien and Amato [2007] propose a greedy approach where concave portions of shape parts are identified and used to recursively split the parts into more convex components. Ren et al. [2011] optimize the criterion of Lien and Amato [2007] in order to minimize the number of parts generated. Kraevoy et al. [2007] propose a greedy region growing approach to obtain weakly convex components.

Moreover, Attene et al. [2008] represent a shape as a tetrahedral mesh and then combine the tetrahedra in a bottom-up manner to form the weakly convex parts. Differently from the other approaches, they explicitly consider the volume of the shape in the decomposition. Asafi et al. [2013] also consider the volume, but without the need of obtaining a tetrahedralization of the shape. In their work, weakly convex components are derived from analyzing the pairs of points in the shape that are visible to each other. We follow this general idea for the first step of our method, although we obtain the decomposition of the shapes with a new algorithm based on the analysis of patches. By using only visibility information, our method is able to handle incomplete models. Recently, Thiery et al. [2013] propose to approximate a mesh by a set of connected spheres that are linearly interpolated. Note that, although this method creates a good approximation of a mesh, it does not represent a decomposition of the original shape.

Point set analysis. 3D scans of real objects provide the quickest way to create large 3D repositories. Since scanning devices yield noisy and incomplete scans, a lot of attention has been directed at post-scan processing of the resulting 3D point set, namely, surface reconstruction. Different reconstruction methods take various approaches; we focus more on geometric primitive fitting as it is more closely related to our method. Gal et al. [2007] use

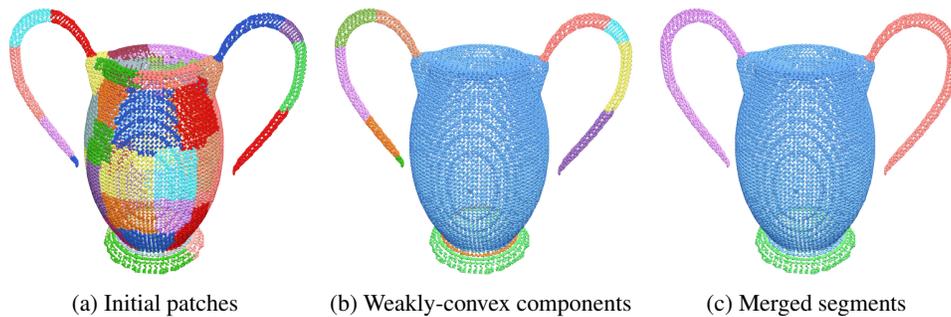


Fig. 3: Overview of our non-parametric segmentation: (a) We obtain a set of small, nearly-convex patches to better estimate the visibility among portions of the shape. (b) The patches are fused into larger weakly-convex components according to their visibility (e.g., the body). (c) Weakly-convex components with similar volumetric profiles are merged together (e.g., the handles) to yield the final segmentation.

a database of local shape priors from specifically-chosen context models and fit them to an incomplete input point set. Schnabel et al. [2009] detect a set of primitive shapes on the input point set itself and use that to seal off holes by continuing surrounding structures as well as create new edges as learned from the detected primitives' intersections. Attene and Patanè [2010] create a hierarchical organization of detected primitives to obtain a multi-resolution representation of a point set. Li et al. [2011] operate simultaneously on both the local and global aspects by fitting primitives locally according to observed data, while optimizing global relations iteratively. Recently, Lafarge et al. [2013] utilize planar primitive detection for point set structuring, where the primitives form the basis for a 3D Delaunay triangulation used to obtain the final surface. Instead of fitting priors or primitives to the point clouds, we decompose a shape into a set of weakly convex components, which form the starting point to build the segmentation of the shape.

Scene segmentation. Aside from reconstruction of single objects from multiple scans, another problem that has attracted attention recently is the acquisition of entire indoor scenes, motivated by the availability of inexpensive depth cameras such as *Kinect*. The main problem of interest in this domain has been segmentation of indoor scans, which consists of separating multiple objects that are part of the same point cloud. A popular approach to address this problem has been to learn how to discriminate objects based on training data, and then assign labels from a pre-defined set to the points of the scan. This has been accomplished by using graphical models [Angelov et al. 2005; Koppula et al. 2011; Shao et al. 2012], a multi-stage labeling procedure [Xiong et al. 2011] or single per-pixel classifiers [Shotton et al. 2011]. Other approaches include interleaving the segmentation and classification [Nan et al. 2012], or performing the learning from a set of initial scans [Kim et al. 2012]. Some of the recent approaches also fit a set of templates to the segmented scans to yield high-quality models of the scenes [Shao et al. 2012; Nan et al. 2012]. In contrast to these works, the focus of our method is segmentation of single shapes, although in Section 6 we show the potential of applying the method on scans of entire scenes. Moreover, we do not make use of training data, but rely on a characterization of the part geometry.

3. PART CHARACTERIZATION

In this section, we discuss our criterion for characterizing a shape part, which guides our segmentation algorithm. Given that our seg-

mentation method is designed to handle incomplete shapes, the input to our method is simply an oriented point cloud. Triangle meshes can also be handled as augmented point clouds. An illustration of our criterion is shown in Figure 2.

Shape part. We loosely define a part in the segmentation of a shape as a *collection of weakly-convex components that have similar geometric properties*.

