Analysis of face and segment level descriptors for robust 3D co-segmentation

David George
d.george.654214@swansea.ac.uk
Gary K.L. Tam
k.l.tam@swansea.ac.uk
Xianghua Xie
X.Xie@swansea.ac.uk

Computer Science Department
Swansea University
Swansea, UK

Abstract

3D shape co-segmentation is an important topic in computer graphics. The idea of co-analysis brings new insights into understanding a collection of shapes. Rather than analysing individual shapes, an entire set is looked at, giving much more information about the class of shape in the set. Existing co-segmentation techniques use both face and segment level descriptors in order to provide enough information to give an accurate co-segmentation result. In the literature, a lot of these descriptors are proposed but there is limited empirical studies to compare which would perform well. In this paper, we have two aims: (a) propose new useful face and segment level descriptors and (b) analyse the effectiveness of them. Our experiment indicates that smoothly varying descriptors (Average Euclidean Distance) that respects geometry would improve the segmentation results.

1 Introduction

Segmentation of 3D shapes is a process of breaking a shape into meaningful parts. These parts usually have semantic meaning. Good partitioning algorithms should produce segments similar to those that a human could produce. The ability to do this segmentation automatically and with little or no supervision is a fundamental problem in computer graphics.

There have been a lot of work in this area in recent years. These include, core extraction [12], shape diameter function (SDF) [20], k-means [21], mean-shift [6], random walks [14], fitting primitives [2] and normalised and randomised cuts [7]. All these propose to segment a single shape based on various feature descriptors extracted from the shape itself. A more detailed discussion of these mesh segmentation techniques can be found in some of the recent surveys [3, 5, 19]. Even though these techniques all showed promising results on certain shapes, no one algorithm can provide good results for every type of shape [5]. This is likely due to the large variation between individual shapes. Certain methods which are good at segmenting one type of shape, can then struggle with another type and vice versa.

This then raises the question of how to develop a technique which can accurately segment many different types of shapes. One solution, which has been getting a lot of attention in recent work, is to look at a set of shapes, rather than individual or pairs of shapes [22].

Analysing a set as a whole will give much more information about the classes of shapes in the set and help describe it. This method is called co-analysis.

Some of these techniques include co-hierarchical analysis of a set of shape structures [24], joint segmentation of heterogenous shape sets using linear programming [10], co-analysis of a set of shapes through active learning [25], learning and fitting part-based templates in large collections of 3D shapes [13] and finding a consistent set of part arrangements in a set of 3D shapes [27].

Co-segmentation, similarly, considers extracting information from the whole set (like co-analysis) and then providing a consistent segmentation across the set (i.e., corresponding segment labels across the set are semantically similar). The idea started in image processing with the work by [17], who worked on a pair of images. They use a generative model and match the appearance histogram of each image in the pair, while enforcing spatial consistency between the images. Also, other work in the area by [11], which uses discriminative clustering in order to find a maximal separation between classes in the set, giving a co-segmentation of the images.

Following the work in the image domain, pioneering work in the 3D shape domain was presented by [8], which involved aligning the shapes and finding correspondence points between meshes. This resulted in similar shaped objects getting the same label from the segmentation, thus giving a consistent segmentation throughout the set. [22] proposed to use shape descriptors and spectral clustering. This removed the requirement for the shapes to be spatially similar, as the descriptors would differentiate between key parts of the shapes giving a consistent segmentation throughout the set.

[15] and [26] follow a similar process of [22] but differ in terms of using various descriptors and initial segmentation techniques. [15] uses 4 face level descriptors and no segment level descriptors in their work. They first segment each mesh into primitive patches and calculate a dissimilarity matrix between all pairs using the descriptors. An affinity matrix is then constructed and Normalized Cuts [7] are used to cluster them. From here an iterative multi-label optimisation is run on the clusters to refine the results. [26] uses 5 face level descriptors and no segment level descriptors in their work. They cluster the mesh into patches and then run spectral clustering on each of the descriptor spaces, thus giving 5 separate spectral spaces. These spaces are then fused using affinity aggregation, this is meant to give a better embedding compared to embedding concatenated descriptors. All these described approaches are unsupervised, demonstrating that co-segmentation, taking into account of a whole dataset, would produce meaningful results, and is comparable to supervised techniques.

