Eurographics Symposium on Geometry Processing 2012
Eitan Grinspun and Niloy Mitra
(Guest Editors)

Volume 31 (2012), Number 5

Co-Segmentation of 3D Shapes via Subspace Clustering

Ruizhen Hu Lubin Fan Ligang Liu†

Department of Mathematics, Zhejiang University, China
State Key Laboratory of CAD&CG, Zhejiang University, China

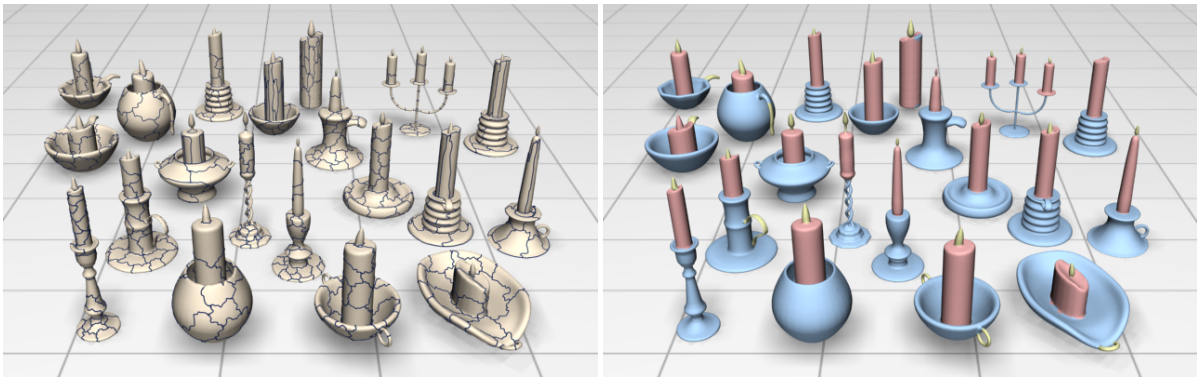


Figure 1: *The co-segmentation result of the Candelabra set by our algorithm. Starting from the over-segmented patches of the shapes (left), our algorithm automatically obtains the consistent segmentations among these objects by grouping the patches using subspace clustering in multiple feature spaces. Corresponding parts are shown in the same colors (right).*

Abstract

We present a novel algorithm for automatically co-segmenting a set of shapes from a common family into consistent parts. Starting from over-segmentations of shapes, our approach generates the segmentations by grouping the primitive patches of the shapes directly and obtains their correspondences simultaneously. The core of the algorithm is to compute an affinity matrix where each entry encodes the similarity between two patches, which is measured based on the geometric features of patches. Instead of concatenating the different features into one feature descriptor, we formulate co-segmentation into a subspace clustering problem in multiple feature spaces. Specifically, to fuse multiple features, we propose a new formulation of optimization with a consistent penalty, which facilitates both the identification of most similar patches and selection of master features for two similar patches. Therefore the affinity matrices for various features are sparsity-consistent and the similarity between a pair of patches may be determined by part of (instead of all) features. Experimental results have shown how our algorithm jointly extracts consistent parts across the collection in a good manner.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [Computer Graphics]: Segmentation—

1. Introduction

Segmentation of a 3D shape into semantic parts is a fundamental task in high-level shape analysis and processing [AKM*06,

† Corresponding: ligang.liu@gmail.com

© 2012 The Author(s)
Computer Graphics Forum © 2012 The Eurographics Association and Blackwell Publishing Ltd. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Sha08, CGF09]. In recent years, co-segmentation of a set of 3D shapes, i.e., segmentation of the shapes as a whole into consistent semantic parts with correspondences, has received increased attention [GF09, XLZ*10, HKG11, SvKK*11]. It has been demonstrated that more knowledge can be inferred from multiple shapes rather than an individual shape and co-analysis of the set produces better segmentations than the single-shape algorithms [GF09,vKZHCO10,HKG11, SvKK*11]. However, extraction of appropriate knowledge inherent to multiple shapes for consistent segmentation remains challenging.

In this paper, we propose a novel unsupervised framework to consistently segment a set of 3D shapes from the same class. We consider the co-segmentation as a clustering problem by grouping the primitive patches of all shapes into corresponding parts (see Figure 1). The core solution paradigm of our algorithm is a subspace clustering optimization [Vid10] on the patches.

Multiple feature descriptors, which respect shape geometry and context, are used to measure similarity of patches. According to our observation, two parts of models perceived as corresponding may *not* necessarily be similar in *all* features and may even significantly differ on some. Figure 2 shows an example of two table models, of which the legs are semantically in correspondence. Their averaged geodesic distance (AGD) features [HSKK01] are similar, while their shape diameter function (SDF) features [SSS*10] are quite different, as shown in the colormaps in Figure 2 (right). Hence, concatenating the features into a higher dimensional descriptor may confuse clustering algorithms by augmenting the difference between two similar parts, as shown in Figure 2 (lower left).

Inspired by recent works in image processing [Vid10, CLW*11], we propose a new subspace clustering formulation of optimization with a consistent multi-feature penalty to guarantee the consistency of co-segmentation results according to various features. By solving this optimization we identify the most similar patch pairs consistent within multiple feature spaces and the prominent features contributing to measuring the similarity of each patch pair. Figure 2 (upper left) shows the co-segmentation results with the consistent multi-feature penalty .

To the best of our knowledge, we are the first to explore the fusion of multiple features in shape segmentation and geometry processing to this extent. It is worthwhile pointing out that popping-up of the prominent features which are used to measure the similarity between patches is different with the feature selection scheme in the learning based approach [KHS10] where features are selected by JointBoost in order to classify each training face into its corresponding label. As an unsupervised method, our approach does not require training data. Instead of selecting the features to satisfy some conditions, we allow the features which contribute most to similarity between two patches to pop up by themselves in the results.

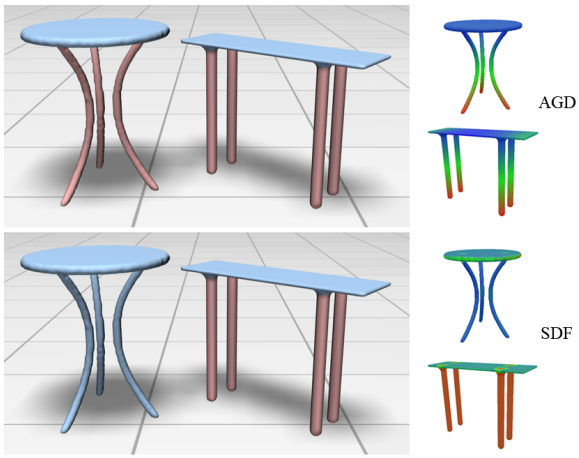


Figure 2: *The legs of two table models are semantically in correspondence (upper left). They are quite similar in AGD features (upper right) while they differ a lot in SDF features (lower right). Hence, two parts of models perceived as corresponding may not necessarily be similar in all features. Upper left: result with the consistent multi-feature penalty; Lower left: result with the concatenated feature descriptor.*

We evaluate our proposed approach on various 3D shape categories and make comparisons with state-of-the-art approaches. These results demonstrate that our approach achieves comparable performance to the supervised approach and produces better results than the others. Moreover, our approach is more efficient than the previous approaches as the convex optimization we used can be effectively solved. Besides, as a patch-level approach, it takes much less time for our method to get the satisfactory results than any previous face-level approach and is more flexible than segment-level approaches.