First, according to the minima rule [Hoffman and Richards 1984], humans tend to perceive part boundaries at the concavities of shapes. Thus, by decomposing a shape into weakly-convex components, we obtain an initial decomposition of the shape where the component boundaries follow the minima rule closely. Note that we do not enforce a strict convexity of the parts, since this can generate many small and redundant components. Secondly, if many such weakly-convex components share similar geometric properties, the components are typically perceived as a single rather than separate entity, as shown in Figure 3(b) to (c). Identifying parts with similar properties is a relatively simpler task when all parts are weakly convex. Finally, starting from a set of weakly-convex components allows us to implicitly complete missing portions of the data by using the components as natural *priors*. That is a key aspect of the method for the successful segmentation of incomplete shapes.

Note that methods that explicitly search for segment boundaries, such as the randomized cuts [Golovinskiy and Funkhouser 2008] or concavity-aware field [Au et al. 2012] methods, cannot be trivially extended to handle shapes as the one shown in Figure 2. This shape is composed of two disconnected portions (left and right) that would have to be processed separately by such methods.

Next, we elaborate more on the notions of weak convexity and the geometric property that we consider when defining a part, namely, the *volumetric profile* of the part.

Weakly-convex parts. We follow the definition of weak convexity proposed in [Asafi et al. 2013], which is directly derived from the definition of *strict* convexity. Two points on a shape are said to be mutually visible or in a line-of-sight (LoS), if the straight line segment between the two points does not leave the volume of the shape. In a strict convex decomposition, all the points in a convex component are mutually visible. The definition is relaxed in [Asafi et al. 2013] so that in a weakly-convex component only a certain percentage of the points need to be mutually visible.

More specifically, let S be a set of sample points and $LoS(S)$ the set of all pairs of points in S that are in a line-of-sight. That

is, $(i, j) \in LoS(S)$ if and only if the points $i \in S$ and $j \in S$ are mutually visible. We then define the *convexity rank* of a set S as:

$$CR(S) = |LoS(S)| / |S|^2. \quad (1)$$

In a weakly-convex decomposition, we seek to minimize the number of components while maximizing the convexity rank for each component.

Note that this definition does not make any assumptions about the set S : the points are not necessarily taken from a watertight, complete, or connected surface. Additionally, there is no need to compute a convex hull or other geometric structures to estimate the convexity of the set, as opposed to alternative measures such as [Attene et al. 2008] or [Ren et al. 2011]. Thus, the convexity rank is directly applicable to the context of point clouds. To determine whether the line segment between two points does not leave the volume, we can in practice represent every point as a local disk, where the radius of the disk is estimated from the point's nearest neighbors and the orientation is given by its normal.

Volumetric profile. Generally, as exemplified in Figure 3(c), semantic shape parts maintain a consistent *volumetric profile*, even when they are not convex. That is, a part is unlikely to exhibit major changes in its overall silhouette across its volume. Following this observation, we characterize the volume of a weakly-convex component by a volumetric signature, and join two components if their signatures are similar. More specifically, given two weakly convex components C_i and C_j , with histograms of the shape diameter function [Shapira et al. 2008] h_i and h_j , respectively, we measure the volumetric dissimilarity between the parts using the Earth Mover's Distance (EMD):

$$dist(C_i, C_j) = EMD(h_i, h_j). \quad (2)$$

Two adjacent parts with high volumetric similarity are likely to belong to the same semantic shape part and can therefore be joined.

4. EXTRACTION OF WEAKLY-CONVEX COMPONENTS

Since decomposing a shape into convex components is an NP-hard problem, we follow a heuristic algorithm for the component extraction. Differently from the clustering method presented in [Asafi et al. 2013], we obtain the components with a two-step procedure, allowing us to better estimate the mutual visibility of points. We estimate the visibility between points more robustly by computing the visibility between a set of small, nearly-convex patches. Next, we use the patch visibility to derive larger weakly convex components. The key property of these components is that the patches that form the components should be *mutually visible*. An overview of the full algorithm is shown in Figure 3.

Initial patches. We begin by over-segmenting the input shape into a set of nearly-convex patches. The patches allow for a more robust visibility estimation as the visibility between all pairs of patches can be explicitly tested. The patches are extracted by constructing a *point-convexity graph* of the points in the shape, where point i and point j are connected if they are k -nearest neighbors in Euclidean space and the angle between their normals does not form a concavity. We use $k = 10$ in our implementation. We then perform a spectral clustering on this point-convexity graph to obtain the initial set of patches.

Next, we estimate the visibility between each pair of patches. A constant number s_{vis} of points ($s_{vis} = 10$ in our implementation) is uniformly subsampled from each patch. These points serve as

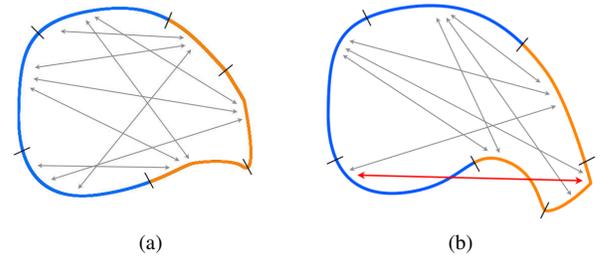


Fig. 4: Illustration of our mutual visibility criterion: given two components (in orange and blue), we merge them together in (a) since all the patches (separated by the little black bars) are visible to each other. In (b), the components are not merged as two patches (shown by the red line) are not mutually visible. Note that the actual criterion that we use is a relaxed version of this idea, as explained in Section 4.

source and target points for inter-patch ray casting, where a source and a target are visible to each other if the ray cast between them is not occluded by any other point in the shape. Using the sum of binary visibility for each of the s_{vis}^2 rays, we are able to estimate the ratio of visibility between a pair of patches with more confidence.

