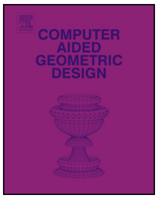
Computer Aided Geometric Design 43 (2016) 159–171

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd



Volumetric shape contexts for mesh co-segmentation

Xuanmeng Xie, Jieqing Feng *

State Key Lab of CAD&CG, Zhejiang University, China

ARTICLE INFO

Article history:
Available online 12 February 2016

Keywords:
3D shape descriptor
Volumetric shape context
Mesh segmentation

ABSTRACT

In the field of mesh segmentation, co-segmentation techniques achieve state-of-the-art performance; however, the segmentation results rely on the shape descriptors used in the segmentation process. In this paper, we propose a novel type of descriptor called the "volumetric shape context" (VSC). For each triangle in the mesh, the VSC describes the distribution of the shape's volume relative to the center of the triangle. This descriptor is descriptive, robust, and invariant under rigid transformations, uniform scaling, mirror imaging and model degeneration. We compare the VSC with state-of-the-art descriptors in a supervised mesh segmentation framework, and the results show that the VSC is most frequently selected as the first descriptor and that combining the VSC with other descriptors improves the segmentation results, thereby demonstrating the descriptiveness of the VSC.

1. Introduction

Many mesh segmentation techniques have been proposed, and these techniques can be classified into two types: part segmentation and surface-patch segmentation (Shamir, 2008). In this paper, we discuss part segmentation, in which a mesh is segmented into semantic parts. For example, a human body model can be segmented into several such parts: the head, torso, arms and legs. In recent years, researchers have realized that segmenting a set of shapes as a whole (co-segmentation) can yield better performance than segmenting individual shapes or pairs of shapes. For most co-segmentation approaches, their performance relies heavily on the 3D shape descriptors that are used. In these approaches, the triangles in the mesh are clustered into several semantic parts according to the similarity of their descriptors.

There are many descriptors that are widely used for mesh segmentation, but most are designed for solving problems in other fields, such as object recognition, and few studies have focused on the design of specialized descriptors for mesh co-segmentation. Furthermore, no descriptor is effective for all categories of models, and combinations of different descriptors can yield better performance than only one. Thus, co-segmentation approaches will benefit from the proposal and adoption of new descriptors, especially descriptors that provide significantly different information from that offered by existing descriptors.

In this paper, we propose a 3D shape descriptor for mesh segmentation named the "volumetric shape context" (VSC), which describes the volume distribution of the model around the center of each mesh triangle. It is robust to noise because of its context-based nature, and it is invariant under rigid transformations, uniform scaling, mirror imaging and model degeneration. The proposed descriptor is inspired by context analysis, which has been proven to be robust and effective in measuring the similarity between 3D shapes (Osada et al., 2002) or semantic parts (Laga et al., 2013).

* Corresponding author.
E-mail address: jqfeng@cad.zju.edu.cn (J. Feng).

http://dx.doi.org/10.1016/j.cagd.2016.02.006
0167-8396/

The VSC was evaluated in the framework of Kalogerakis et al. (2010), which is a supervised co-segmentation method that achieves state-of-the-art segmentation and labeling results. In this framework, several shape descriptors are extracted to form a descriptor vector, and JointBoost classifiers (Torralba et al., 2007) automatically select the descriptors that are most useful for a particular segmentation task. Hence, this framework is suitable for comparing the descriptiveness of 3D shape descriptors. The experimental results demonstrate that the VSC is most frequently selected as the first descriptor by JointBoost. With regard to the effect of removing a single descriptor, the labeling accuracy decreases most significantly upon the removal of the VSC, thereby indicating that the VSC represents the largest contribution to the high labeling accuracy achieved in the experiments.

The main contribution of this paper is our proposal of a new descriptor for mesh co-segmentation, which is descriptive and significantly improves the labeling accuracy. A secondary contribution is our proposal of an automatic context-based method and an interactive method to estimate the auxiliary orientations of the models. Both methods produce consistent auxiliary orientation estimation results, thereby verifying the descriptiveness of the VSC.

2. Related work

In this section, we review several 3D descriptors that are widely used in mesh segmentation methods and several mesh co-segmentation methods that take advantage of 3D descriptors.

There are many descriptors that are widely used for mesh segmentation, such as the curvature (Gal and Cohen-Or, 2006), principal component analysis (PCA) singular values, the shape diameter function (Shapira et al., 2010), the volumetric shape image (Liu et al., 2009), the average geodesic distance (Hilaga et al., 2001; Zhang et al., 2005), the shape context (Belongie et al., 2002), the spin image (Johnson and Hebert, 1999), the light field descriptor (Ding-Yun et al., 2003), the conformal factors (Ben-Chen and Gotsman, 2008), and the geodesic distance to the base of the shape (Sidi et al., 2011). Of these, the shape context and the spin image are originally 2D descriptors, and they have been adapted for extension to the 3D mesh segmentation task. These descriptors have proven to be effective for achieving good segmentation results, but none of them describes the volume distribution information of the models.

The shape context is widely used in mesh segmentation approaches and exhibits high descriptiveness. It treats a 3D model as a shell model and evaluates the distribution of the surface area; thus, the volume distribution information is not considered. Moreover, it does not construct a local 3D reference frame on each triangle; instead, it divides the neighboring space in only two dimensions, yielding insufficiently detailed distribution statistics.

The 3D shape context (Frome et al., 2004) extends the concept of the shape context to 3D point clouds and offers a solution for constructing local 3D reference frames. In this solution, the spherical support region is regarded as a globe, and the direction of the North Pole is the normal estimated at the base point. The globe is equally divided in the azimuthal and elevational dimensions and is logarithmically divided in the radial dimension. The local reference frames that are constructed in this manner are not unique, and one degree of freedom remains (the globe can rotate around its own central axis); as a result, the descriptors based on this solution are not unique and, consequently, are not suitable for mesh co-segmentation.

Many different mesh segmentation approaches have been proposed. Several surveys can be found in Agathos et al. (2007), Attene et al. (2006), Shamir (2008), Theologou et al. (2015). In recent years, researchers have realized that high-level analyses of 3D models can yield better performance than can the use of low-level geometric cues alone and thus have proposed co-segmentation approaches. Co-segmentation approaches can be classified into three categories: supervised (Kalogerakis et al., 2010; Van Kaick et al., 2011), unsupervised (Hu et al., 2012; Huang et al., 2011; Luo et al., 2013; Meng et al., 2013; Shapira et al., 2008; Sidi et al., 2011; Wang et al., 2012; Wu et al., 2014, 2013; Xu et al., 2010) and semi-supervised (Lv et al., 2012). Supervised approaches can achieve the highest labeling accuracy but require training data consisting of a set of manually segmented and labeled models, and the training phase is time-consuming. Unsupervised approaches offer lower performance than do supervised methods, which is not surprising because they use less information, but they also do not require training data and are much faster. The semi-supervised approach is a trade-off between the supervised and unsupervised approaches in that it learns from both labeled and unlabeled models. Descriptors are used in the algorithms of almost all co-segmentation approaches in these 3 categories, and their performance relies heavily on the adopted descriptors. In fact, among all of the approaches mentioned above, there was only one early study (Shapira et al., 2008) in which descriptors were not adopted, and the approaches in which descriptors are applied exhibit better performance. Supervised and semi-supervised methods use descriptors to form descriptor vectors, whereas unsupervised methods use descriptors predominantly for patch clustering and over-segmentation. These approaches will all benefit from the proposal and adoption of novel descriptors.