Contributions. Our contributions are twofold.

- We propose a novel framework for shape co-segmentation based on subspace clustering. It simultaneously generates the segments and their correspondences from the over-segmented patches within multiple feature spaces. Various features can be introduced in our framework, which makes our algorithm flexible and feasible for co-segmenting different shapes.
- We propose a consistent multi-feature penalty in the subspace clustering optimization, which guarantees the consistency of the co-segmentation results according to various features and makes prominent features used to measure the similarity between each patch pair stand out actively.

2. Related work

A large variety of approaches have been proposed for segmenting single shape into meaningful parts [AKM*06, Sha08]. It has been shown [CGF09] that no segmentation algorithm always performs well for all models because the ge-

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

ometry of an individual shape may lack sufficient cues to identify all parts that would be perceived as meaningful to a human observer.

Co-segmentation of 3D shapes. There have been various recent works on consistently segmenting a set of 3D shapes from the same class into semantic parts. Kraevoy et al. [KJS07] perform an initial segmentation of each model into parts and then create a consistent segmentation by matching the parts and finding their correspondences. Huang et al. [HKG11] present a linear programming based approach for jointly segmenting a heterogenous database of shapes, which significantly outperforms single-shape segmentation techniques. However, these two methods provide only mutually consistent segmentation but cannot guarantee the consistency of the final segmentations across the set.

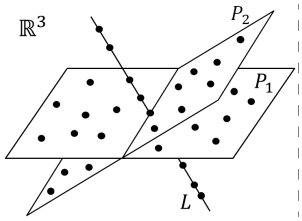
Golovinskiy and Funkhouser [GF09] consider the consistent segmentation as a graph clustering problem. By assuming that there is a global rigid alignment between matching shapes, their approach builds connection among the corresponding parts using ICP, and thus can handle limited model types only. To deal with non-homogeneous part scales, Xu et al. [XLZ*10] classify the shapes based on their styles and then establish part correspondences in each style group. However, the graph generation process is computationally expensive.

Kalogerakis et al. [KHS10] present a supervised learning based technique that employs information from shapes in a training set to segment a given shape, demonstrating significant improvement over single-shape segmentation algorithms. van Kaick et al. [vKTS*11] further incorporate prior knowledge learned from a set of pre-segmented and labeled shapes for performing part correspondences. Consistent segmentation may be established based on individual labeling of the shapes, however, a large number of manually segmented training shapes are needed in these learning based approaches, while our approach is entirely unsupervised and does not require such training data.

Sidi et al. [SvKK*11] propose an unsupervised approach for co-segmenting a set of shapes with large variation. The correspondences between dissimilar parts could be built via linking through third-parties. However, this approach needs initial segmentations for the shapes, which might result in unsatisfactory results when the per-shape segmentation cannot reflect the semantic parts well. On the other hand, our approach simultaneously generates the segmentations and their correspondences from the over-segmented patches, which is more flexible.

In an independent recent work, Meng et al. [MXLH12] also present an unsupervised algorithm for co-segmenting a set of similar 3D shapes by clustering the oversegmentated primitive patches and empolying the multi-label optimization to improve the results. Unlike our method, they adopt only two shape descriptors to cluster the patches which might fail in cases of dissimilar objects.

Subspace clustering. Part of our research is inspired by the recent works of subspace clustering methods [Vid10].



Subspace clustering aims to cluster the high-dimensional datasets into *multiple* low-dimensional linear subspaces simultaneously (see the figure on the right) and has been widely used in computer vision, image processing, and data regression, etc. (see [PHL04, Vid10] and references therein). We treat the co-segmentation of a set of shapes as a subspace clustering problem within multiple feature spaces by introducing a consistent multi-feature penalty in the optimization.

3. Overview

Our co-segmentation algorithm takes a set of meshes from an object category as input and produces a consistent segmentation of these meshes as output. First, we independently compute a set of primitive patches for each input mesh. Then, we calculate a few feature vectors for each patch. Finally, we perform subspace clustering on all patches in multiple feature spaces and obtain co-segmentations of these meshes.

Over-segmentation. Similar to the idea of superpixels in image segmentation [SM00, RM03] which are used to balance segmentation quality and computational cost, we perform an over-segmentation on each shape by partitioning it into primitive patches [HKG11]. We employ the normalized cuts (NCuts) [GF08] to generate the primitive patches for each shape. The number of patches per shape is set to be $p = 50$ in our implementation. See Figure 1 for two examples of the over-segmentation results.

Feature descriptors. We rely on geometric features to cluster the patches into parts among the set of shapes. Thus we prefer to choose a set of feature descriptors that is as informative as possible to distinguish patches of different parts.

We select $H = 5$ feature descriptors in our algorithm. Four descriptors, i.e., Gaussian curvature (GC) [GCO06], shape diameter function (SDF) [SSS*10], average geodesic distance (AGD) [HSKK01], and shape contexts (SC) [BMP02], are selected based on a study on feature selections in the learning approach (see Figure 5 in [KHS10]). The fifth descriptor is the conformal factor (CF) introduced in [BCG08]. All these feature descriptors are defined and computed on mesh triangles.

For each feature descriptor, we define a feature vector for each patch by computing a histogram capturing the distribution of the feature measurement on the triangles of this patch (see examples of AGD feature vectors shown in Figure 3). The number of bins for the histogram is set to be $d = 100$ in our implementation. Note that the SC feature vector of the patch is computed differently. We first compute and normalize the SC

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

descriptors by the total number of triangles in this model for each triangle and cluster all triangle descriptors into d clusters by K-means. The SC feature vector of a patch is then calculated by the histogram of the cluster category of its triangles. As a result, every patch is associated with a feature vector in each feature space $\mathbb{R}^d$.

Co-segmentation by subspace clustering. In each feature space, we treat the co-segmentation problem as that of subspace clustering, thus we are able to simultaneously generate the segmentations and their correspondences. All the patches in a single cluster represent a certain class of parts. To fuse multiple features, we propose a novel optimization formulation with a consistent multi-feature penalty. Our scheme facilitates both identification of most similar patches and identification of prominent features used for measuring the similarity between two similar patches. By considering all segments as primitives we can also perform subspace clustering on these segments to refine the co-segmentation results.

4. Subspace clustering

To make our paper self-contained, we introduce the background of subspace clustering in this section. Our co-segmentation algorithm based on subspace clustering is described in Section 5.

4.1. Problem of subspace clustering

The problem of subspace clustering aims to cluster data points into multiple subspaces and find a low-dimensional subspace fitting each cluster of points [Vid10].