Component fusion. We fuse patches into larger weakly-convex components according to their *mutual visibility* (Figure 4). Our objective is to minimize the number of resulting components while keeping the intra-component mutual visibility above a threshold θ . This threshold quantifies the extent to which a fused component is allowed to deviate from perfect convexity. To this end, we employ an iterative fusion method, where the first iteration enforces strict mutual visibility between patches and the following iterations gradually relax this constraint. Each iteration fuses multiple pairs of adjacent patches in a greedy manner according to their mutual visibility.

Given a set of initial patches $\mathcal{P} = \{P_1, \dots, P_n\}$, we fuse them to obtain a set of weakly-convex components $\mathcal{C} = \{C_1, \dots, C_m\}$. Initially, each component C_i corresponds to patch P_i . To perform one fusion iteration, first we sort all pairs of adjacent components according to their visibility ratios in descending order. Next, following this ordering, we fuse all pairs of components C_i and C_j where θ is satisfied for all pairs of patches contained in C_i and C_j . This iterative approach is advantageous in that it encourages the creation of highly self-visible components in the earlier iterations, which are then further extended and fused with other components if their updated mutual visibility allows it. In our experiments, we employed 3 iterations with $\theta_1 = 0.9$, $\theta_2 = 0.8$, and $\theta_3 = 0.7$.

5. MERGING OF COMPONENTS

The previous step segments the input shape into a set of weakly-convex components. In some cases, when a shape is comprised only of weakly-convex semantic parts, this already provides a meaningful segmentation of the shape. However, most shapes possess a more complex structure and require further component merging to reach a close-to-semantic segmentation. As defined in Section 3, we rely on the volumetric consistency within semantic parts to guide the merging. The consistency is determined by a component signature based on the shape diameter function.

Merging by volumetric signature. Given a set of weakly-convex components $\mathcal{C} = \{C_1, \dots, C_m\}$, for each component C_i , we assign a volumetric signature which is a histogram of SDF values h_i . To compute this signature, we begin by uniformly sampling

s_{vol}^i points on the surface of each C_i . Each sample point $\mathbf{p}_u \in C_i$ forms the tip of a cone with angle α , and in the direction $-\mathbf{n}_u$, where \mathbf{n}_u is the normal of \mathbf{p}_u . We then consider all rays between \mathbf{p}_u and all $\mathbf{p}_v \in C_i$ that fall within this cone. The lengths of these rays are weighted by the cosine of the angle between \mathbf{p}_u 's normal and the rays, to penalize rays that deviate too much from the normal. The median length is taken as the SDF value for that point. When no such rays fall within a point's cone, the point is marked and not taken into account in the signature computation. We use parameters $s_{\text{vol}}^i = \max(100, 0.1 \cdot |C_i|)$ and $\alpha = 2\pi/18$ in all of our experiments.

Using the SDF values for all $\mathbf{p}_u \in C_i$, we create a histogram h_i to capture the volumetric profile of C_i . Note that in cases where C_i is planar or nearly so, the vast majority of points in C_i will be marked as having no SDF value and the resulting h_i will be falsely empty. To remedy this, we mark any component with an empty histogram as *flat*. Flat components are then excluded from the next merge step and are dealt with later on.

Using the volumetric signatures, we calculate the dissimilarity between all pairs of components according to (2), resulting in a component distance matrix \mathcal{D} . We also take into account the degree of convexity of the seam between a pair of components. To formulate this we introduce the *seam sets* of a component. These sets characterize the boundaries to other components and are used to select which neighboring component is the best choice for merging. The seam sets are defined using the k -nearest neighbors graph \mathcal{G}_K as follows:

$$\text{CVXSeam}_{i,j} = \{(\mathbf{p}_u, \mathbf{p}_v) : \mathbf{p}_u \in C_i, \mathbf{p}_v \in C_j, (\mathbf{p}_u, \mathbf{p}_v) \in \mathcal{G}_K, \text{angle}(\mathbf{n}_u, \mathbf{n}_v) \leq \pi\},$$

$$\text{CNCSeam}_{i,j} = \{(\mathbf{p}_u, \mathbf{p}_v) : \mathbf{p}_u \in C_i, \mathbf{p}_v \in C_j, (\mathbf{p}_u, \mathbf{p}_v) \in \mathcal{G}_K, \text{angle}(\mathbf{n}_u, \mathbf{n}_v) > \pi + \epsilon\},$$

$$\text{Seam}_{i,j} = \{(\mathbf{p}_u, \mathbf{p}_v) : \mathbf{p}_u \in C_i, \mathbf{p}_v \in C_j, (\mathbf{p}_u, \mathbf{p}_v) \in \mathcal{G}_K\},$$

where $\text{CVXSeam}_{i,j}$ and $\text{CNCSeam}_{i,j}$ are the convex and concave seams between C_i and C_j , respectively, and $\text{Seam}_{i,j}$ is the entire seam between them. We use $\epsilon = 0.05 \cdot \pi$ in our implementation, in order to prevent very subtle concavities from influencing the results. For the same reason, CVXSeam purposefully only includes angles $\leq \pi$. By using the seam sets, we disregard any adjacency where $\frac{|\text{CNCSeam}_{i,j}|}{|\text{CVXSeam}_{i,j}|} \geq 0.85$. Finally, we merge any two adjacent components i and j such that $\mathcal{D}_{i,j}/\max(\mathcal{D}) \leq \sigma$. We use $\sigma = 0.12$ in our implementation.