The proposed VSC descriptor was evaluated in the framework of Kalogerakis et al. (2010), which is a representative supervised mesh segmentation method. In this framework, the labeling of parts of the mesh is regarded as a problem of optimizing a conditional random field (CRF) (Lafferty et al., 2001). Several descriptors are extracted to form the descriptor vectors for all mesh triangles, and these descriptor vectors are used to train the basic terms of the CRF using JointBoost classifiers (Torralba et al., 2007). Different types of descriptors can be used in this framework, and adding more descriptors typically improves the performance; therefore, this framework is suitable for evaluating the descriptiveness of different descriptors.

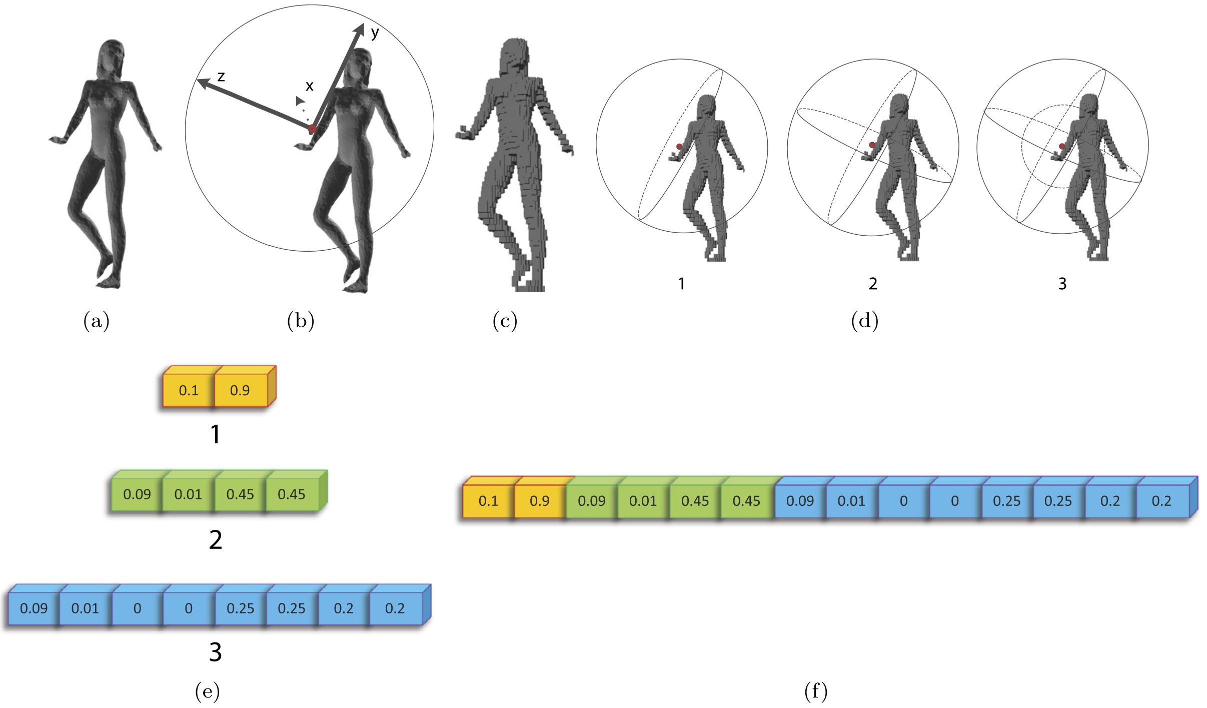


Fig. 1. Procedure for extracting the VSC. (a) A 3D mesh model. (b) For the center of a triangle on the arm as an example base point, a local 3D reference frame (LRF) is constructed at the base point. (c) The voxelization result for the model. (d) The spherical support region is divided into several bins using different bin division resolutions with reference to the LRF. (e) The normalized bin values corresponding to the 3 bin division resolutions, which describe how much of the volume of the model falls into the bins. (f) All bin values are combined to obtain the final VSC of the considered triangle on the arm.

3. Volumetric shape context

3.1. Overview

The VSC is a descriptor defined at the center of a mesh triangle (called the "base point" of the triangle) and describes the volume distribution of the model relative to the base point. The procedure for defining the VSC is illustrated in Fig. 1. First, a local reference frame (LRF) is constructed at the base point. Then, the spherical support region is divided into bins using different bin division resolutions with reference to the LRF. For each bin division resolution, the bin values are calculated as the volume of the model that falls into each bin. Finally, the bin values at the different bin division resolutions are combined to obtain the final VSC.

3.2. Local reference frame

The first step of the procedure is to construct a 3D Cartesian LRF at each base point. By introducing the LRF, the support region can be divided in 3 dimensions, thereby enhancing the descriptiveness of the proposed descriptor. We must first define two orthogonal coordinate axes, and the third is then determined by the other two coordinate axes using the left-hand or right-hand rule.

Because the VSC is defined at the center of a triangle, the outward normal to the triangle can serve as one axis of the LRF. Selecting a second axis is challenging, however, because no other intuitive choice is available. Thus, an "auxiliary orientation" is defined according to the shape of the mesh model. Note that the auxiliary orientation is estimated at the model level, not at the triangle level; therefore, it is consistent for the LRFs of all triangles in the model.

The auxiliary orientations should be defined such that they are semantically consistent among different models in the same category. For example, the auxiliary orientations of human bodies are consistent if they all point to the head or all point to the feet. To the best of our knowledge, no existing method can achieve good consistency in auxiliary orientation for all categories of models. Here, we present two methods of defining the auxiliary orientations, one of which is interactive and the other of which is automatic.

Semantic consistency can only be judged by humans; therefore, human interaction is a suitable means of defining the auxiliary orientations. The procedure for estimating the auxiliary orientation of a model is intuitive. For example, a 3D model can be dragged and rotated until the user's desired auxiliary orientation is orthogonal to the screen and points outward. In our experiments, an untrained user was able to define the auxiliary orientation of a model in less than 2 seconds on average. Thus, in practice, the interactive method is acceptable if the quantity of models is not too large.