Notation. Given a set of N points $\{x_i\}_{i=1}^N$, assume that they are from an unknown union of $K \geq 1$ linear subspaces (see the floating figure in Section 2). Subspace clustering aims to segment the points into K clusters such that points in each cluster lie in one of the linear subspaces.

4.2. Sparse subspace clustering (SSC)

The approach of sparse subspace clustering (SSC) [EV09] is based on the observation that each data point in a union of linear subspaces can always be represented as a linear combination of the points belonging to the same linear subspace. Thus, the combination could be sparse if the point is written as a linear combination of all other points. By searching for the sparsest combination, points lying in the same subspace can automatically be obtained, that is, x_i can be written as a sparse linear combination of all other points $x_j (j \neq i)$, where w_{ij} is the corresponding coefficient. Thus the goal is to minimize the number of nonzero coefficients (the ℓ_0 norm of $(w_{ij})_j$), i.e. $\min_{\{w_{ij}\}_j} \|(w_{ij})_j\|_0$ subject to $x_i = \sum_{j\neq i} w_{ij} x_j$ for each $i \in \{1,2,\cdots,N\}$. Unfortunately, this problem is highly non-convex and requires a combinatorial solution time. Practically it is common to replace the ℓ_0 norm by the convex ℓ_1 norm [Don06]. Thus a simpler ℓ_1 optimization problem is solved: $\min_{\{w_{ij}\}_j} ||(w_{ij})_j||_1 = \sum_{j\neq i} |w_{ij}|$ subject to $x_i = \sum_{j\neq i} w_{ij} x_j$, for each $i \in \{1,2,\cdots,N\}$. These N optimization problem can be written as a single optimization problem in $O(N^2)$ variables as $\min_{\{w_{ij}\}} \sum_{i=1}^N \sum_{j\neq i} |w_{ij}|$ subject to $x_i = \sum_{j\neq i} w_{ij} x_j$.

This optimization can be written in matrix form as

$$\min_W \|W\|_{1,1}, \ \text{s.t. } X = XW, \ \text{diag}(W) = 0 \tag{1}$$

where $W = (w_{ij})$ is an $N \times N$ coefficient matrix and $\|W\|_{1,1} = \sum_{i,j} |w_{ij}|$. The constraint $\text{diag}(W) = 0$ makes sure that W would not degenerate to the identity matrix. It is worth pointing out that the $\ell_{1,1}$ norm of matrix used in the optimization enforces the sparsity of the variables and has been well studied in the scientific community [Don06].

Generally, $X = XW$ is regarded as a soft constraint and thus the sparse coefficients are found by solving the following problem

$$\min_W \|XW - X\|_F^2 + \lambda \|W\|_{1,1}, \ \text{s.t. } \text{diag}(W) = 0 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of matrices. The term $\|W\|_{1,1}$ is seen as a penalty item in the optimization (2), which favors the sparsity of the optimal solution $\overline{W}$.

Clustering. Each entry of the matrix $\overline{W}$ measures the linear correlation between two points in the dataset. It is used to define an affinity matrix $S = (s_{ij})$ as

$$s_{ij} = |\overline{w}_{ij}| + |\overline{w}_{ji}| \tag{3}$$

The NCut method [SM00] is then applied to this affinity matrix S to segment the object set into K clusters.

4.3. Subspace clustering via quadratic programming

However, solving (2) is computationally expensive when N is large. For this reason, Wang et al. [WYY*11] adopt a new penalty item $\|W^T W\|_{1,1}$ in the optimization (2) as

$$\begin{aligned} \min_W \quad & \|XW - X\|_F^2 + \lambda \|W^T W\|_{1,1} \\ \text{s.t.} \quad & W \geq 0, \ \text{diag}(W) = 0 \end{aligned} \tag{4}$$

where $\|W^T W\|_{1,1}$ also enforces sparsity of the optimal solution $\overline{W}$. The introduction of $W \geq 0$ provides better interpretations for the clustering process in practical applications. This is much faster than SSC in the practice of subspace clustering. At the mean time, it has been theoretically proven [WYY*11] that $\overline{W}$ must be block diagonal when λ is large enough, which is the desired property for an affinity matrix.

5. Algorithm

The co-segmentation problem can be formulated as follows.

Notation. Given n meshes $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_n$ from the same class, we are supposed to consistently segment each mesh into K meaningful parts. By over-segmentation, each mesh $\mathcal{M}_i$ is decomposed into p_i patches, $i = 1,2,\cdots,n$. Thus there are $N = \sum_{i=1}^n p_i$ patches in total. We have defined H feature

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

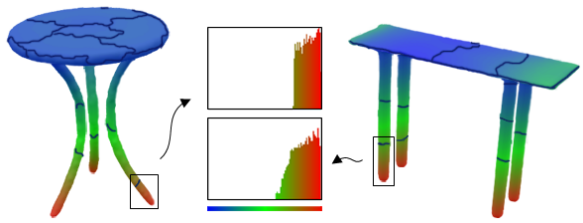


Figure 3: *Colormaps of AGD features of two tables with over-segmented patches. The AGD feature vectors of the two patches (marked in rectangles) from each table's leg have similar distribution, as shown in histograms in the middle. We see that these two feature vectors lie in a common subspace generated by standard basis corresponding to the nonzero entries.*

vectors on each patch as explained in Section 3. For the h-th feature space ($h = 1,2,\cdots,H$), denote x_{hi} as the feature vector of the i-th patch ($i = 1,2,\cdots,N$). Then we have a feature matrix $X_h = [x_{h1}, x_{h2}, \cdots, x_{hN}]$ for all patches in the h-th feature space ($h = 1,2,\cdots,H$). Our goal is to segment all patches into K clusters according to the feature matrices $X_1, X_2, \cdots, X_H$, where each cluster defines the corresponding parts of the meshes.

5.1. Single-feature co-segmentation

Consider the case of a single feature, for example, the h-th feature space ($h \in \{1,2,\cdots,H\}$). Each patch of the meshes is mapped to a feature vector in the h-th feature space $\mathbb{R}^d$. It is observed that similar patches from corresponding parts usually have similar geometric features, which results in similar distribution of the face-level feature descriptors on the triangles of these two patches. For a single patch, one face-level feature descriptor always dominates within a few scale ranges. As a result, the feature vector of a single patch may have only a few nonzero entries and thus corresponding patches are likely to be in one common subspace in the feature space. Take the table models shown in Figure 2 as an example. As discussed in Section 1, AGD can be an appropriate feature to measure similarity between these tables. We can see from Figure 3 that two corresponding patches have similar distributions (histograms), which means their feature vectors lie in the same subspace generated by standard basis corresponding to the nonzero entries. Therefore the task of segmenting the patches into parts can be considered as clustering their feature vectors $\{x_{h1}, x_{h2}, \cdots, x_{hN}\}$ in their respective subspaces.