Contributions This work focuses on the analysis of co-segmentation techniques. From our observation, all these techniques use similar co-analysis processing pipeline, but different shape descriptors. Our main research question is *how effective are these shape descriptors towards a better co-segmentation technique?* In particular, we (a) propose to use new descriptors that have not been used in existing literature. These include face level descriptors: SHOT signatures [23], Average Euclidean Distance (AED) and segment level descriptor D2 histogram [16]; (b) investigate if certain combination of these descriptors work together with existing ones, namely, Average Geodesic Distance (AGD), shape context (SC), and shape diameter function (SDF).

In Section 2 we first discuss the processing pipeline of co-segmentation as well as the existing descriptors that they use. In Section 3, we discuss our methodology, namely our adjustment of existing pipeline of [22] and descriptions of each of our new introduced descriptors. In Section 4, we show our experimental results compare them to existing tech-

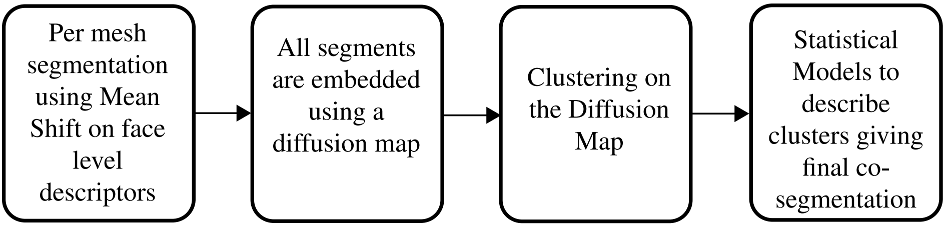
Per mesh segmentation using Mean Shift on face level descriptors
All segments are embedded using a diffusion map
Clustering on the Diffusion Map
Statistical Models to describe clusters giving final co-segmentation

Figure 1: The pipeline of the co-segmentation technique in [22]. First each shape in the set is segmented with mean-shift [6], then all segments from all shapes are gathered and embedded into a diffusion space where they are clustered. The clusters are then described using a statistical model and the results are refined.

nique [22]. In Section 5, we further discuss some observations during our implementation. Finally, in Section 6, we conclude with possible future work.

2 Background

In order to test the effectiveness of 3D descriptors on co-segmentation we opted to employ the techniques described in [22]. Here, we briefly introduce the technique for consistently segmenting a set of 3D shapes. All shapes in the set belong to the same class (i.e., chairs, vases etc.), and segmentation is carried out by only using descriptors obtained from the shapes. Figure 1 shows the pipeline which [22] follows.

The work first extracts three face level descriptors, namely, (i) the angle between the normal of the face and the upright orientation vector, (ii) the geodesic distance between the face and the base of the shape and (iii) the shape diameter function (SDF) [20], which estimates the thickness of the shape at a point. These descriptors are used to determine the initial segmentation of each shape via mean-shift [6].

They then define 5 segment level descriptors. The first three are histograms of the distributions of all three face level descriptors in that segment. The next two segment level descriptors are the area of each segment normalised by the total area of the shape, and a vector describing the overall geometry of the segment. This vector is used as it gives an indication of how linear, planar and spherical the segment is. Its components are calculated from eigenvalues obtained from applying PCA on the vertices in the segment (See [22] for more details).

Our method mainly follows the technique discussed above. We pick this technique because it is the pioneering work in 3D shape co-segmentation that uses descriptors. A similar pipeline is also subsequently used in relevant work [15, 26], with slightly different stages and descriptors. We will compare the results obtained with the addition of new descriptors to the results using only the original descriptors.

3 Methodology

Here we outline the method behind our work. We describe the revised pipeline from our re-implementation of [22] and also describe each of the descriptors that we have chosen to

