Active 3D Shape Co-segmentation with Graph Convolutional Networks

Zizhao Wu
Hangzhou Dianzi University

Ming Zeng
Xiamen University

Feiwei Qin
Hangzhou Dianzi University

Yigang Wang
Hangzhou Dianzi University

Jiří Kosinka
University of Groningen

We present a novel active learning approach for shape co-segmentation based on graph convolutional networks (GCN). The premise of our approach is to represent the collections of 3D shapes as graph-structured data, where each node in the graph corresponds to a primitive patch of an over-segmented shape, and is associated with a representation initialized by extracting features. Then, the GCN operates directly on the graph to update the representation of each node based on a layer-wise propagation rule, which aggregates information from its neighbors, and predicts the labels for unlabeled nodes. Additionally, we further suggest an active learning strategy that queries the most informative samples to extend the initial training samples of GCN, to generate more accurate predictions of our method. Our experimental results on the Shape COSEG Dataset demonstrate the effectiveness of our approach.

Segmentation of 3D shapes is a fundamental operation in geometric modeling and shape analysis. Recently, researchers have observed that by segmenting a set of 3D shapes as a whole into consistent parts one can infer more knowledge than from an individual shape. This is the problem of co-segmentation. The results of shape co-segmentation can be applied to various applications in computer graphics and vision, such as modeling, shape retrieval, texturing, etc.

Most of the existing co-segmentation approaches[1-5] cast the co-segmentation problem as the clustering problem, where one of the key ingredients is to devise robust descriptors which can characterize different parts of shapes appropriately. For example, parts belonging to the same category end up close to each other in descriptor space, while they are distant if they belong to different categories. However, since different features characterize different aspects of shapes, it is not a trivial task to directly infer the semantic information from their feature descriptors of their parts. Recently, deep learning methods, especially deep convolution neural networks (CNNs), have been widely used in various computer vision tasks with great success. Many researchers have tried to introduce deep learning techniques to facilitate various 3D shape analysis tasks. However, it is non-trivial to adapt a CNN designed for images to 3D shapes, particularly to 3D shapes modeled by irregular triangle meshes or point clouds with no regular neighborhoods like those available in images. To address this problem, many researchers have devoted their efforts to various modifications of CNNs, such as Voxel-based CNNs[6], View-based CNNs[7], and Graph-based CNNs[8], demonstrating the great potential of these methods.

Following this trend, we present a novel shape co-segmentation method based on the Graph Convolutional Network (GCN) by casting the co-segmentation problem as a graph learning problem. Specifically, we first decompose the input shapes into primitive patches, which are associated with representations initialized by extracting features. We then build a graph whose nodes are these primitive patches with feature representations, and edges are constructed by searching the nearest neighbors in feature space. Finally, we employ the GCN to dynamically update the hidden representation of each node by using a layer-wise propagation rule, and infer the class labels for all the unlabeled nodes in the network.

It is difficult to achieve (near) error-free results in the one-shot process of the GCN, particularity in the case when the initial labeled samples are sparse. To this end, we suggest an active learning technique to leverage the abundant unlabeled samples with a few labeled samples to generate a strong hypothesis. Our active learning technique is devised to query the most informative samples in the unlabeled samples to extend the initial training dataset. The query of the active learning process is raised at the end of every epoch of the GCN training process. As the process progresses, the GCN will generate more

accurate predictions as more informative labeled nodes are provided for training. By iteratively performing these improvement steps, our algorithm eventually achieves nearly error-free results.

Comparing with state-of-the-art active learning approaches[1,2] for shape co-segmentation, our approach has several advantages. First, the GCN in our approach jointly performs feature selection and classification, instead of performing these separately[1]. Second, the graph representation in our network is dynamically adjusted, rather than relying on the fixed graph structure of the label propagation approach[2], making our algorithm more efficient. Our main contribution is the introduction of the graph neural network architecture for shape co-segmentation. Further, we exploit an active learning strategy to alleviate the bottleneck of training samples. Our experimental results on a benchmark dataset demonstrate the effectiveness and utility of our approach.

RELATED WORK

In this section, we give a brief review of existing work on shape co-segmentation, CNNs for 3D shape analysis and active learning.