According to (4), the optimal coefficient matrix $\overline{W}_h \in \mathbb{R}^{N\times N}$ can be computed by solving: $\min_{W_h} \mathcal{F}(W_h)$, subject to $W_h \geq 0$ and $\mathrm{diag}(W_h) = 0$, where

$$\mathcal{F}(W_h) = \|X_h W_h - X_h\|_F^2 + \lambda \|W_h^T W_h\|_{1,1}. \qquad (5)$$

After obtaining the affinity matrix as (3), we can get our co-segmentation results for this single feature by employing the NCut method [SM00].

5.2. Multi-feature co-segmentation

As mentioned in Section 1, simply concatenating multiple features into one descriptor might confuse the clustering algorithm if the corresponding patches differ in some of the features. Instead of concatenating the features, we propose a new optimization scheme for considering multiple features with some penalty among them as follows:

$$\begin{aligned} \min_{W_1,\ldots,W_H} \quad & \sum_{h=1}^{H} \mathcal{F}(W_h) + \mathcal{P}_{cons}(W_1, W_2, \cdots, W_H), \qquad (6) \\ \text{s.t.} \quad & W_h \geq 0, \\ & \mathrm{diag}(W_h) = 0, h = 1,2,\cdots,H, \end{aligned}$$

where $\mathcal{F}(W_h)$ is the objective function (5) for the h-th feature and $\mathcal{P}_{cons}$ is the penalty on the matrices $W_1, W_2, \cdots, W_H$. Notice that without the penalty $\mathcal{P}_{cons}$, the formulation (6) will reduce to a naive solution which is exactly the same as applying subspace clustering to each feature matrix X_h independently. However, this cannot guarantee the *sparsity-consistency* of their affinity matrices which means that the matrices are sparse and have their nonzero entries in the same positions. This is because that similarities of two patches measured by different features might change a lot (as shown in Figure 2), resulting in non-consistency of zero entries in their affinity matrices. Thus the *key* to address this issue is to design an appropriate penalty $\mathcal{P}_{cons}$ in the optimization.

Consistent multi-feature penalty. The penalty $\mathcal{P}_{cons}$ has to *jointly* infer a collection of affinity matrices for all the features concerned in order to induce the final consistent affinity matrix. Two things are taken into account at this point: first, we want to find the most similar patch pairs considering all features, which requires the similarities measured by different features to be consistent, i.e., their affinity matrices should be sparsity-consistent; second, two patches are considered as similar if they are similar in a subset of feature spaces and have large similarity measurement in each of these feature spaces, but need not to be similar in all feature spaces.

By considering both aspects, we introduce the following penalty:

$$\mathcal{P}_{cons}(W_1, W_2, \cdots, W_H) = \alpha \|W\|_{2,1} + \beta \|W\|_{1,1}, \qquad (7)$$

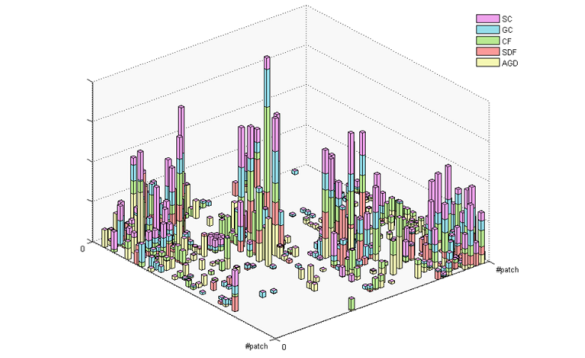
where the $H \times N^2$ matrix W is formed by concatenating $W_1, W_2, \cdots, W_H$ (each matrix in one row) together:

$$W = \begin{pmatrix} (W_1)_{11} & (W_1)_{12} & \cdots & (W_1)_{N^2} \\ (W_2)_{11} & (W_2)_{12} & \cdots & (W_2)_{N^2} \\ \vdots & \vdots & \ddots & \vdots \\ (W_H)_{11} & (W_H)_{12} & \cdots & (W_H)_{N^2} \end{pmatrix},$$

and $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$ norm defined by $\|W\|_{2,1} = \sum_{j=1}^{N^2} \|W(*,j)\|_2$, $W(*,j)$ denotes the j-th column vector of W, $\|\cdot\|_2$ is the Euclidean norm (not squared), and parameters $\alpha > 0, \beta > 0$ are used to balance the effects of the two terms.

The penalty $\mathcal{P}_{cons}$ defined in (7) plays a key role in our method. First, the $\ell_{2,1}$ penalty on W, which is the ℓ_1 penalty

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.



# nonzero components	0	1	2	3	4	5
# columns	999	292	157	62	47	43

Figure 4: *Illustration of the matrix W obtained by applying the optimization (6) on co-segmenting the two table models (Figure 2 (upper left)) into two parts. Each column of W is shown as a vertical bar with five components (shown in different colors), representing the affinities determined by the five features respectively. The numbers of columns with different nonzero components are also shown. We can see that the bars are sparsely distributed and most of the bars have less than five components. Moreover, W has a two-block diagonal structure via permutation according to the co-segmentation result.*

on the ℓ_2 norm of column vectors of W, induces column sparsity of W. As a result, some columns of W are shrunken to be entirely zero, which means that the corresponding pairs of patches will likely not be in the same cluster. Thus, this penalty facilitates the identification of the pairs of *most similar patches*, corresponding to the non-zero columns of W. Secondly, the $\ell_{1,1}$ penalty on W induces the overall sparsity of entries of W. Thus, for those non-zero columns, the sparsity of W controls the number of features entering one column, which means that a *subset* of features are used to measure the similarity between the pair of patches corresponding to this column. Hence, this penalty enables the *prominent features* to pop up for measuring the similarity between two similar patches. As a result, by combing the $\ell_{2,1}$ penalty and the $\ell_{1,1}$ penalty together, the sparsity-consistency of the matrices $W_1, W_2, \cdots, W_H$ is guaranteed.

We call the combined penalty in (7) as the *consistent multi-feature penalty*. By using this penalty in the optimization (6), our approach not only finds a subset of similar patch pairs but also finds a subset of features to measure their similarity. Note that various similar penalties were used in image segmentation and data regression [FHT10, CLW*11]. We introduce this penalty in the context of shape co-segmentation, which is very different from the previous works.

Figure 4 illustrates the matrix W (we have permuted the columns of W according to the clustering result) computed by adopting the optimization (6) for segmenting the two table models shown in Figure 2 (upper left) into corresponding parts ($K = 2$). We used all 5 features described in Section 2 in the optimization. To make the illustration clear, each table model is over-segmented into only 20 patches. Hence there are totally 40 patches and W is a 5×40^2 matrix. In this figure, each column of W is shown as a vertical bar with 5 components (shown in different colors), each of which represents the similarity measurement of the two patches corresponding to this column, according to a feature. It is seen that the non-zero bars are sparsely distributed (i.e., W has sparse non-zero columns) and many bars have less than 5 components (i.e., only the prominent features are used to measure the similarity between two patches). To make it clearer, we also show the statistics of column numbers with nonzero components in the figure. We can see that the strength of each feature can be reflected by the sum of lengths of bars with corresponding colors. In this example, the weakest descriptor is SDF (shown in red color), which confirms that SDF (shown in Figure 2) is not the proper feature used for co-segmenting the two tables as we mentioned in section 1. By permuting the columns of W according to the co-segmentation result, we can see that W is a 2-block diagonal matrix in which each block corresponds to a co-segment of the models (also see the accompanying video).