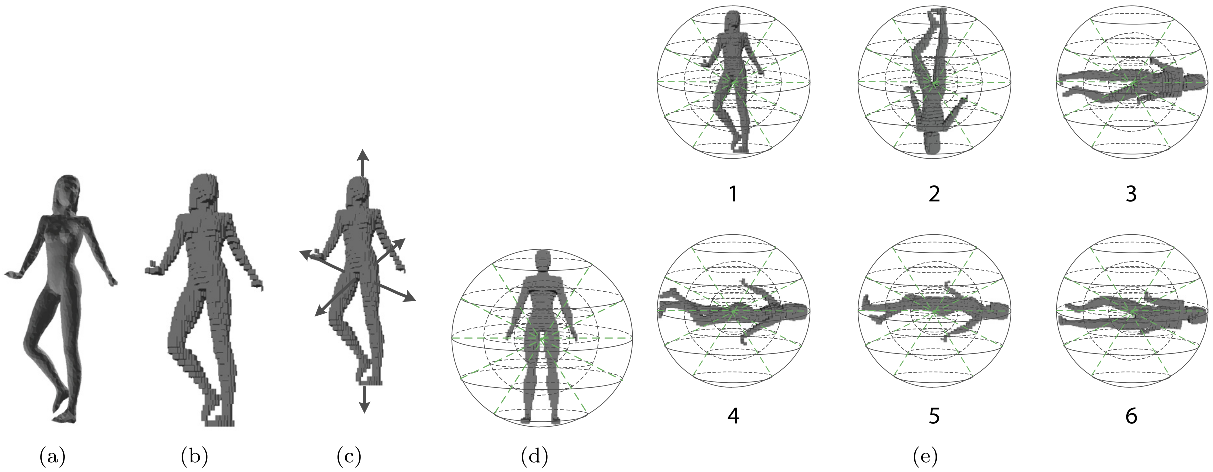


Fig. 2. Procedure for constituting the automatic context-based orientation estimation method. (a) The original model. (b) The voxelization result for the model. (c) The 6 directions defined based on (b). (d) Context C for the reference model. (e) The Context Cs corresponding to the 6 directions indicated in (c). The orientation of the model is set to the direction for which the Context C is most similar to (d).

Because the VSC is designed for automatic mesh segmentation, we also propose an automatic method of estimating the auxiliary orientations. The most popular methods for this purpose, such as CPCA (Vranić, 2004) and NPCA (Papadakis et al., 2007), are based on PCA; however, the central axes derived using PCA-based methods are not always semantically consistent among different models. PCA-based methods work well for long, thin models such as screws and fish, for which the largest eigenvalue is much greater than the other eigenvalues. Unfortunately, this condition does not hold for many other categories of models, such as planes and birds. Moreover, the axes generated using PCA-based methods do not have inherent directions; therefore, selecting consistent directions for the axes is also a challenging task.

We propose a context-based method based on PCA to select the correct axes and directions to improve the consistency of the auxiliary orientations. As illustrated in Fig. 2, the procedure is as follows:

1. A reference model is selected from each category. Models in the same category are generally similar to each other, although a few may somewhat differ from the others in shape or pose. The criterion for selecting the reference model is that it must be representative of its category. The auxiliary orientation of the reference model is specified interactively.
2. For each model in each category except the reference model, all 3 PCA axes are calculated, yielding 6 directions (each axis has 2 opposing directions, for a total of 6 directions). Then, the direction for which the context is most similar to the context of the reference model is selected. For convenience, we call the context used to select the best direction out of the 6 possible directions “Context C”.

Unlike the VSC, Context C is a global descriptor that is defined on a model with reference to a reference direction; by contrast, the VSC is a local descriptor that is defined at the center of a triangle and does not rely on any reference direction. The definition of Context C is as follows. The reference direction determines the North Pole direction, and the bounding sphere of the model is divided into M and N segments in the radial and latitudinal dimensions, respectively, as shown in Fig. 3. Then, we voxelize the model and count how many voxels fall into each bin. The resulting vector of bin values is precisely the Context C of the model with reference to the reference direction. Because determining the VSC also requires voxelization, the algorithm for determining Context C can share the voxelization result calculated for determining the VSC to save computation time and memory space.

Let $\boldsymbol{v}_r$ be the Context C of the reference model r, and let $\boldsymbol{v}_{ij}$ be the Context C of another model i corresponding to reference direction j. The difference between $\boldsymbol{v}_{ij}$ and $\boldsymbol{v}_r$ is measured as follows:

$$D_{ij} = \frac{\sum_{t=1}^{M\times N} \left|\boldsymbol{v}_{ij}[t] - \boldsymbol{v}_r[t]\right|}{M \times N}$$

Here, $\boldsymbol{v}_{ij}[t]$ is the tth element of vector $\boldsymbol{v}_{ij}$, and $\boldsymbol{v}_r[t]$ is the tth element of vector $\boldsymbol{v}_r$. The bins related to the Context C can be organized in any order, as long as the orders are consistent among different Context Cs, because each bin value is treated equally in the calculation of the difference.

From among the 6 possible directions, j^*, which minimizes D_{ij}, is selected as the auxiliary orientation of the model.

The auxiliary orientation is probably not orthogonal to the normal to the considered mesh triangle; therefore, it cannot directly serve as the second axis. Instead, we select the orientation of the cross product of the normal and the auxiliary orientation as the second axis. If the normal and the auxiliary orientation are approximately collinear, thereby invalidating

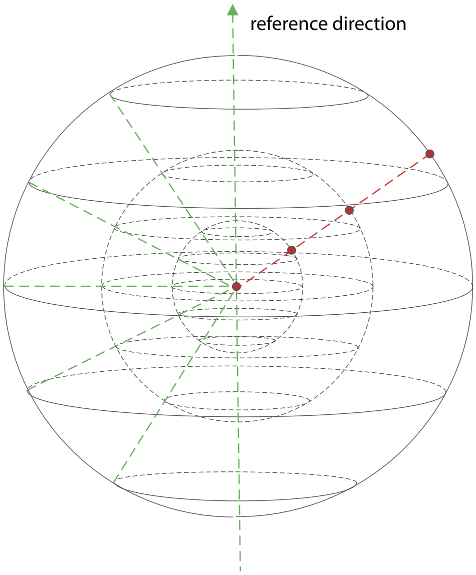


Fig. 3. Bin division scheme for Context C. In this example, the radius is evenly divided into 3 segments, as shown by the red lines and red points, and the 180 degrees of the latitudinal dimension are evenly divided into 6 segments, as shown by the green lines. As a result, the bounding sphere is divided into 18 bins. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

this method of determining the second axis, we simply replace the auxiliary orientation with a random orientation that is perpendicular to the original auxiliary orientation to obtain a viable second axis. Fortunately, this situation rarely arises, and experimental results indicate that it has little influence on the descriptiveness of the VSC.

Once two orthogonal axes have been obtained, the third axis can be straightforwardly determined, yielding the LRF. We use $x+$, $y+$ and $z+$ to represent the positive directions of the three axes of the LRF. The normal to the triangle serves as $z+$, the cross product of the normal and the auxiliary orientation serves as $x+$, and the cross product of $x+$ and $z+$ serves as $y+$.

3.3. Bin division

Before the bin division procedure is applied, the support region for the VSC of each triangle must be specified. Different shapes of support regions could be considered, such as a spherical one or a cubic one, and the one we adopt in this paper is a solid sphere whose center coincides with the center of the triangle. The radius of the sphere is related to the size of the model and will be discussed at the end of this section.