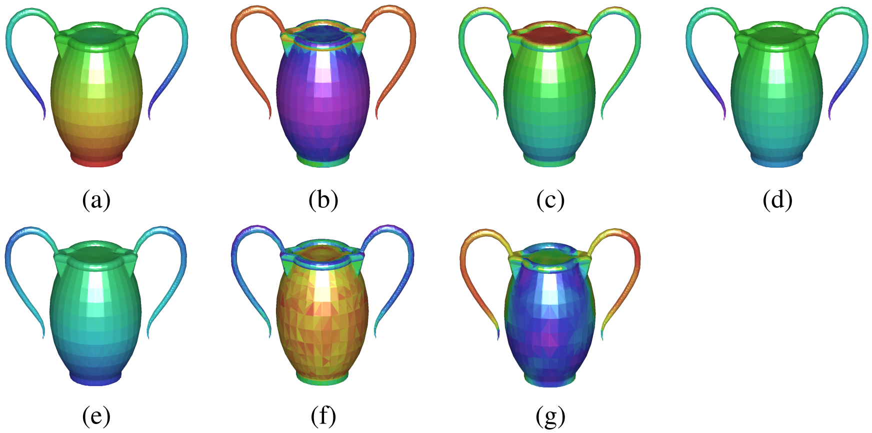

Figure 2: Distribution of each of the face level descriptors for a vase. (a) Geodesic distance from the face to the base. (b) SDF (c) Angle to the upright vector. (d) AGD (e) AED (f) SC (g) SHOT. The colours represent the value of the descriptor at that face. Low values are mapped to hot colours (reds) high values are mapped to cold colours (blue and violet)

use in the process.

Our main research question is to analyse how effective other 3D descriptors (both new and used in more recent work) are in a working method of co-segmentation [22]. From our experience, graph-cut can further improve the segmentation results, but it is heavily sensitive to a good initial segmentation and tuning. To better compare the effectiveness, we opted to remove this final refinement stage and calculate the accuracies after the diffusion process. These results are then compared to [22] with our faithful re-implementation. Because [22] has achieved a fairly high accuracy (over 70%), we intend to start off by adding descriptors to the existing ones in the pipeline. Since most existing descriptors used in [22] are scalars or small vectors, our choices of descriptors will follow similar strategy here.

3.1 Face level descriptors

Apart from the three face level descriptors defined in [22], we also introduce four new face level descriptors to the pipeline. Two of them have been used in recent co-analysis work (namely SC and AGD) [15, 26]. Average Euclidean Distance is a variation of AGD, and the SHOT descriptor [23] has not yet been used in co-analysis:

1. **Average Geodesic Distance (AGD) [9]** Given a face, the geodesic distance is calculated to every other face on the shape. These values are then averaged to give the AGD value for the face. This measure is used in several other co-analysis works [15, 26] and gives a good indication of where the face lies in the shape. If the face is far from the majority of other faces it lies far from the center of the shape, and thus a large value.

2. **Average Euclidean Distance (AED)** is similar to AGD, but uses Euclidean distance between faces instead of geodesic distances. This measure is less sensitive than AGD, see Figure 2. The AGD descriptor has to follow the surface, so has high values at the tips of the handles. The AED, on the other hand, does not rely on following the shapes surface, so its values at the tips of the handles are lower.

3. **Shape Contexts (SC) [4]** is a 3D histogram which represents the distribution of points within a sphere defined around the current point. This descriptor describes a local neighbourhood around a point. Similar to the AGD, this descriptor is also popular in recent co-analysis works. However, their work define it as a scalar and not a 3D histogram. Here, we use PCA to extract first five principal components as a face level descriptor.

4. **Signature of Histogram of Orientations (SHOT)** [23] is a descriptor that combines both histogram and signature with a robust local reference frame. It has shown to be good for finding point-to-point correspondence. We average the point descriptor to obtain face level descriptor and compress it using PCA. We use these to summarise the surface characteristics, and provide an alternative descriptor similar to SC.

Figure 2 shows the values of all face level descriptors for a single vase shape. It shows how different descriptors can make specific regions of a shape stand out. For example, the SDF values are very similar throughout the handles, which describes them well.

3.2 Segment level descriptors

Here, we further introduce five new segment level descriptors. Similar to [22], for every segment, we define a histogram for each of the four new face level descriptors. Moreover, the histogram represents the distribution of all values of that descriptor in the segment. There is one histogram for each descriptor per segment. We further define another segment level descriptor **D2 histogram [16]**. It is a distribution of Euclidean distances between pairs of randomly sampled points in a shape. It is an easy to compute descriptor and is frequently used in 3D shape retrieval literature. Here, we use them to describe a segment (rather than a shape). In our experiments we opt for a bin count of 20 and the number of sampled points depends on the size of the segment.