Co-segmentation. Our multi-feature co-segmentation can be solved by the minimization problem (6) where $\mathcal{F}(W_h)$ is formulated as (5) and the penalty is formulated as (7). We employ the spectral project gradient method [BMR00] to solve this minimization problem because it is simple and efficient. Denote $(\overline{W}_1, \overline{W}_2, \cdots, \overline{W}_H)$ as the optimal solution. Thus we obtain an affinity matrix $S = (s_{ij})$ as

$$s_{ij} = \frac{1}{2}\left(\sqrt{\sum_{h=1}^{H}(\overline{W}_h)_{ij}^2} + \sqrt{\sum_{h=1}^{H}(\overline{W}_h)_{ji}^2}\right). \tag{8}$$

In order to make segmentation boundaries be more semantic, we consider the minima rule [LLS*05] in the segmentation. Specifically, for each pair of adjacent patches, we define a corresponding feature value mc as the average of minimum curvature values (normalized as in [LLS*05]) of all vertices on their common boundary. Then we update the affinity value of these adjacent pairs by $s = s * (1 - mc)$. In this way, we reduce the affinity of two adjacent patches resident on both sides of a boundary with small negative curvatures. The NCut method [SM00] is then applied to obtain the segmentations of all models. Here we use the code provided by [CYS] and the only parameter, e.g. the number of clusters K, is specified by the users.

Refinement of cutting boundaries. After the co-segmentation, we adopt fuzzy cuts [KT03] to smooth the cutting boundaries of each model. Specifically, we construct the dual graph of one model and refine each boundary individually. We define the weight of the arc in corresponding parts of the dual graph as in [KT03] for each pair of adjacent parts. Then we apply graph cut algorithm to refine the cutting boundary in a fuzzy region with a sufficiently wide strip of triangles on both sides of it. In our implementation, the strip is 10 triangles wide.

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

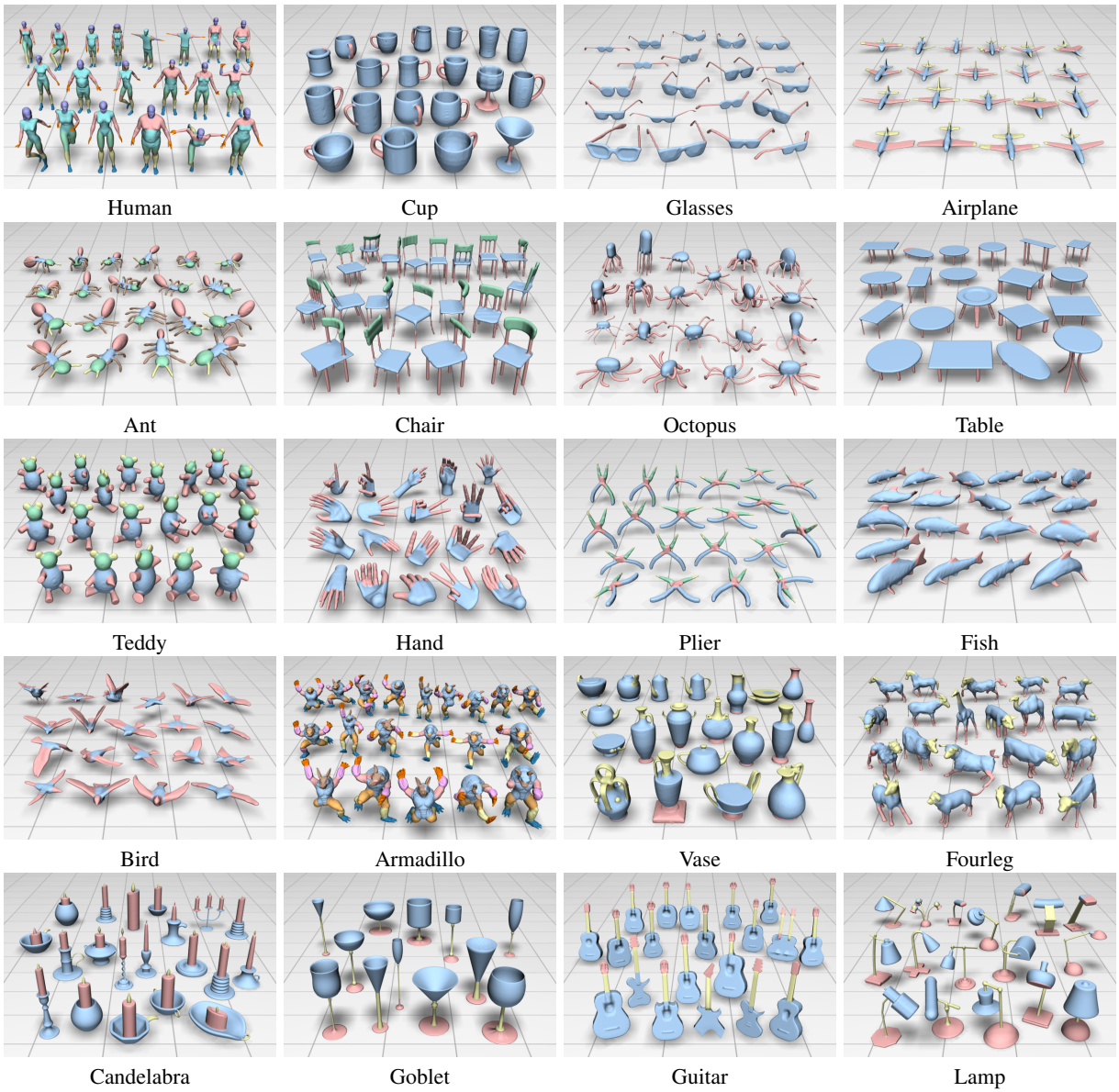


Figure 5: *Co-segmentation results on all the models of 20 categories produced by our algorithm. 16 categories are from the PSB dataset [CGF09] (the first 4 rows) and 4 categories are from the dataset used in [SvKK*11] (the last row).*

6. Results

We present experimental results and demonstrate the performance of our co-segmentation algorithm in this section.