As described in Section 3.2, a unique LRF is defined at the center of each triangle. By regarding the support region as analogous to a globe, with the $z+$ direction corresponding to the direction of the North Pole and the $x+$ direction corresponding to a longitude of 0, the support region can be divided along three dimensions — the radial, longitudinal and latitudinal dimensions — as shown in Fig. 4.

The radial dimension is divided logarithmically, such that the region near the origin of the LRF contains a larger number of bins. Because the values of the bins near the origin are more closely related to the base point (the center of a triangle), this approach ensures that more information is collected nearer to the underlying triangle, thereby improving the descriptiveness of the descriptor. Suppose that the support region is logarithmically divided into M segments in the radial dimension. Let r_i represent the radius of the ith smallest boundary sphere, where i is an index that takes values from 1 to M. r_i can be expressed in the following form:

$$r_i = a \times b^{i-1}$$

Here, a and b are two unknown variables. Let r_{max} be the radius of the solid spherical support region. Given only M and r_{max}, the division of the radial dimension is not unique; thus, r_{min} is defined as the radius of the minimum boundary sphere. We then have

$$r_{min} = r_1 = a \times b^0 = a$$
$$r_{max} = r_M = a \times b^{M-1}$$

Upon eliminating a and b, the radius of any boundary sphere can be expressed in terms of M, r_{max} and r_{min}:

$$r_i = r_{max}^{(i-1)/(M-1)} \times r_{min}^{(M-i)/(M-1)}$$

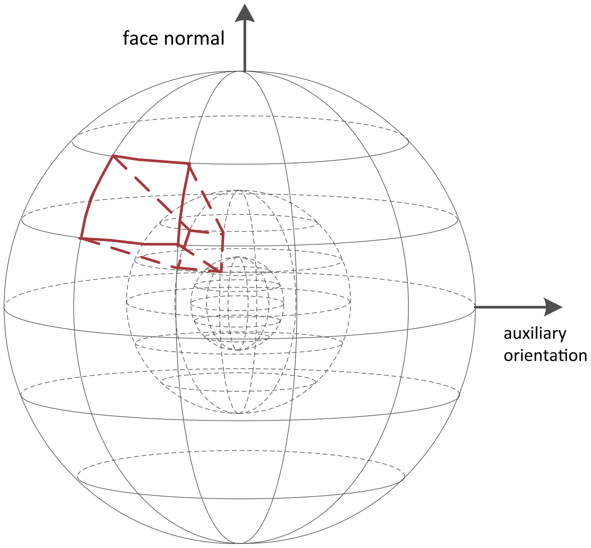


Fig. 4. Bin division scheme for the VSC. The support region, a solid sphere, is divided along 3 dimensions — the radial dimension, the longitudinal dimension and the latitudinal dimension — in analogy to a globe. The face normal corresponds to the direction of the North Pole, and the cross product of the face normal and the auxiliary orientation corresponds to a longitude of 0 degrees. The region demarcated by red lines is an example of a bin. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The support region is evenly divided into N segments from 0 to 2π in the longitudinal dimension and evenly divided into P segments from 0 to π in the latitudinal dimension.

The bins can be organized in any order, as long as the orders are consistent among different VSCs; that is to say, bins in similar positions with respect to their corresponding LRFs should have the same order.

The radius of the support region should be comparable to the radius of the bounding sphere of the model for the following reasons. Obviously, the descriptiveness of the VSC would be enhanced in the case of a larger radius. The disadvantage of a larger radius is that more computation time and more memory space are required. Fortunately, however, the logarithmic division schema ensures that only a few bins will be added as the radius is increased. If the radius is too small, the VSC will tend to be local, and its resistance to noise will be reduced. However, once the support region is sufficiently large that it already contains the entire model, it is meaningless to further enlarge the radius. In summary, it is recommended that the radius of the support region be comparable to the radius of the bounding sphere of the model.

In our implementation, the radius of the support region (i.e., r_{max}) is set equal to the radius of the bounding sphere of the model, and r_{min} is set to $r_{max}/100$.

3.4. Bin value calculation

As is described in previous sections, the VSC of a mesh triangle describes the volume distribution of the model relative to the base point of the constructed LRF (the center of that triangle), and the value of each bin represents the portion of the volume of the model that falls into that bin. Because the volume of the model is difficult to calculate, voxelization is applied to simplify the calculation. If the automatic context-based auxiliary orientation estimation method is used, the voxelization results can be shared with the auxiliary orientation estimation procedure.

The procedure for calculating the bin values is as follows. First, we construct a bounding box for the model and divide this bounding box into K^3 voxels. K controls the voxelization resolution of the model, and the proper selection of the value of K will be discussed in Section 4.2. Then, we remove the voxels whose centers are located outside the model and count how many of the remaining voxels' centers are located inside each bin. Finally, we normalize the bin values by dividing them by the sum of all bin values. In fact, we combine the results for different bin division resolutions to enhance the descriptiveness; each bin division resolution yields a corresponding vector of bin values, and the vectors of bin values for different bin division resolutions are combined to obtain the final VSC vector of the triangle, as shown in Fig. 1.

4. Results

For simplicity, in Table 1, we define abbreviations for the descriptors discussed in this section. Additionally, the VSCs calculated using the two different auxiliary orientation estimation methods, i.e., the interactive and automatic methods, are referred to as the "interactive VSC" and the "automatic VSC", respectively.

Table 1
Abbreviations of descriptors.

Descriptor	Abbreviation	References
Volumetric shape context	VSC	
Curvature	CUR	Gal and Cohen-Or (2006)
PCA singular values	PSV	
Shape context	SC	Belongie et al. (2002)
Average geodesic distance	AGD	Hilaga et al. (2001), Zhang et al. (2005)
Shape diameter function	SDF	Shapira et al. (2010)
Volumetric shape image	VSI	Liu et al. (2009)
Spin image	SI	Johnson and Hebert (1999)

4.1. Dataset and evaluation methods

We tested the VSC on Kalogerakis' dataset (Kalogerakis et al., 2010), which was drawn from the Princeton Segmentation Benchmark (Chen et al., 2009) with certain modifications to suit the supervised mesh segmentation framework. The dataset contains 19 model categories, each containing 20 mesh models. The ground truths for segmentation and labeling are also included in the dataset.

To evaluate the performance of the VSC, we employed a supervised mesh segmentation framework (Kalogerakis et al., 2010) that simultaneously segments and labels meshes. This segmentation framework was briefly introduced in Section 2, and an excellent detailed explanation is provided in Kalogerakis et al. (2010).

The labeling accuracy is the major criterion used to measure the descriptiveness of descriptors. The labeling accuracy of a single mesh can be expressed as follows:

$$A = \frac{area_{cor}}{area_{all}}$$

where $area_{cor}$ is the sum of the areas of the correctly labeled triangles and $area_{all}$ is the sum of the areas of all triangles in the mesh. The labeling accuracy of a set of meshes (e.g., the labeling accuracy of all meshes in the same category) is the average labeling accuracy of all meshes in that set:

$$A_s = \frac{\sum A_i}{n}$$

where A_i is the labeling accuracy of the mesh with index i and n is the number of meshes in the set.