Next, in case flat components have been detected during the SDF histogram creation, we merge them with adequate adjacent components. To avoid violating component convexity, the search is guided by the seam sets in the following manner. Given a flat component C_i , we sort its neighbors C_j by descending order of $|\text{Seam}_{i,j}|$, allowing us to consider first neighboring components with which C_i shares the longest borders. Going over the sorted neighbors, we select the first one that shares a strictly convex seam with C_i . If such a neighbor does not exist, we sort the neighbors by descending order of $\frac{|\text{CVXSeam}_{i,j}|}{|\text{CNCSeam}_{i,j}|}$ and select the first neighbor for which $\frac{|\text{CVXSeam}_{i,j}|}{|\text{CNCSeam}_{i,j}|} \geq 1$ and $\theta_{i,j} \geq 0.4$, where $\theta_{i,j}$ is the mutual visibility between C_i and its neighbor C_j . This step favors merging C_i with a neighbor according to the extent of convexity of the shared seam, as well as mutual visibility. If no adequate neighbor is found, C_i is left unmerged. Finally, to account for the newly-merged flat components, the histograms of the changed components are updated and one more merge iteration is performed as described above.

Note that, if we simply estimate the SDF for each point in an incomplete shape, and then cluster the points according to their SDF, the results will not be satisfactory, as the discontinuities in the volume of the data will create discontinuities between the parts. On the other hand, estimating the volumetric profile of a weakly-convex component is more robust in this regard, as we obtain a volumetric signature for it.

Border refinement. As the final step, we employ a point-level graph cut optimization on the merged components to refine the boundaries between the segments. The k -nearest neighbors graph \mathcal{G}_K is used to imply the connectivity between points, and we create a label for each segment. We set the data term to strictly preserve each segment, but allow the borders to move along a narrow band between segments to better align the borders with the concavities of the shape. We define the data term formally as:

$$\mathcal{D}_u(f_u) = \text{cost}_l(u) / \max_v \text{cost}_l(v), \text{ when } f_u = l, \quad (3)$$

$$\text{where } \text{cost}_l(u) = \begin{cases} 0, & \text{if } s_u = l, \\ \text{dist}_{B_l}(u)^{1.5}, & \text{otherwise.} \end{cases} \quad (4)$$

Here, s_u is the label of point u in the original segmentation, f_u is the label assigned by the graph cut, and $\text{dist}_{B_l}(u)$ is the distance of the point to the boundary B_l of the segment with label l . The power 1.5 increases the cost faster for points away from the boundary, but in a more gentle manner than when using a quadratic power.

The smoothness term follows the Potts model weighted by the angle between two points, to assign a smaller cost to cuts at concave regions. We define the smoothness cost as:

$$\mathcal{S}_{u,v}(f_u, f_v) = \begin{cases} \beta_{u,v}/\pi, & \text{if } f_u \neq f_v, \\ 0, & \text{if } f_u = f_v. \end{cases} \quad (5)$$

where $\beta_{u,v}$ is the angle between the normals of points u and v . The data cost is weighted by 0.1 when adding up the data and smoothness costs. In summary, in the formulation above, the additional cost of moving the boundary away from the original segment is balanced with the cost of the segment cut.

6. RESULTS

In this section, we evaluate our segmentation algorithm on a collection of incomplete shapes, and also present a quantitative comparison to state-of-the-art unsupervised mesh segmentation algorithms. Note that our source code, datasets, and results are available at <http://www.cs.tau.ac.il/~noafish/wcseg/>.

Segmentation results. Figure 5 presents visual examples of segmentation results that we obtain on diverse point clouds acquired with a range scanner, where many of the shapes are incomplete and have missing portions. We notice that the method is robust when applied to these shapes and is able to implicitly complete some of the missing portions with the weakly-convex decomposition. For example, the ‘yoga poses’ shapes possess several missing portions in the head, body and limbs, while the airplane is made of a single scan from the top of the model. Note also that it would not be possible to segment some of these models relying only on shape concavities. In general, several of the semantic components are correctly segmented. However, since the point clouds are very sparse in certain regions, we also see that some parts were mistakenly merged with their neighboring parts, due to the absence of significant volumetric information or concavities in the shape.

Figure 6 presents additional results obtained on triangle meshes, where we first transform the meshes into point clouds, and then



Fig. 5: Segmentations obtained with our non-parametric algorithm on a collection of point clouds acquired with a range scanner.

map the segmentation results back to the meshes. We create a point corresponding to the center of each triangle, which then gives a natural mapping to transfer the segmentation of the point cloud back to the triangle mesh. In this way, we do not introduce resampling artifacts. We can see that the method also performs satisfactorily on these complete shapes, obtaining meaningful segmentations similar to those of state-of-the-art methods applied directly on the meshes.

Analysis of convergence. Figures 1 and 7 present an analysis of the convergence of the algorithm when applied to incomplete shapes, such as partially-scanned shapes. We start with the segmentation obtained by scanning an input shape from a single viewpoint, and then show the progress of the algorithm as more views are added to the scan. For these experiments, we use a virtual scanner to acquire samples from a watertight mesh from different viewpoints. Thus, the results obtained on the point cloud formed of several views approach those obtained on the original mesh. From



Fig. 6: Segmentations obtained with our non-parametric algorithm on a collection of triangle meshes. A quantitative evaluation on the full segmentation benchmark with 380 meshes is reported in Figure 8.

this analysis we see that, as more scans are added and the volume of the shape is better represented, our method approaches the quality of results obtained on the mesh. Even after adding only a few scans, results similar to the original mesh are obtained, due to the robustness of the algorithm to holes in the shapes. Moreover, Figures 1 and 7 also serve as an analysis of the performance of the algorithm at various levels of incompleteness of the shapes.