Experimental dataset. As mentioned in [SvKK*11], there is no rigorous large-scale dataset for co-segmentation of sets of shapes yet. We construct our dataset by selecting categories of models from the Princeton Segmentation Benchmark (PSB) [CGF09] and [SvKK*11]. Specifically, our dataset consists of 20 different object categories, of which 16 categories are from the PSB dataset [CGF09] and 4 categories (Candelabra, Goblet, Guitar, and Lamp) are from [SvKK*11] (see Figure 5 and Table 1). We leave out three categories (Bearing, Mech, and Bust) from 19 categories in PSB because the models in these categories do not have meaningful correspondences between segmentations. For some models whose mesh faces are not suitable for over-segmentations we adopt the retiling method [Tur92] to remesh them.

Co-segmentation results. Our approach is entirely unsupervised. All the results obtained in this paper were produced with the fixed parameters $\alpha = 0.01, \beta = 0.1$, and $\lambda = 1000$. Figure 5 shows the co-segmentation results for all the 20 categories in our dataset (also see the accompanying video).

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

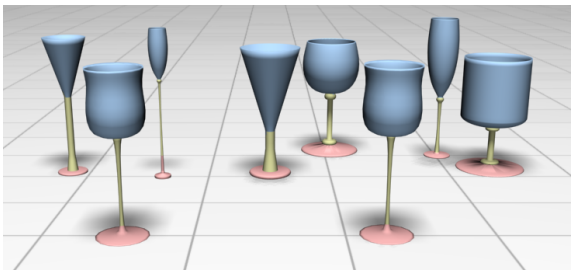


Figure 6: *The co-segmentation results by applying our algorithm on two sets of goblet models. There are 3 models in the set shown in the left and 5 models in the set shown in the right. The foremost same cup model is included in each set. By applying our algorithm on these two sets independently, we obtain the same segmentation results on this cup model.*

Our algorithm does not require the input models to have the same topologies since it creates the parts by clustering the patches. Although the models in the Cup category are topologically different, their constituent parts can still be segmented with correspondences using our algorithm. It can also be seen that our algorithm allows for outlier segments, i.e., it detects handles only on those cups that have them in the Cup category. In addition, some models in the Armadillo set do not have legs or arms and our algorithm still generates the other parts correctly.

The Ant models are in different poses and some ants have different shapes of heads and abdomens. Our algorithm correctly co-segments the different parts of these models, showing its insensitivity to poses and shape variations, which can be reinforced by the results of the Teddy category and the Candelabra category. The results of Fourleg and Guitar categories can also be confirming instances.

We can see from these examples that our algorithm is insensitive to topological changes and able to extract meaningful correspondence parts, despite the significant variety in the shape parts. Our algorithms works for a large variety of categories including man-made models, organic models, and articulated models as shown in Figure 5.

Like previous works our algorithm is inclined to generate better results if more models are given in the input. But it is not necessary to require many models and our algorithm can generate the satisfactory co-segmentation results from only a few models. Figure 6 shows the co-segmentation results of two subsets of the Goblet category including a common goblet model (the frontmost one). Our algorithm works well for these sets with few models and obtains the same segmentation results on this common model. Therefore, we can decompose the input set into a few subsets with common models and then apply our algorithm on each subset independently if the input set has a large number of models.

Evaluation. For the 4 categories selected from [SvKK*11] we use the co-segmentation results provided by the authors as ground truth. For the other 16 categories from the PSB we use the manually labeled training data [KHS10]. However, as an unsupervised one, our algorithm can hardly classify all sematic parts, e.g. the corresponding individual five fingers, thus we slightly merged parts of labels of some categories to set up our ground truth and evaluated our algorithm on how it accomplished these tasks. Specifically, we used their ground truth of 8 categories (Human, Cup, Ant, Chair, Octopus, Table, Plier and Fish) directly, and re-labeled the other 8 categories with minor variations.

Category	Ours	CFV	Category	Ours	CFV
Human	70.4	–	Plier	86.0	68.9
Cup	97.4	85.0	Fish	85.6	66.5
Glasses	98.3	97.9	Bird	71.5	71.4
Airplane	83.3	75.3	Armadillo	87.3	–
Ant	92.9	69.6	Vase	80.2	66.5
Chair	89.6	83.6	Fourleg	88.7	69.2
Octopus	97.5	95.3	Candelabra	93.9	44.2
Table	99.0	99.1	Goblet	99.2	59.8
Teddy	97.1	97.0	Guitar	98.0	90.0
Hand	91.9	88.2	Lamp	90.7	59.8
			Average	**90.4**	–

Table 1: *Statistical evaluations of the average classification accuracy of the co-segmentation results produced by our algorithm (columns of "Ours") and by the subspace clustering technique on the concatenated feature vector (columns of "CFV") on all datasets.*

To evaluate our algorithm, we use the classification accuracy criterion presented in [KHS10], which measures what percentage of the mesh's surface area is correctly labeled. We average the co-analysis labeling accuracies over all the shapes in the same category. The statistical evaluation of the co-segmentation results on these 20 categories in our dataset is shown in Table 1 (columns of "Ours"). Our algorithm has obtained an average accuracy of 90.4% over all categories.

Comparisons to the concatenated feature vector. To see the impact of our strategy of fusing multiple features more clearly, we compare against the subspace clustering technique on the concatenated feature vector, i.e., a high dimensional descriptor is constructed by concatenating the 5 feature vectors and then is used in the single-feature subspace clustering algorithm. We adopt the same parameters as we use in our algorithm.

The statistical evaluation of the co-segmentation results on all 20 categories in our dataset is also shown in Table 1 (columns of "CFV"). Notice that the average accuracies of the Human and Armadillo set are unable to compute since the models are too complex for this algorithm to obtain the correspondences properly (see result of the Armadillo set in Figure 7).

From Table 1, we can see that, on one hand, for simple sets with few number of parts, e.g. Glasses, Octopus, and Table, this algorithm can get good results as our algorithm does, which indicates that the subspace clustering is a proper choice

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

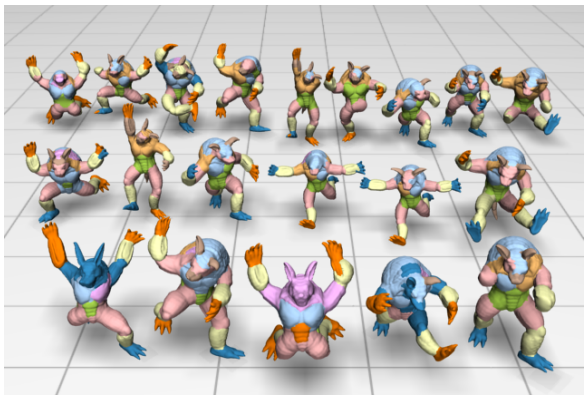


Figure 7: *Co-segmentation result of the Armadillo set via subspace clustering on the concatenated feature vector. The algorithm obtains improper correspondences and the results are much worse than those shown in Figure 5.*

for dealing with the co-segmentation problem. On the other hand, for complex set like Human and Armadillo, this algorithm fails to find the corresponding parts and cannot even segment each single model well, see Figure 7 for an example. The reason is that different features may introduce different correspondences and cause conflict here, thus introducing the consistent multi-feature penalty is necessary and useful for these types of models.