Because the segmentation framework is a supervised-learning-based method, the dataset was divided into 3 subsets: the training set, the validation set and the test set. In our experiments, the 20 models in each category were evenly divided into 5 groups, each containing 2 training models, 1 validation model and 1 test model. Hence, the average labeling accuracy of the 5 test models in each category was the labeling accuracy for that category, and the average labeling accuracy of all 19 categories was the labeling accuracy for the entire dataset.

4.2. VSC parameters

The voxelization resolution and bin division resolution are important parameters of the VSC and exert considerable influence on its descriptiveness and efficiency.

The voxelization resolution for the VSC is K^3, where K is the number of voxels in each dimension. As K increases, the accuracy of the volume calculation also increases and, consequently, the descriptiveness of the VSC increases; concurrently, however, the calculation of the VSC becomes more time-consuming and requires more memory space. To achieve a compromise between descriptiveness and efficiency, we tested various values of K to determine the best one. The entire test addressed two cases: segmentation using the VSC only and segmentation using the VSC and 7 other descriptors (CUR, PSV, SC, AGD, SDF, VSI and SI). Auxiliary orientations specified via human interaction were used in both cases. For each case, we tested 14 different values of K: 2, 4, 6, 8, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Because the supervised approach is quite time-consuming, this experiment could not be performed for the entire dataset. The experiment was instead conducted on only 6 categories (human, cup, ant, chair, teddy and vase), which were selected based on their variance in model shape and complexity.

As shown in Fig. 5, in both cases, the labeling accuracy increases as K increases from 2 to 20, but it tends to remain steady as K increases from 20 to 100. Therefore, K was set to 20 in our experiments. In fact, 20 is a very small resolution parameter for voxelization; therefore, the VSC results could be rapidly extracted and consumed little memory space.

Suppose that the bin division resolution for the VSC calculation is (M,N,P), corresponding to the numbers of segments in the radial, longitudinal and latitudinal dimensions, respectively. Then, the length of the vector of bin values corresponding to this bin division resolution is $M \times N \times P$. After trying different bin division resolution options, two of them are chosen in our implementation. One is $(5, 5, 5)$, which works the best when the VSC is used alone; the other is a combination of 13 bin division resolutions — $(5, 6, 4)$, $(5, 6, 1)$, $(5, 1, 4)$, $(1, 6, 4)$, $(1, 1, 2)$, $(1, 1, 4)$, $(1, 1, 6)$, $(1, 2, 1)$, $(1, 4, 1)$, $(1, 6, 1)$, $(2, 1, 1)$, $(4, 1, 1)$ and $(6, 1, 1)$ — which works the best when the VSC is used together with the other 7 descriptors mentioned above.

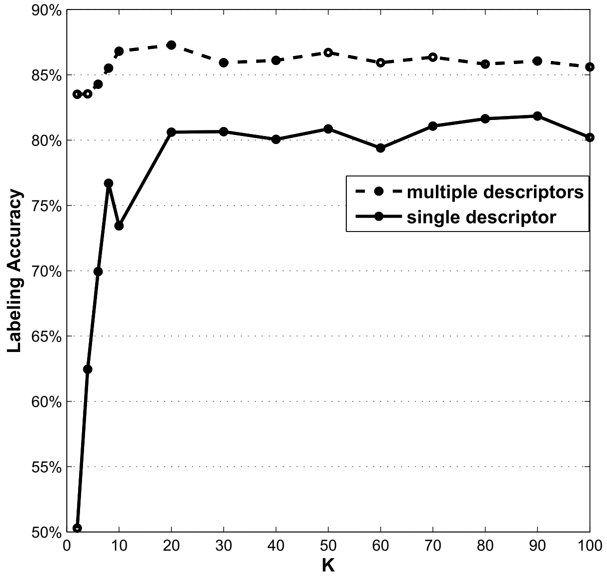


Fig. 5. The relation between the labeling accuracy and the voxelization resolution for the VSC calculation. The dotted line shows the labeling accuracy achieved using multiple descriptors, including the VSC and 7 other descriptors, whereas the solid line shows the labeling accuracy achieved using only the VSC.

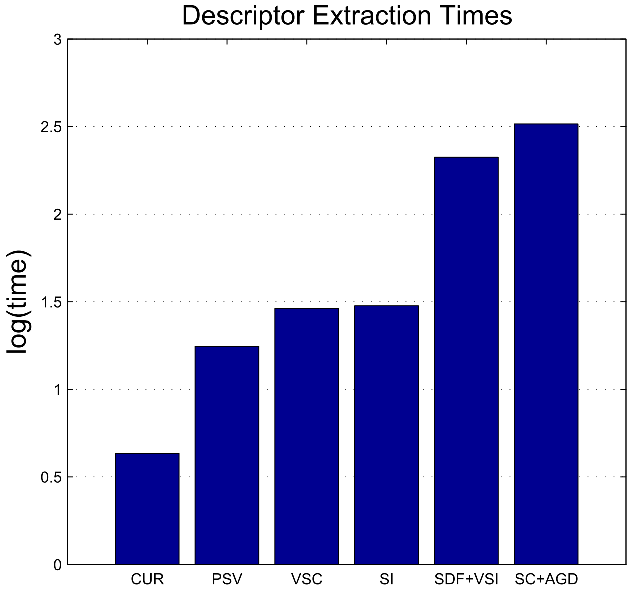


Fig. 6. Logarithmic values of the extraction times (in seconds) for the descriptors that were considered in this experiment.

4.3. Extraction time

The extraction times for the VSC and the other 7 descriptors (CUR, PSV, SC, AGD, SDF, VSI, and SI) were compared in this experiment. The same bin division resolutions used in Section 4.2 were adopted, and the voxelization resolution was set to 20. The SC and AGD were extracted together to reduce the necessary extraction time because they both rely on geodesic distances. The SDF and VSI were also extracted together because of their similarity. The experiments were performed on a PC with an Intel Core i7-4790 4 GHz CPU (only one CPU core was used) and 32 GB of RAM. The extraction times for the descriptors are shown in Fig. 6.

The automatic auxiliary orientation estimation procedure requires only an additional 0.03 seconds, so the automatic and interactive VSCs are not distinguished here. The experimental results show that approximately 29 seconds on average is required to extract the VSC. This is a much shorter time than that required for “SDF+VSI” or “SC+AGD”, primarily because an additional voxelization step is applied to speed up the extraction process.