3.3 Implementation

The usage of these descriptors in our revised pipeline is as follows. For each test,
Compute Face Level Descriptors For every face on each mesh we compute the face level descriptors.
Per-Mesh Segmentation Together with the original descriptors [22], we add new descriptors (see Section 3.1) to compute initial per-mesh clustering using mean-shift for each shape.
Compute Segment Descriptors For each segment, we produce a histogram of the distribution of each face level descriptor in that segment. Also the original segment level descriptors are computed, and depending on the test, the D2 histogram [16] is calculated.
Diffusion Process and Co-Segmentation For every pair of segments a dissimilarity cost is calculated (See[22]). This cost is then weighted by a Gaussian kernel and then embedded into a diffusion space. Clustering is then applied in the embedded diffusion space to provide the final segment result.

The above process is then iterated for each test, and the parameters are randomly initialised for each iteration.

The experiments are carried out on all of the original descriptors and one or more new descriptors. These results are then compared to the results from only using the original descriptors [22].

4 Experiments and Results

We will now outline some of our experimental results obtained using the descriptors described in Section 3. Our experiments were done on 4 datasets from the COSEG dataset [25], namely candelabra, vases, guitars and goblets. We chose these as it gives an indication of the descriptor strength on both easier (goblets) and harder (vases, candelabra and guitars) sets.

The accuracy of the results is calculated using the same measure defined in [22]:

$$Accuracy(l,gt) = \frac{\sum_i a_i \delta(l_i = gt_i)}{\sum_i a_i} \tag{1}$$

where a_i is the area of face i, l_i is the label assigned to face i by the diffusion process and gt_i is the ground truth label of face i. $\delta(l_i = gt_i)$ is assigned to 1, if and only if, the assigned label is the same as the ground truth label; otherwise zero.

Our implementation of the pipeline is written mainly in MATLAB, with some of the functions in C++. We utilise the Point Cloud Library [18] for computing the shape context descriptor, and the CGAL library [1] for their implementation of the shape diameter function. For each dataset we pre-compute and store the SHOT, SC and AGD descriptors once as they take the longest to compute and reload them throughout all test runs.

Throughout our tests, we experiment the best parameters for each dataset for the given test. These parameters include the σ used in the Gaussian kernel for computing the affinity matrix and also the bandwidth of search space used for mean-shift. We have to do so because such parameters are not discussed in the original paper [22]. Some examples of our results can be seen in Figure 3.

For our tests, we sampled values for the bandwidth of the mean shift in the range of [0.2 : 0.6], typically getting reasonable initial segmentations for most sets with values between 0.25 and 0.4. It is worth noting that we keep this bandwidth constant for all the shapes in a given test.

Similarly, we estimate the σ required in the Gaussian kernel, which is part of the diffusion process. We tried using values such as the standard deviation of the Gaussian's input set (the dissimilarity matrix), and the mean of this input. Both did not give good results, the former was too sensitive and the latter was not sensitive enough. Due to this we opted to use the mean of the set but then reduce the value by using a weight. This weight varied heavily between the sets and experiment, but we sampled values in the range of (0 : 1].

Performance Our implementation of the pipeline took roughly 5-6 minutes per test, this does not include the pre-computation involved in computing the three descriptors mentioned above. The tests were run on a Intel Core i7 4.00 GHz processor with 32 GB memory. That said, only about 4GB was ever used for a single test. See Table 2 for a breakdown of the timings of all tests.

5 Discussion

Our initial expectation of the results is that AGD and SC should produce improvement as they are already mentioned in relevant works [15, 26]. Also AED, which is similar to AGD, should provide similar performance improvement. Finally, SHOT, which is similar to SC, describes distribution around points, should provide similar or better performance.

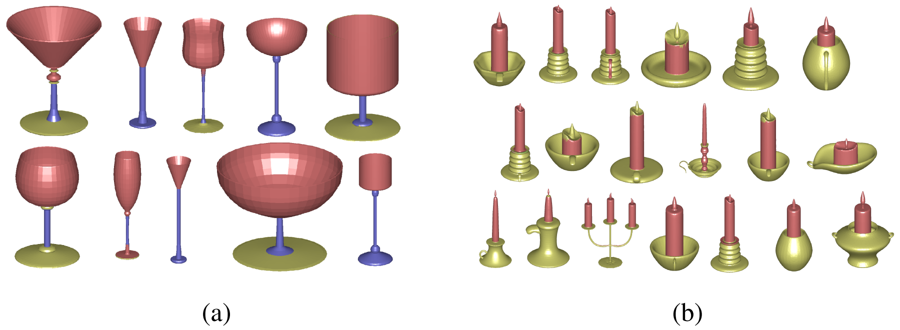

Figure 3: Results from our tests. (a) goblets dataset (b) candelabra dataset. Corresponding segments throughout the set are denoted by the same colour.