Comparisons to state-of-the-art. We have compared our approach to the supervised approach [KHS10] on 10 categories (Human, Cup, Ant, Chair, Octopus, Table, Plier, Fish, Candelabra, and Lamp) and the unsupervised approach [SvKK*11] on 5 categories (Fourleg, Candelabra, Goblet, Guitar and Lamp) based on the same ground truths provided by the authors. For the sets from PSB [KHS10], authors shared their leave-one-out-error experiment results, which are the most accurate correspondence results of their method. For the sets from [SvKK*11], accuracy was evaluated as the way described in their paper.

Figure 8 (left) shows the comparison results of classification accuracy between our approach and the learning based approach [KHS10] on 10 categories. The average of accuracies over the compared categories obtained by our method and theirs are: 90.1% and 96.1% respectively. The overall performance of our approach is a bit worse than the learning based approach. However, the supervised approach requires correctly labeled data and usually takes a couple of hours for training the classifiers. Our approach, on the other hand, exploits the natural geometric information of shapes in the largest degree, which is much more efficient.

Figure 8 (right) shows the comparison result of classification accuracy between our approach and the very recent unsupervised co-segmentation approach [SvKK*11] on 5 categories. The average accuracies are 94.4% and 88.2% respectively. It is seen that our approach gets higher correspondence results than theirs for the other 4 categories except the Lamp

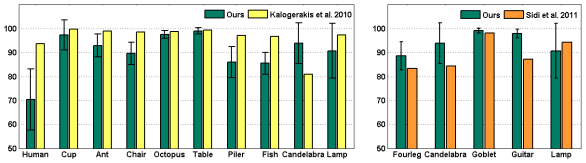


Figure 8: *Comparison of segmentation accuracies to state-of-the-art approaches (Left: [KHS10]; Right: [SvKK*11]) on various categories.*

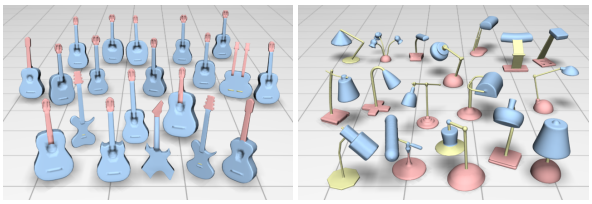


Figure 9: *Co-segmentation results of the Guitar and Lamp sets by applying the method in [SvKK*11].*

set. Figure 9 shows the co-segmentation results of the Guitar and Lamp sets by applying the method in [SvKK*11] (see the supplementary material for the comparison results on the other sets). We can see that our algorithm distinguishes headstock and neck from body of each Guitar model and builds the correspondence simultaneously while the method of Sidi et al. [SvKK*11] always confuses the classification. But some lamp models have geometrically similar base and head parts thus our algorithm incorrectly classifies them into the same cluster, such as the upper right one of the Lamp set in Figure 5. These outliers reduce the accuracy and greatly increases the variation of our approach, as shown in Figure 8 (right). However, the method of Sidi et al. [SvKK*11] also fails in some cases (e.g., the lower left one in Figure 9 (right)) while our algorithm works well.

It is worth mentioning that our algorithm directly extracts similar parts from the sets without using any third-parties, thus we can get good performance when the number of input models is really small, see Figure 6. Moreover, the method of Sidi et al. [SvKK*11] relies on good initial segmentations, thus for those organic models like Fourleg the difficulty of getting satisfactory initial segmentation results in an unsatisfactory final co-segmentation. Distinct from this segment-level approach, we process directly from patches, which allows our approach to deal with more kinds of models.

Remarks. As shown in Table 1, three categories having the lowest average accuracies are Human, Bird, and Vase, which represent three types of limitations of our algorithm. Most of all, the average accuracy of Human set is more than 20% lower than that of result obtained by the supervised approach [KHS10] (shown in Figure 8 (left)).

In the Human set, there are some cartoon models and models with inseparate hands and bodies, as shown in the right 6 models in the last row of the Human set in Figure 5. For

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

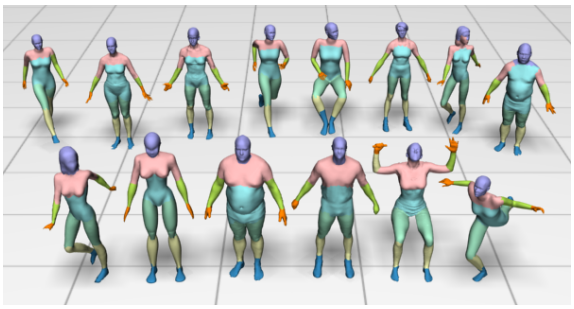


Figure 10: *Co-segmentation results of the Human set without the 6 models (cartoon models and the models with inseparate hands and bodies as shown in the right 6 models in the last row of the Human set in Figure 5). The results have more accurate segmentations and correspondences than those shown in the Human set in Figure 5.*

these models, the body proportions are different from that of a normal human, thus the features vary a lot between corresponding parts of different models, which results in the bad correspondences in the co-segmentation results. To see the effect introduced by these models, we run our algorithm on the Human set without these 6 models and show the result in Figure 10. As we can see, our algorithm obtains more consistent co-segmentations with correct correspondences, and the average accuracy largely increases to 79.7%. However, the segmented parts are not as fully semantic as those in the manually labeled ground truth since we only use the geometric properties to distinguish patches and classify them into parts, thus the accuracy is still lower compared against other sets.

For the Bird models shown in Figure 11 (left), our approach classifies the wings and tails (shown in pink) into the same cluster as they share high similarities in geometry. Our algorithm cannot always distinguish two different parts with high geometric similarity.

For the Vase models shown in Figure 11 (right), we classify the handles with the tops as a cluster (shown in yellow) since the corresponding handles have low geometric similarity and cannot constitute a subspace themselves. Our algorithm cannot always recognize corresponding parts with low geometric similarity.

Performance. All experiments were performed on an Intel(R) dual-core 2.93Hz CPU with 4GB RAM. The average running time using our algorithm on a set of 20 shapes was less than 8 minutes. Our algorithm is more efficient than both the the supervised approach [KHS10] and the method of [SvKK*11]. The reason is twofold. First, we perform the clustering on patches rather than triangles which reduces the complexity of computation. Second, the convex optimization can be effectively solved.