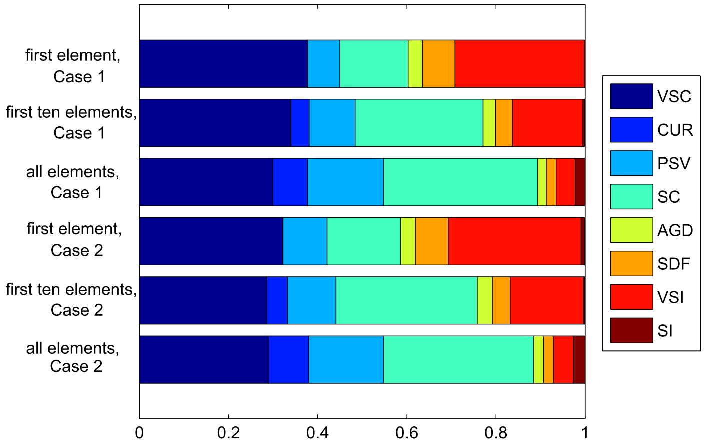


Fig. 7. Descriptor selection results. The length of each bar represents the percentage of being selected by JointBoost. The interactive VSC was used in Case 1, and the automatic VSC was used in Case 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

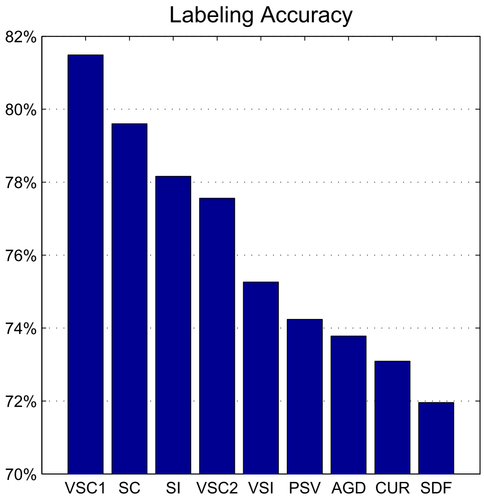


Fig. 8. Labeling accuracy achieved using a single descriptor (VSC) for mesh segmentation and labeling.

4.4. Descriptor selection

In this experiment, in each of the 95 groups (19 categories, each containing 5 groups, as described in Section 4.1), the VSC and the other 7 descriptors (CUR, PSV, SC, AGD, SDF, VSI, and SI) were adopted in the segmentation, and the descriptor vectors of these 8 descriptors were simply concatenated into a new descriptor vector. JointBoost automatically selected suitable elements one by one from the new descriptor vector. When an element is selected, we also say that the descriptor to which this element originally belongs is selected. A descriptor is considered to be more descriptive if its elements are selected earlier and more frequently by JointBoost. The experiments were performed in 2 cases: the interactive VSC was adopted in Case 1, and the automatic VSC was adopted in Case 2. We compare these 8 descriptors, according to the frequency of their elements being selected as the first element/in the first ten elements/in all the elements by JointBoost, in all the 95 groups.

The descriptor selection results are shown in Fig. 7. In terms of their frequency and order of selection, the VSC, SC and VSI are the 3 most descriptive descriptors. Moreover, regardless of whether the auxiliary orientations were estimated interactively or automatically, the VSC was most frequently selected as the first descriptor, demonstrating its descriptiveness.

4.5. Labeling results obtained using a single descriptor

Although almost all co-segmentation methods employ more than one descriptor in their approaches, we tested the labeling results obtained using a single descriptor to compare the descriptiveness of the descriptors.

As shown in Fig. 8, the VSC, SC, SI and VSI are the most descriptive descriptors. The labeling accuracy obtained using the interactive VSC is the highest. The accuracy decreases with the automatic VSC, but it remains higher than those of the other

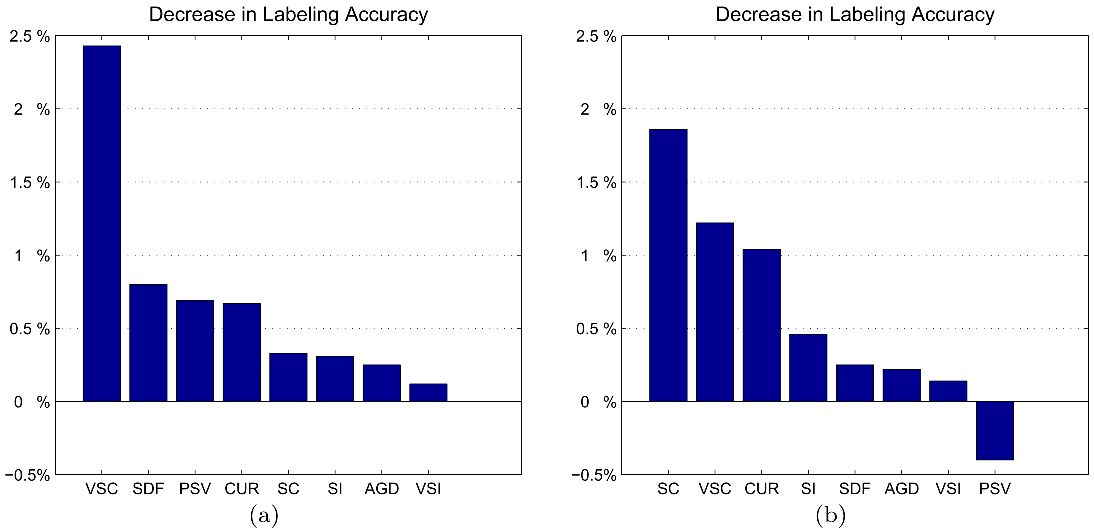


Fig. 9. Decrease in labeling accuracy upon the removal of a descriptor. The interactive VSC was considered in (a), and the automatic VSC was considered in (b). A negative value means that the labeling accuracy increases upon the removal of the corresponding descriptor.

5 descriptors. Therefore, the VSC is among the most descriptive descriptors, regardless of whether the auxiliary orientations are estimated interactively or automatically.

4.6. Labeling results obtained using multiple descriptors

In most co-segmentation approaches, multiple descriptors are used in combination; therefore, the labeling results obtained using multiple descriptors can better reflect the descriptiveness of the descriptors in practice. All 8 descriptors were initially employed, and one was then removed to test the extent to which the labeling accuracy would decrease. A descriptor was considered more useful if the labeling accuracy decreased to a greater extent upon its removal.

The interactive and automatic VSCs were evaluated separately. For the interactive VSC, the labeling accuracy obtained using all 8 descriptors was 84.15%, whereas the labeling accuracy was 82.94% for the automatic VSC. Fig. 9(a) shows that the labeling accuracy decreased by 2% upon the removal of the interactive VSC, a much larger decrease than those for the other 7 descriptors. Fig. 9(b) shows that the labeling accuracy decreased by more than 1% upon the removal of the automatic VSC, SC or CUR, thereby demonstrating the descriptiveness of them.

A descriptor is useful if it is descriptive and provides significantly different information from that offered by the existing descriptors. If a descriptor is not sufficiently descriptive, then the labeling accuracy will not markedly decrease upon its removal. If a descriptor provides information that is similar to that offered by the existing descriptors, it will be effectively replaced by the other descriptors upon its removal, and again, the labeling accuracy will not markedly decrease. Therefore, the results of this experiment prove that the VSC not only is descriptive but also provides information that is considerably difference from that offered by the existing descriptors.

4.7. Consistency of the auxiliary orientations

In this experiment, we analyzed the results of auxiliary orientation estimation obtained using 3 different methods: human interaction, the simple PCA method and the context-based PCA method. Because the simple PCA method can extract only an undirected axis for a model, we used the following simple method to assign a direction to this axis. We first defined a vector beginning at the center of the bounding sphere of the model and ending at the center of gravity of the model. Then, the axis was assigned the direction for which the angle between the directed axis and this vector was no more than $\pi/2$.