Quantitative evaluation. To provide a comparison to state-of-the-art segmentation algorithms, we evaluate our method with the benchmark and protocol of Chen et al. [2009]. We also compare to the concavity-aware segmentation method of Au et al. [2012], since it has been shown to outperform the other methods in the benchmark. Although the benchmark is composed of a collection of 380 watertight triangle meshes (not point clouds), we use it to evaluate our algorithm since this type of shapes has been the fo-

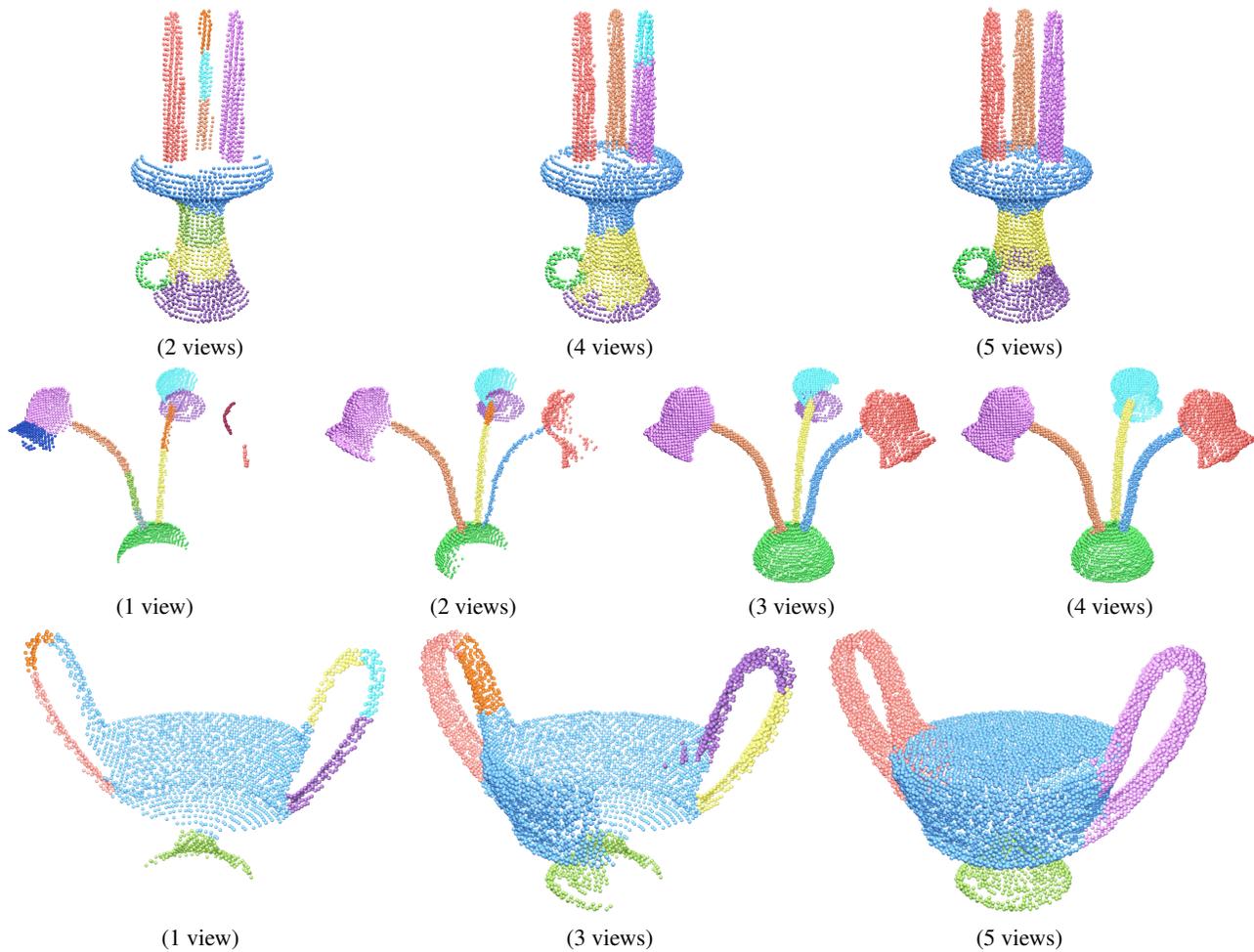


Fig. 7: Convergence of the segmentation results as the point clouds are complemented with an increasing number of scanned views. Notice how with a small number of views, the results are already of similar quality to the results with several views (in the last column).

cus of previous work on shape segmentation. To run our algorithm, we first transform the meshes into point clouds, and then map the segmentation results back to the meshes, as described before. Figure 8 shows a comparison of the algorithms according to the various measures of the benchmark. Table I presents a ranking of the algorithms based on the Rand Index. In short, the Rand Index measures the likelihood that a pair of faces are either in the same segment in two segmentations, or in different segments in both segmentations. Please refer to Chen et al. [2009] for an explanation of the other metrics and more details about the evaluation protocol.

From the results in Figure 8, we can see that our algorithm is comparable to the state-of-the-art, achieving results similar to concavity-aware segmentation, shape diameter function and randomized cuts, while also having the advantage of being applicable to incomplete shapes. According to the ranking of Table I, our algorithm performs the best for more classes of shapes than any other algorithm and is the most consistent among different classes as evident by the low rank achieved on all classes.

Comparison to decomposition by clustering. In Figure 9, we show a comparison of our method to the weakly-convex de-

composition of Asafi et al. [2013], which clusters points into components according to their mutual visibility. It can be observed that by first creating a set of initial patches, we are able to obtain an accurate estimate of the visibility between these patches and better control the approximate convexity of each part. The resulting segments have a higher convexity rank than those obtained with the method of Asafi et al. [2013]. Moreover, their method tends to create less desirable boundaries between the parts, e.g., the blue segment spans from the left elbow of the ballerina across her head to her right arm, due to the partial visibility between these points.