Dataset	Original	+AGD	+AED	+SC	+SHOT	+D2	+AGD+AED+D2
Candelabra	80.3	**80.4**	**84.0**	**87.0**	67.1	73.1	**86.3**
Goblets	84.2	82.3	**85.7**	82.9	76.3	**90.3**	83.7
Vases	74.3	74.1	**77.1**	**78.7**	66.3	72.9	**76.2**
Guitars	85.9	81.6	85.0	79.5	77.2	79.5	84.7

Table 1: The average co-segmentation accuracy. Original is using only the descriptors from [22], then each subsequent column is the original features plus the new descriptor(s), bold results indicate an increase in accuracy over the original. These results have had no graph-cut refinement process.

Despite our results showing improvement, they is not perfect (e.g., none of the candles have any of their flames segmented). We believe it may be caused by tuning errors or the strength of the descriptor itself.

A breakdown of all of our results can be found in Table 1. The results are interesting, firstly, the original descriptors can preform better or nearly as good as when a new descriptor was added. It also shows that some of the descriptors we chose outperform others (i.e., AGD and AEG compared to SHOT). We also found that the accuracies could be high but when visualised they did not look as good as it would be expected. This often happens when more dominant segments were correctly labeled and other segments were missed (See candelabra results in Figure 3)

The experiments results suggested that SHOT does not give a good result in this pipeline, this is perhaps because it is too discriminative for the points. Also, the good results from SC agree with the existing literature [15, 26] and our expectations.

We noticed that although some of the descriptors are very descriptive for points but they do not necessary result in accurate results for segmentation. This is probably to do with over describing the segments, which resulted in even similar ones being clustered separately.

AED consistently provides the best performance improvement. It seems that AED is better than AGD for rigid shape datasets. Although some of the recent works all opt for AGD in their pipeline, AED might in fact, be better suited for rigid shapes.

Another observation is that after looking at the results of the mean shift, in future it might be better to alter the value of the bandwidth per shape, as some shapes achieve a good initial segmentation with a much higher bandwidth than what was used.

One conclusion we can draw is that co-segmentation pipeline favours smoothly varying

Dataset	Original	+AGD	+AED	+SC	+SHOT	+D2	+AGD+AED+D2
Candelabra	238	240	265	250	252	270	291
Goblets	47	48	52	56	56	52	58
Vases	406	410	430	420	416	450	510
Guitars	163	166	179	170	168	180	219

Table 2: The execution time taken (in seconds) to run each of the tests on a corresponding set. The goblets set is far quicker than the others because it is smaller in both size and number of faces per mesh. Similarly, the vases set is slower because they have more objects in the set and, on average, more faces per mesh.

function that respect the underlying geometry. This may be explained by the use of diffusion map a variant of kernel PCA, which uses local neighbourhood information for dimension reduction. A smooth function provides a smoother embedding space, favouring clusters analysis to be carried out.

We try to use D2 histogram to independently define a new segment level descriptor. However, the results are not satisfactory and further investigation is required. For the goblets set, there was a stark improvement from using it, but for the other sets a decrease in accuracy was noted. This may have something to do with the size of the goblets set and the number of faces in each shape. These numbers are much smaller than all other sets, so the number of sampled points computed for D2 is smaller. This means the D2 histograms were low density and more likely to be similar.

Also in this work we removed one part of the process, the graph-cut refinement. In future we would like to add this stage back and examine the results further, and also reduce the level of tuning that is required for accurate results. The refinement stage also requires descriptors as input, so it would be interesting to see the effect of the new descriptors on the overall process as a whole.

We also want to investigate why the addition of D2 histograms did not improve the results in most cases. Finally we will investigate if the combination of SC and AED is viable, as the results here were promising.

6 Conclusion

In this paper we have presented, incorporated and tested the effectiveness of four new face level and five new segment level descriptors for co-segmentation of 3D shapes. This was achieved by adding the new descriptors to an existing pipeline from recent work [22]. The tests were ran as a comparison to the current descriptors that are already being used in the literature.