The simple PCA method was found to work well for certain categories, such as the human body category and the bearing category, but poorly for others, such as the airplane category and the teddy bear category. In the airplane category, 10 of the auxiliary orientations pointed to the head, 4 pointed to the wing and 6 pointed to the tail; therefore, there were 10 auxiliary orientations that were inconsistent with the others. In the teddy bear category, 3 of the auxiliary orientations were inconsistent with the other 17. By contrast, in the case of the context-based method, only 1 auxiliary orientation was inconsistent with the other 19 for the airplane category, and all auxiliary orientations were consistent for the teddy bear category. The auxiliary orientations obtained using the simple PCA method and the context-based method are depicted in Fig. 10.

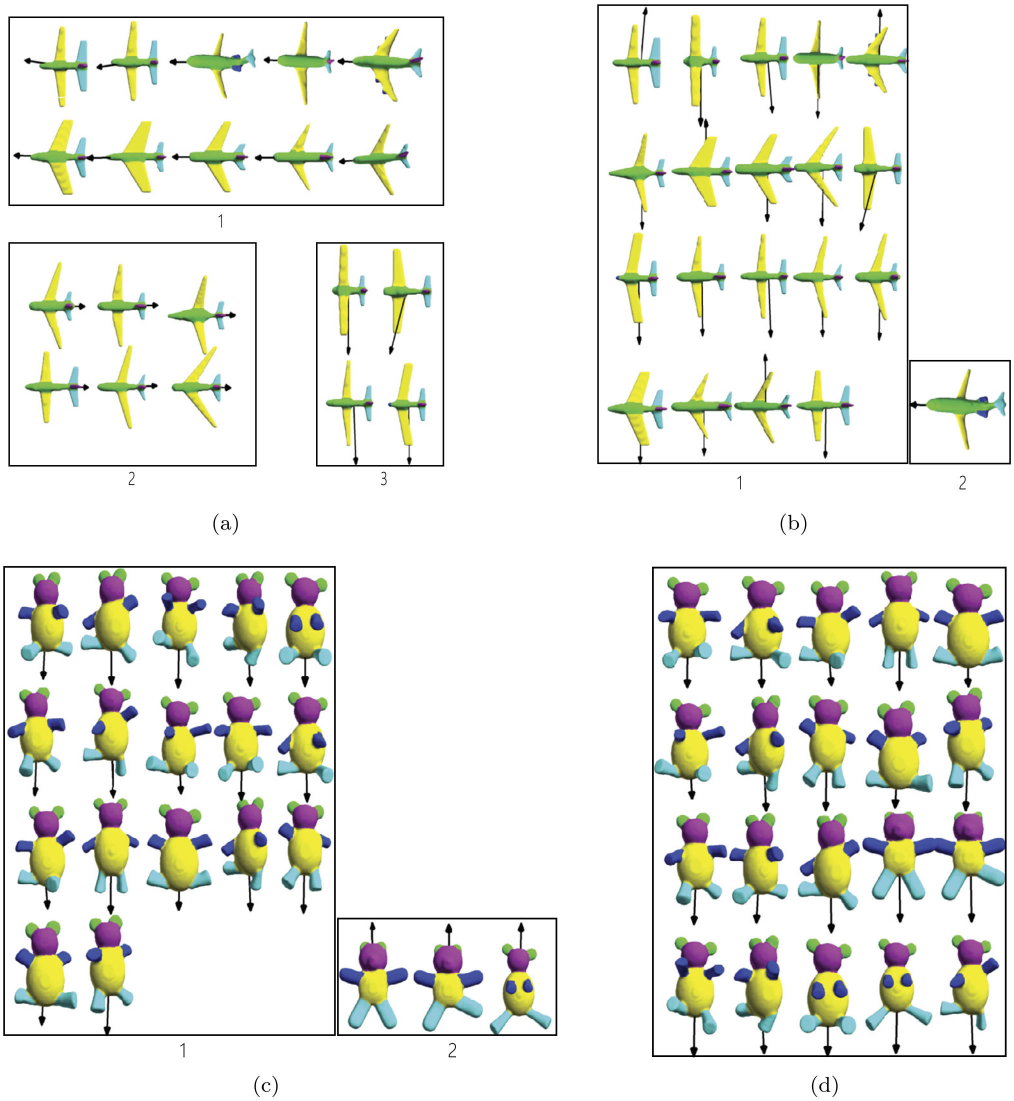


Fig. 10. Comparison of orientation consistency between the results of the simple PCA method and the context-based PCA method. In (a) and (c), the orientations were estimated using the simple PCA method; in (b) and (d), they were estimated using the context-based method. Each subfigure contains 20 models, and models with consistent orientations are grouped together. Because the models of airplanes are symmetrical, auxiliary orientations pointing to both wings are considered to be consistent. The colors show the manual ground-truth segmentations. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

The experimental results show that the context-based PCA method yields better consistency in auxiliary orientation and better labeling results than does the simple PCA method. Among these 3 auxiliary orientation estimation methods, the human interaction method exhibits the best performance.

5. Conclusions and future work

In this paper, we propose a novel 3D descriptor for mesh co-segmentation named the “volumetric shape context” (VSC), which describes the volume distribution of a mesh. The performance of this descriptor relies on the auxiliary orientations extracted from the mesh; therefore, we also propose an interactive method and an automatic context-based method of estimating the auxiliary orientations. The experimental results prove that the VSC offers state-of-the-art descriptiveness and can significantly improve the results of segmentation and labeling when it is employed in combination with other descriptors, regardless of which auxiliary orientation estimation method is applied.

Although the VSC obtained using the automatic context-based auxiliary orientation estimation method has been proven to be descriptive, the context-based method was not meticulously designed. In our future work, we plan to further improve the performance of the context-based method. For example, the symmetry of the models could be considered. At present, we select the best direction from among 6 possible directions; however, we could also design an algorithm to select the best direction from among all possible directions. Moreover, the consistency of the auxiliary orientations is still an open problem, and the VSC would benefit from the proposal of other effective automatic methods for their extraction.

The VSC is designed for use in mesh co-segmentation, including supervised, unsupervised and semi-supervised approaches. In our experiments, the VSC was successfully applied in a supervised approach, but the parameters used in our experiments may not be suitable for use in unsupervised or semi-supervised approaches. Therefore, it will be necessary to investigate what values of the parameters are most suitable in these cases.

Acknowledgements

The authors would like to thank the anonymous reviewers who gave valuable suggestions to improve the quality of the paper. This work was supported by the National Natural Science Foundation of China under Grant No. 61472349.

References

Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N., Azariadis, P., 2007. 3D mesh segmentation methodologies for CAD applications. Comput-Aided Des. Appl. 4 (1–6), 827–841. http://dx.doi.org/10.1080/16864360.2007.10738515.