Scenes with multiple objects. Although the focus of our method is the segmentation of single objects into meaningful parts, as given by our part characterization, we also show in Figure 10 an example of applying our method to the segmentation of point clouds corresponding to scenes with multiple objects. In this example, we segment a virtual scan of an indoor office scene. The point cloud was created by registering scans from three different views, where we separated the foreground objects from the background (floor and walls) before the segmentation. We see that the

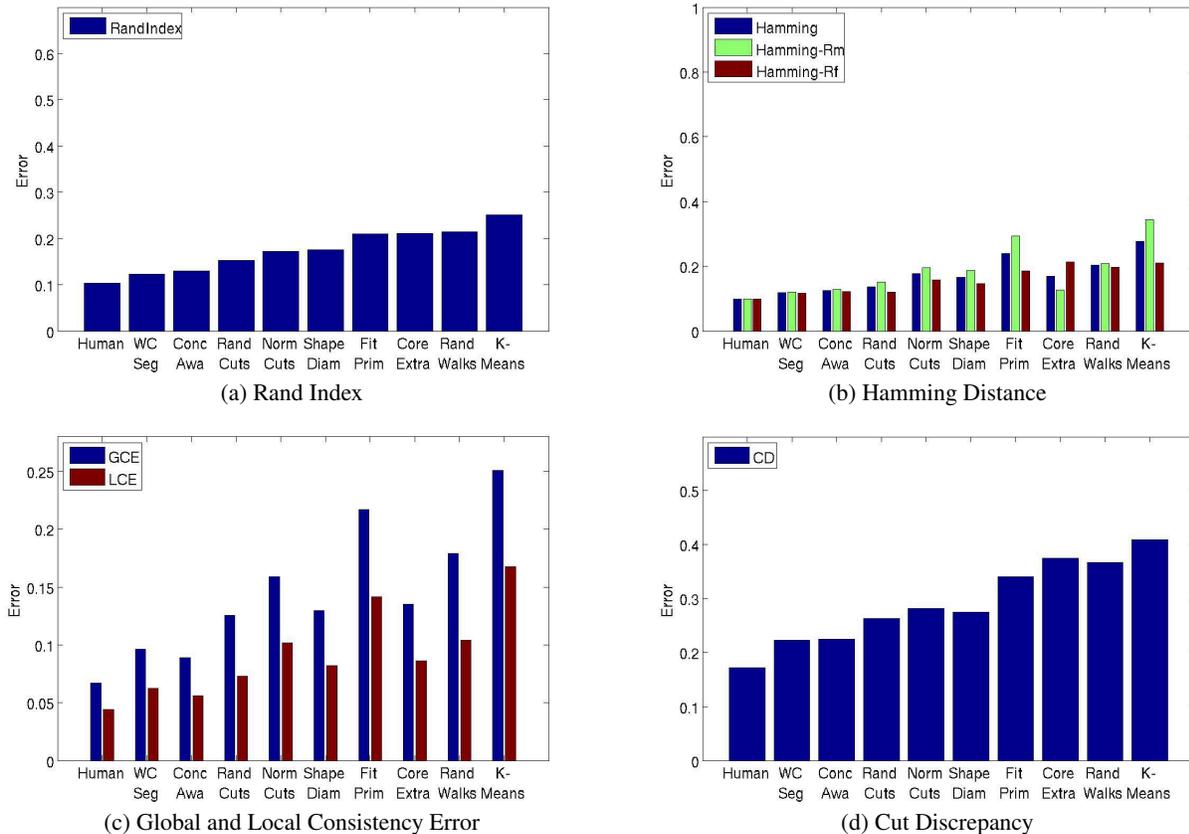


Fig. 8: Quantitative evaluation of our segmentation algorithm on the benchmark of Chen et al. [2009]. Our algorithm is named WC Seg. Notice that our algorithm is comparable to the other top-ranking algorithm (concavity-aware field segmentation).

combination of visibility and volumetric information also has the potential of addressing this multi-object segmentation problem.

Parameter sensitivity. Our method is non-parametric in that we do not need to specify the number of expected segments. However, our method uses a number of parameters that have to be properly set. We have tested a reasonable range for these parameters and obtained similar results to the ones shown here. More specifically, we varied each of the parameters θ_1 , θ_2 , θ_3 , and σ around the selected values, since these are the parameters that most impact the created segments. We verified that in a range within 6% of the value of each parameter ($\pm 3\%$ added to each value), the quality of the results is similar. Other parameters such as s_{vis} and s_{vol}^i impact less on the segmentation and are kept fixed, as their main function is to increase the accuracy of the visibility and volume estimates. While certain classes of shapes may benefit from adjusting some of the parameters, the selected parameters are not class-specific; we obtained all of our results with a single set of parameters.

Timing and complexity. Our current unoptimized implementation takes on average 3 minutes to process 10K points, where the most expensive operation is the creation of initial patches. Note that we keep constant the number of initial patches and number of rays used to compute the visibility between patches. Thus, the complexity grows linearly with the number of points, since the main operation is searching for the points that are hit first by the rays cast. This

step can be improved with the use of appropriate data structures for faster intersection computations.