The results showed that even if a descriptor is very good at describing a point (e.g., SHOT), it doesn't necessarily make it a good descriptor for co-segmentation. Our results also indicate that several descriptors (e.g., AGD and SC) do not consistently provide good results contrary to some literatures. On the other hand, the new proposed descriptor (AED) has produce some promising results. Overall, we also observe that the co-segmentation process [22] favours a face level descriptor that (a) smoothly transitions between faces and (b) respects geometry and parts well.

In the future we hope to explore more face and segment level descriptors, and their combination. We will add back the graph-cut component to the pipeline, and this should then give a more complete idea of how the new descriptors impact the pipeline as a whole. We also hope to reduce the level of tuning required to get accurate co-segmentation results.

References

[1] CGAL, computational geometry algorithms library. URL http://www.cgal.org.

[2] Marco Attene, Bianca Falcidieno, and Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.

[3] Marco Attene, Sagi Katz, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, and Ayellet Tal. Mesh segmentation-a comparative study. In *Proc. IEEE Conf. on Shape Modeling and Applications*, pages 7–7, 2006.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pat. Anal. & Mach. Intell.*, 24(4):509–522, 2002.

[5] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics*, 28(3), 2009.

[6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pat. Anal. & Mach. Intell.*, 24(5):603–619, 2002.

[7] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3D mesh analysis. *ACM Trans. on Graphics*, 27(5), 2008.

[8] Aleksey Golovinskiy and Thomas Funkhouser. Consistent segmentation of 3D models. *Computers & Graphics*, 33(3):262–269, 2009.

[9] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. Conf. on Comp. Graphics and Interactive Techniques.*, pages 203–212, 2001.

[10] Qixing Huang, Vladlen Koltun, and Leonidas Guibas. Joint shape segmentation with linear programming. In *ACM Trans. on Graphics*, volume 30, page 125, 2011.

[11] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *Proc. IEEE Conf. CVPR*, pages 1943–1950, 2010.

[12] Sagi Katz, George Leifman, and Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005.

[13] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3D shapes. *ACM Trans. on Graphics*, 32(4):70, 2013.

[14] Yu-Kun Lai, Shi-Min Hu, Ralph R Martin, and Paul L Rosin. Fast mesh segmentation using random walks. In *Proc. Symp. on Solid and Phys. Modeling*, pages 183–191, 2008.

[15] Min Meng, Jiazhi Xia, Jun Luo, and Ying He. Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization. *Comput. Aided Des.*, 45(2):312–320, 2013.

[16] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. on Graphics*, 21(4):807–832, 2002.

[17] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. In *Proc. IEEE Conf. CVPR*, pages 993–1000, 2006.

[18] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. IEEE Conf. ICRA*, pages 1–4, 2011.

[19] Ariel Shamir. A survey on mesh segmentation techniques. In *Comp. Graphics Forum*, volume 27, pages 1539–1556, 2008.

[20] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.

[21] Shymon Shlafman, Ayellet Tal, and Sagi Katz. Metamorphosis of polyhedral surfaces using decomposition. *Comp. Graphics Forum*, 21(3):219–228, 2002.

[22] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *Proc. ACM SIGGRAPH ASIA*, 30(6), 2011.

[23] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proc. Euro. Conf. on Comp. Vis.*, pages 356–369. 2010.

[24] Oliver van Kaick, Kai Xu, Hao Zhang, Yanzhen Wang, Shuyang Sun, Ariel Shamir, and Daniel Cohen-Or. Co-hierarchical analysis of shape structures. *ACM Trans. on Graphics*, 32(4):69, 2013.

[25] Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *ACM Trans. on Graphics*, 31(6): 165, 2012.

[26] Zizhao Wu, Yunhai Wang, Ruyang Shou, Baoquan Chen, and Xinguo Liu. Unsupervised co-segmentation of 3D shapes via affinity aggregation spectral clustering. *Computers & Graphics*, 37(6):628–637, 2013.

[27] Youyi Zheng, Daniel Cohen-Or, Melinos Averkiou, and Niloy J Mitra. Recurring part arrangements in shape collections. In *Comp. Graphics Forum*, volume 33, pages 115–124, 2014.