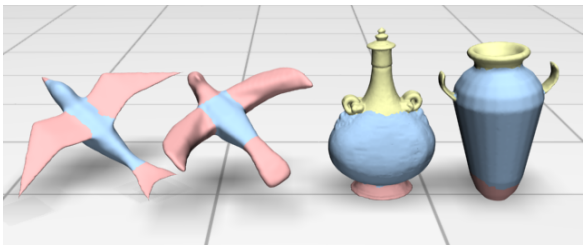


Figure 11: *Our approach may classify different semantic parts with high similarity in geometry into one cluster (left: the wings and tails (in pink) of the Bird models are classified into the same cluster as they share high similarity in geometry) and may not distinguish the semantic parts with low geometric similarity from the other clusters (right: the handles of the Vase models are clustered into the tops (in yellow) as they have low geometric similarity).*

7. Conclusion

In this work, we present an entirely unsupervised approach for consistently segmenting a set of 3D shapes from the same class. After over-segmenting the input models into primitive patches, we group the similar patches via a subspace clustering scheme. As previous works, we employ a set feature descriptors to measure the similarity between a pair of patches. However, instead of concatenating the features into a high-dimensional descriptor, we propose a consistent multi-feature penalty in the optimization to guarantee the sparsity-consistency of the affinity matrices according to various features. Our optimization enables the identification of the most similar patch pairs and of the prominent features that measure the similarity of each patch pair. Experimental results have shown how the segments can be clustered in subspace, and our algorithm efficiently extracts consistent part structure across the model set.

Limitations and future work. Our approach is purely geometry-based and thus suffers a few limitations. The success of our approach depends on the applicability of the feature descriptors. Therefore, it might not work well for complex cases, e.g., the cartoon human models and the real human models in the PSB dataset, where the corresponding semantic parts have very different geometric features.On the other hand, our approach might cluster different sematic parts with high geometric similarity into the same part and might not recognize corresponding parts with low geometric similarity, like the examples shown in Figure 11. It is feasible to incorporate other types of shape descriptors, e.g., the upright feature used in [SvKK*11] and topological structures [BGSF08] in our optimization. We will look for possible semantic level feature descriptors and incorporate them into our framework in the future.

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. We thank Oliver van Kaick and Richard Zhang for their results and dataset of [SvKK*11], Evangelos Kalogerakis and Aaron Hertzmann for their results and dataset of [KHS10], Huang Qixing for his code of over-segmentation, Shusen Wang for his code of SSQP, and Juan Sagredo for video narration. This work is supported by the National Natural Science Foundation of China (61070071) and the 973 National Key Basic Research Foundation of China (2009CB320801).

References

[AKM*06] ATTENE M., KATZ S., MORTARA M., PATANE G., SPAGNUOLO M., TAL A.: Mesh segmentation - a comparative study. In *Proc. IEEE International Conference on Shape Modeling and Applications* (2006), pp. 1–7. 1, 2

[BCG08] BEN-CHEN M., GOTSMAN C.: Characterizing shape using conformal factors. In *Proc. Eurographics Workshop on Shape Retrieval* (2008), pp. 1–8. 3

[BGSF08] BIASOTTI S., GIORGI D., SPAGNUOLO M., FALCIDIENO B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 1-3 (2008), 5–22. 10

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence 24*, 4 (2002), 509–522. 3

[BMR00] BIRGIN E., MARTÍNEZ J., RAYDAN M.: Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization 10*, 4 (2000), 1196–1211. 6

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (2009), 73:1–12. 1, 2, 7

[CLW*11] CHENG B., LIU G., WANG J., HUANG Z., YAN S.: Multi-task low-rank affinity pursuit for image segmentation. In *Proc. IEEE ICCV* (2011), pp. 1–8. 2, 6

[CYS] COUR T., YU S., SHI J.: Matlab normalized cuts segmentation code. URL: `http://www.cis.upenn.edu/~jshi/software/`. 6

[Don06] DONOHO D.: Compressed sensing. *IEEE Transactions on Information Theory 52*, 4 (2006), 1289–1306. 4

[EV09] ELHAMIFAR E., VIDAL R.: Sparse subspace clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 2790–2797. 4

[FHT10] FRIEDMAN J., HASTIE T., TIBSHIRANI R.: A note on the group lasso and a sparse group lasso. *Arxiv preprint arXiv:1001.0736* (2010). 6

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.* (2006), 130–150. 3

[GF08] GOLOVINSKIY A., FUNKHOUSER T.: Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA) 27*, 5 (2008), 145:1–10. 3

[GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3D models. *Computers and Graphics (Proc. Shape Modeling International) 33*, 3 (2009), 262–269. 2, 3

[HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint-shape segmentation with linear programming. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA) 30*, 6 (2011), 125:1–11. 2, 3

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 203–212. 2, 3

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3D mesh segmentation and labeling. *ACM Trans. on Graphics (Proc. SIGGRAPH) 29*, 3 (2010), 102:1–10. 2, 3, 8, 9, 10

[KJS07] KRAEVOY V., JULIUS D., SHEFFER A.: Model composition from interchangeable components. In *Proc. Pacific Graphics* (2007), pp. 129–138. 3

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (2003), 954–961. 6

[LLS*05] LEE Y., LEE S., SHAMIR A., COHEN-OR D., SEIDEL H.-P.: Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design* (2005), 444–465. 6

[MXLH12] MENG M., XIA J., LUO J., HE Y.: Unsupervised co-segmentation for 3dshapes using iterative multi-label optimization. *Computer-Aided Design (Proc. ACM Symposium on Solid and Physical Modeling)* (2012). 3

[PHL04] PARSONS L., HAQUE E., LIU H.: Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter 6*, 1 (2004), 90–105. 3

[RM03] REN X., MALIK J.: Learning a classification model for segmentation. In *Proc. IEEE ICCV* (2003), pp. 10–17. 3

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556. 1, 2

[SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 888–905. 3, 4, 5, 6

[SSS*10] SHAPIRA L., SHALOM S., SHAMIR A., COHEN-OR D., ZHANG H.: Contextual part analogies in 3D objects. *International Journal of Computer Vision 89*, 2-3 (2010), 309–326. 2, 3

[SvKK*11] SIDI O., VAN KAICK O., KLEIMAN Y., ZHANG H., COHEN-OR D.: Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graph. (Proc. SIGGRAPH ASIA) 30*, 6 (2011), 126:1–9. 2, 3, 7, 8, 9, 10

[Tur92] TURK G.: Re-tiling polygonal surfaces. In *Proc. of SIGGRAPH* (1992), pp. 55–64. 7

[Vid10] VIDAL R.: A tutorial on subspace clustering. *IEEE Signal Processing Magazine 28*, 2 (2010), 52–68. 2, 3, 4

[vKTS*11] VAN KAICK O., TAGLIASACCHI A., SIDI O., ZHANG H., COHEN-OR D., WOLF L., , HAMARNEH G.: Prior knowledge for part correspondence. *Computer Graphics Forum (Proc. Eurographics) 30*, 2 (2011), 553–562. 3

[vKZHCO10] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. In *Proc. of Eurographics State-of-the-art Report* (2010), pp. 1–24. 2

[WYY*11] WANG S., YUAN X., YAO T., YAN S., SHEN J.: Efficient subspace segmentation via quadratic programming. In *Proc. AAAI Artificial Intelligence* (2011), pp. 519–524. 4

[XLZ*10] XU K., LI H., ZHANG H., COHEN-OR D., XIONG Y., CHENG Z.: Style-content separation by anisotropic part scales. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA) 29*, 5 (2010), 184:1–10. 2, 3

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.