7. CONCLUSION AND FUTURE WORK

We have presented a non-parametric segmentation algorithm that uses a characterization of the expected geometry of a part to decompose the shape. The main idea is to characterize a part as a collection of weakly-convex components that possess a similar volumetric profile. Thus, we directly search for the parts that fit this characterization, rather than follow a Markovian approach based on the search of a graph cut that agrees well with the shape concavities. Note that, although we also use a graph cut optimization to refine the segmentation borders, this is mainly a post-processing step, as the decomposition is guided primarily by our part characterization. Moreover, we demonstrated that by means of a weakly-convex decomposition, the method is robust to missing data. We are able to implicitly complete missing portions of the shapes by using the weakly-convex components as natural priors. This is a key aspect for the successful segmentation of incomplete shapes.

Limitations. One limitation of our approach is that not all semantic parts adhere to our part characterization. As shown in the example in Figure 11, there exist semantic parts that are composed of two or more weakly-convex components with different geometric properties. In general, semantic parts that possess concavities,

Object Categories	WC Seg	Conc Awa	Rand Cuts	Shape Diam	Norm Cuts	Core Extra	Rand Walks
Human	2	1	3	4	7	5	9
Cup	1	2	3	4	7	8	5
Glasses	4	1	2	3	6	7	8
Airplane	1	4	3	6	2	5	9
Ant	1	4	3	5	2	8	6
Chair	2	4	6	1	3	8	7
Octopus	1	2	6	5	3	9	4
Table	2	1	9	3	6	5	7
Teddy	3	2	1	6	4	8	5
Hand	3	2	1	5	9	8	6
Plier	2	1	4	6	9	5	3
Fish	1	2	5	7	3	9	4
Bird	2	1	3	6	4	8	5
Armadillo	1	5	4	7	2	3	9
Bust	3	2	1	8	5	6	7
Mech	3	2	6	1	5	7	8
Bearing	2	6	3	4	1	5	9
Vase	2	3	1	5	6	8	4
FourLeg	1	4	3	8	2	5	6
Overall	1	2	3	4	5	6	7

Table I.: Comparison of segmentation algorithms for each object category. Entries represent the rank of the algorithm (1 is the best) according to the Rand Index. We display only the top seven algorithms in the table.

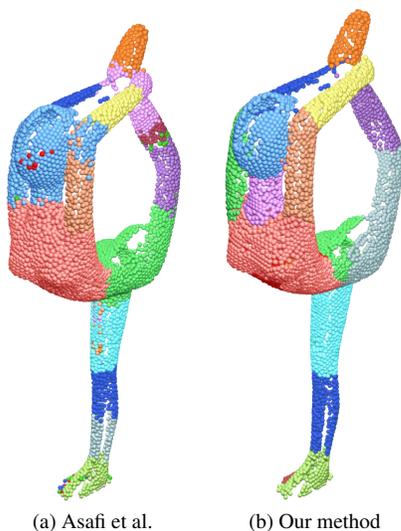


Fig. 9: Comparison of our method to Asafi et al. [2013]. Note how the segments obtained with our method have a higher convexity rank.

such as cups and open liquid containers, will be separated into multiple components. Similarly, there may be nearby semantic parts that do possess similar properties, but should remain separated, such as chairs made of a combination of parts with the same diameter, or humans where the arm and forearm have a similar radius. Another limitation is that when the data is sparse (e.g., a scan from a single viewpoint), there may not be much volumetric information available in the point cloud. Then, the computation of the volumetric profile will not be accurate enough to guide the part merging.

The presence of concavities affecting the segmentation is closely-related to the fact that, although our method is non-parametric (where the user does not have to specify the number of segments), the user still needs to specify the degree of weak convexity of the parts (θ threshold). In addition to θ , there are other parameters in the algorithm that need to be selected adequately.



Fig. 10: A point cloud corresponding to an indoor scene with multiple objects, segmented with our method.



Fig. 11: Limitations of our approach: (1) Thin structures (such as the chair legs) have low internal visibility, thus they may not be joined during the component fusion step; (2) When the surface has variance in its width (as in the top part of the back of the chair or the corners of the seat), the components may not be joined in the merging step since the bins of their volumetric histograms are spread across a short range of values.

Nevertheless, as discussed in Section 5, we have demonstrated that meaningful results on a large benchmark of shapes from different categories can be obtained with a fixed set of parameters.

Future work. Although our segmentation is guided by a part characterization, the search is still based on heuristics. Thus, a logical step for improving the quality of the results is to search for the optimal parts with more advanced optimization techniques, such as linear programming. Moreover, we can look at additional geometric properties when merging the weakly-convex components into larger segments. For example, it is possible to fit specific primitives to the components and estimate a richer set of geometric properties. Finally, we also see potential in the application of the decomposition for other shape analysis tasks, such as shape completion or data-driven shape analysis, where the weakly-convex components can serve as primitives for the computation of descriptors and statistical learning of shape properties.

Acknowledgements. We thank Hui Huang for providing some of the point cloud models and Chen et al. for the segmentation benchmark. This work is supported in part by the Israeli Science Foundation (grant no. 1790/12) and the U.S.-Israel Binational Science Foundation (grant no. 2012376). Oliver van Kaick is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