Attene, M., Patané, G., Katz, S., Spagnuolo, M., Mortara, M., Tal, A., 2006. Mesh segmentation – a comparative study. In: Proceedings – IEEE International Conference on Shape Modeling and Applications 2006. SMI 2006, vol. 2006, p. 7.

Belongie, S., Malik, J., Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. 24, 509–522. http://dx.doi.org/10.1109/34.993558.

Ben-Chen, M., Gotsman, C., 2008. Characterizing shape using conformal factors. In: Eurographics 2008 Workshop on 3D Object Retrieval, pp. 1–8.

Chen, X., Golovinskiy, A., Funkhouser, T., 2009. A benchmark for 3D mesh segmentation. ACM Trans. Graph. 28 (3), 73:1–73:12. http://dx.doi.org/10.1145/1531326.1531379.

Ding-Yun, C., Xiao-Pei, T., Yu-Te, S., Ming, O., 2003. On visual similarity based 3D model retrieval. Comput. Graph. Forum 22 (3), 223–232. http://dx.doi.org/10.1111/1467-8659.00669.

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In: Computer Vision – ECCV 2004. In: Lecture Notes in Computer Science, vol. 3023. Springer, Berlin, Heidelberg, pp. 224–237.

Gal, R., Cohen-Or, D., 2006. Salient geometric features for partial shape matching and similarity. ACM Trans. Graph. 25 (1), 130–150. http://dx.doi.org/10.1145/1122501.1122507.

Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L., 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 203–212.

Hu, R., Fan, L., Liu, L., 2012. Co-segmentation of 3D shapes via subspace clustering. Comput. Graph. Forum 31 (5), 1703–1713. http://dx.doi.org/10.1111/j.1467-8659.2012.03175.x.

Huang, Q., Koltun, V., Guibas, L., 2011. Joint shape segmentation with linear programming. ACM Trans. Graph. 30 (6), 125:1–125:12. http://dx.doi.org/10.1145/2070781.2024159.

Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 433–449. http://dx.doi.org/10.1109/34.765655.

Kalogerakis, E., Hertzmann, A., Singh, K., 2010. Learning 3D mesh segmentation and labeling. ACM Trans. Graph. 29 (4), 102:1–102:12. http://dx.doi.org/10.1145/1833351.1778839.

Lafferty, J.D., McCallum, A., Pereira, F., 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289. http://dl.acm.org/citation.cfm?id=645530.655813.

Laga, H., Mortara, M., Spagnuolo, M., 2013. Geometry and context for semantic correspondences and functionality recognition in man-made 3d shapes. ACM Trans. Graph. 32 (5), 150:1–150:16. http://dx.doi.org/10.1145/2516971.2516975.

Liu, R., Zhang, H., Shamir, A., Cohen-Or, D., 2009. A part-aware surface metric for shape analysis. Comput. Graph. Forum 28 (2), 397–406. http://dx.doi.org/10.1111/j.1467-8659.2009.01379.x.

Luo, P., Wu, Z., Xia, C., Feng, L., Ma, T., 2013. Co-segmentation of 3D shapes via multi-view spectral clustering. User Model. User-Adapt. Interact. 29 (6–8), 587–597. http://dx.doi.org/10.1007/s00371-013-0824-2.

Lv, J., Chen, X., Huang, J., Bao, H., 2012. Semi-supervised mesh segmentation and labeling. Comput. Graph. Forum 31 (7), 2241–2248. http://dx.doi.org/10.1111/j.1467-8659.2012.03217.x.

Meng, M., Xia, J., Luo, J., He, Y., 2013. Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization. Comput. Aided Des. 45 (2), 312–320. http://dx.doi.org/10.1016/j.cad.2012.10.014.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. ACM Trans. Graph. 21 (4), 807–832. http://dx.doi.org/10.1145/571647.571648.

Papadakis, P., Pratikakis, I., Perantonis, S., Theoharis, T., 2007. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. Pattern Recognit. 40 (9), 2437–2452. http://dx.doi.org/10.1016/j.patcog.2006.12.026.

Shamir, A., 2008. A survey on mesh segmentation techniques. Comput. Graph. Forum 27 (6), 1539–1556. http://dx.doi.org/10.1111/j.1467-8659.2007.01103.x.

Shapira, L., Shamir, A., Cohen-Or, D., 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. Vis. Comput. 24 (4), 249–259. http://dx.doi.org/10.1007/s00371-007-0197-5.

Shapira, L., Shalom, S., Shamir, A., Cohen-Or, D., Zhang, H., 2010. Contextual part analogies in 3D objects. Int. J. Comput. Vis. 89 (2–3), 309–326. http://dx.doi.org/10.1007/s11263-009-0279-0.

Sidi, O., van Kaick, O., Kleiman, Y., Zhang, H., Cohen-Or, D., 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. ACM Trans. Graph. 30 (6), 126:1–126:10. http://dx.doi.org/10.1145/2070781.2024160.

Theologou, P., Pratikakis, I., Theoharis, T., 2015. A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation. Comput. Vis. Image Underst. 135, 49–82. http://dx.doi.org/10.1016/j.cviu.2014.12.008.

Torralba, A., Murphy, K.P., Freeman, W.T., 2007. Sharing visual features for multiclass and multiview object detection. IEEE Trans. Pattern Anal. Mach. Intell. 29 (5), 854–869. http://dx.doi.org/10.1109/TPAMI.2007.1055.

Van Kaick, O., Tagliasacchi, A., Sidi, O., Zhang, H., Cohen-Or, D., Wolf, L., Hamarneh, G., 2011. Prior knowledge for part correspondence. Comput. Graph. Forum 30 (2), 553–562. http://dx.doi.org/10.1111/j.1467-8659.2011.01893.x.

Vranić, D.V., 2004. 3d model retrieval. Ph.D. thesis. University of Leipzig, Leipzig, Germany.

Wang, Y., Asafi, S., van Kaick, O., Zhang, H., Cohen-Or, D., Chen, B., 2012. Active co-analysis of a set of shapes. ACM Trans. Graph. 31 (6), 165:1–165:10. http://dx.doi.org/10.1145/2366145.2366184.

Wu, Z., Wang, Y., Shou, R., Chen, B., Liu, X., 2013. Unsupervised co-segmentation of 3D shapes via affinity aggregation spectral clustering. Comput. Graph. 37 (6), 628–637. http://dx.doi.org/10.1016/j.cag.2013.05.015.

Wu, Z., Shou, R., Wang, Y., Liu, X., 2014. Interactive shape co-segmentation via label propagation. Comput. Graph. 38 (1), 248–254. http://dx.doi.org/10.1016/j.cag.2013.11.009.

Xu, K., Li, H., Zhang, H., Cohen-Or, D., Xiong, Y., Cheng, Z.-Q., 2010. Style-content separation by anisotropic part scales. ACM Trans. Graph. 29 (6), 184:1–184:10. http://dx.doi.org/10.1145/1882261.1866206.

Zhang, E., Mischaikow, K., Turk, G., 2005. Feature-based surface parameterization and texture mapping. ACM Trans. Graph. 24 (1), 1–27. http://dx.doi.org/10.1145/1037957.1037958.