REFERENCES

- ANGUELOV, D., TASKARF, B., CHATALBASHEV, V., KOLLER, D., GUPTA, D., HEITZ, G., AND NG, A. 2005. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *Proc. IEEE CVPR*. 169–176 vol. 2.
- ASAFI, S., GOREN, A., AND COHEN-OR, D. 2013. Weak convex decomposition by lines-of-sight. *Computer Graphics Forum (SGP)* 32, 5, 23–31.
- ATTENE, M., KATZ, S., MORTARA, M., PATANE, G., SPAGNUOLO, M., AND A.TAL. 2006. Mesh segmentation – a comparative study. In *Proc. Shape Modeling International (SMI)*.
- ATTENE, M., MORTARA, M., SPAGNUOLO, M., AND FALCIDIENO, B. 2008. Hierarchical convex approximation of 3D shapes for fast region selection. *Computer Graphics Forum* 27, 5, 1323–1332.
- ATTENE, M. AND PATANÈ, G. 2010. Hierarchical structure recovery of point-sampled surfaces. *Computer Graphics Forum* 29, 6, 1905–1920.
- AU, O.-C., ZHENG, Y., CHEN, M., XU, P., AND TAI, C.-L. 2012. Mesh segmentation with concavity-aware fields. *IEEE Trans. Vis. & Comp. Graphics* 18, 7, 1125–1134.
- CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. on Graph (SIGGRAPH)* 28, 3, 73:1–12.
- GAL, R., SHAMIR, A., HASSNER, T., PAULY, M., AND COHEN-OR, D. 2007. Surface reconstruction using local shape priors. In *Symp. on Geom. Proc.* 253–262.
- GOLOVINSKIY, A. AND FUNKHOUSER, T. 2008. Randomized cuts for 3D mesh analysis. *ACM Trans. on Graph (SIGGRAPH Asia)* 27, 5, 145:1–12.
- GOLOVINSKIY, A. AND FUNKHOUSER, T. 2009. Min-cut based segmentation of point clouds. In *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*.
- HOFFMAN, D. D. AND RICHARDS, W. A. 1984. Parts of recognition. *Cognition* 18, 1–3, 65–96.
- HU, R., FAN, L., AND LIU, L. 2012. Co-segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum* 31, 5, 1703–1713.
- HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation with linear programming. *ACM Trans. on Graph (SIGGRAPH Asia)* 30, 6, 125:1–12.
- KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. on Graph (SIGGRAPH)* 29, 3, 102:1–12.
- KIM, Y. M., MITRA, N. J., YAN, D.-M., AND GUIBAS, L. 2012. Acquiring 3D indoor environments with variability and repetition. *ACM Trans. on Graph (SIGGRAPH Asia)* 31, 6, 138:1–11.
- KOPPULA, H. S., ANAND, A., JOACHIMS, T., AND SAXENA, A. 2011. Semantic labeling of 3D point clouds for indoor scenes. In *Proc. NIPS*.
- KRAEVOY, V., JULIUS, D., AND SHEFFER, A. 2007. Model composition from interchangeable components. In *Proc. Pacific Graphics*. 129–138.
- LAFARGE, F., ALLIEZ, P., ET AL. 2013. Surface reconstruction through point set structuring. *Computer Graphics Forum (Eurographics)* 32, 2, 225–234.
- LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J. 2011. Globfit: consistently fitting primitives by discovering global relations. *ACM Trans. on Graph (SIGGRAPH)* 30, 4, 52:1–12.
- LIEN, J. AND AMATO, N. 2007. Approximate convex decomposition of polyhedra. In *Proc. ACM symposium on Solid and physical modeling*. 121–131.
- LIU, R., ZHANG, H., SHAMIR, A., AND COHEN-OR, D. 2009. A part-aware surface metric for shape analysis. *Computer Graphics Forum (Eurographics)* 28, 2, 397–406.
- LU, L., CHOI, Y.-K., WANG, W., AND KIM, M.-S. 2007. Variational 3D shape segmentation for bounding volume computation. *Computer Graphics Forum* 26, 3, 329–338.
- MENG, M., XIA, J., LUO, J., AND HE, Y. 2013. Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization. *Computer-Aided Design* 45, 2, 312–320.
- NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM Trans. on Graph (SIGGRAPH Asia)* 31, 6, 137:1–10.
- REN, Z., YUAN, J., LI, C., AND LIU, W. 2011. Minimum near-convex decomposition for robust shape representation. In *Proc. Int. Conf. on Comp. Vis.* 303–310.
- SCHNABEL, R., DEGENER, P., AND KLEIN, R. 2009. Completion and reconstruction with primitive shapes. *Computer Graphics Forum (Eurographics)* 28, 2, 503–512.
- SHAMIR, A. 2008. A survey on mesh segmentation techniques. *Computer Graphics Forum* 27, 6, 1539–1556.
- SHAO, T., XU, W., ZHOU, K., WANG, J., LI, D., AND GUO, B. 2012. An interactive approach to semantic modeling of indoor scenes with an RGBD camera. *ACM Trans. on Graph (SIGGRAPH Asia)* 31, 6, 136:1–11.
- SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer* 24, 4, 249–259.
- SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. 2011. Real-time human pose recognition in parts from a single depth image. In *Proc. IEEE CVPR*. 1297–1304.
- SIDI, O., VAN KAICK, O., KLEIMAN, Y., ZHANG, H., AND COHEN-OR, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graph (SIGGRAPH Asia)* 30, 6, 126:1–10.
- SIMARI, P., NOWROUZEZAHRAI, D., KALOGERAKIS, E., AND SINGH, K. 2009. Multi-objective shape segmentation and labeling. *Computer Graphics Forum (SGP)* 28, 5, 1415–1425.
- THIERY, J.-M., GUY, E., AND BOUBEKEUR, T. 2013. Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Trans. on Graph (SIGGRAPH Asia)* 32, 6, 178:1–12.
- XIONG, X., MUNOZ, D., BAGNELL, J., AND HEBERT, M. 2011. 3-D scene analysis via sequenced predictions over points and regions. In *Proc. IEEE Robotics and Automation (ICRA)*. 2609–2616.
- ZHANG, J., ZHENG, J., WU, C., AND CAI, J. 2012. Variational mesh decomposition. *ACM Trans. Graph.* 31, 3, 21:1–14.