Shape Co-segmentation

In recent years, many techniques[1,2] have been proposed for co-segmentation of collections of 3D shapes. Their goal is to simultaneously segment a set of shapes into meaningful parts, and build correspondences among them. For example, Golovinskiy and Funkhouser[3] obtain a consistent segmentation of a set by aligning all input shapes and clustering their primitives. Sidi et al. [4] have introduced a descriptor-based method for shape co-segmentation, relying on multiple feature descriptors in order to characterize the differences among primitive patches. Wu et al. [5] presented a multiple feature fusion approach for shape co-segmentation based on affinity aggregation spectral clustering. Shu et al. [9] suggested an unsupervised approach to learn the high-level features defined on the patches based on deep learning. Generally, these approaches can achieve promising performance. However, due to their limited low-level feature space and variety, these methods often struggle with largely varying datasets.

To alleviate this problem, supervised approaches have been proposed, which treat segmentation as a labelling problem. For example, Kalogerakis et al.[10] employ traditional machine learning techniques and construct classifiers to predict class labels. Inspired by the great success of CNNs in computer vision tasks, Guo et al.[11] introduce a deep learning framework which uses convolutional neural networks to predict labels. Both of these methods are based on geometry features and carefully labeled training data. Subsequent techniques include PointNet[12], which directly takes points as input and outputs class labels. Kalogerakis et al.[7] combine image-based Fully Convolutional Networks (FCNs) and surface-based Conditional Random Fields (CRFs) to yield coherent segmentations of 3D shapes. We note that the performance of supervised techniques greatly depends on the availability of high-quality training data that covers all shape variations. However, currently, such datasets are assembled manually and are limited in scale. Additionally, the training cost of supervised approaches is expensive.

Semi-supervised learning methods can be seen as a hybrid between the above two schemes, where the classification and analysis are performed based both on the labeled data and the structure of the unlabeled data. Wang et al.[1] suggest an active learning method with the aid of constrained clustering, where the user can actively assist in the co-segmentation process by assigning pairwise constraints. However, the pairwise constraints in their algorithm are not expressive for user interaction. Wu et al.[2] introduce an optimized approach to improve interactive efficiency through label propagation. However, their method is graph-based and the graph representation is fixed during the learning process, which means that the approach heavily relies on the initial graph construction. Further, in cases with noise in the graph, extensive and tedious human labeling efforts are required. We overcome these problems by introducing a GCN-based approach. Our approach jointly performs feature selection and label prediction for co-segmentation, and the graph representation is dynamically optimized by the GCN during the feature learning procedure.

CNNs for 3D Shape Analysis

Despite its success in the setting of 2D image data, the application of CNNs to 3D shape analysis has been hindered due to that fact that 3D shapes do not possess the regular neighborhood structures present in 2D images. As a consequence, many researchers have devoted their efforts to various adaptation studies of the CNNs. Existing CNN structures used for 3D shape analysis usually fall into three types: Voxel-based CNNs[6], View-based CNNs[7], and Graph-based CNNs[8].
The voxel-based CNN[6] applied 3D convolutional neural networks on voxelized shapes for 3D shape analysis, shape retrieval and shape classification. However, the computation cost of 3D convolution is expensive. View-based CNNs[7] have tried to represent 3D shapes as rendered 2D images, and then make

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

predictions based on image CNNs. However, this method loses the geometric details of the input meshes. Graph-based CNNs were first introduced by Gori et al. [14]. Kipf and Welling[3] further extended the graph-based CNN model to the semi-supervised learning setting, and achieved very promising results in their experiments. In this paper, we also draw inspiration from the semi-supervised GCN model[3], and adapt the GCN model to facilitate shape co-segmentation.

Semi-supervised Learning and Active Learning

In many learning-based applications, training effective classifiers requires adequate labeled examples. However, labeling samples is expensive as it requires massive human efforts and expertise. To reduce or even eliminate labeling efforts, two learning strategies have been widely adopted. The first is Active Learning (AL), which aims to minimize the number of queries for building a strong learner by selecting the most informative samples for labeling from a large pool of unlabeled samples. The second is Semi-supervised Learning (SSL), which tries to exploit the availability of unlabeled data for model training and improvement.

Recent studies further investigate the combination of semi-supervised and active learning[13,14]. They have showed that their combination yields better results than either semi-supervised learning or active learning alone using the same number of labeled samples. Building on recent work on active learning [15,16], we propose an active learning strategy to boost the performance of GCNs by querying the most informative samples among the unlabeled sample set.

OVERVIEW

The purpose of our algorithm is to segment and label a set of 3D shapes into meaningful parts simultaneously. The input to our system is a set of shapes in the same category (e.g., 'airplanes') and a label of interest (e.g., 'wing'), and the output is a per-point label that indicates which part the point belongs to. In addition, our algorithm is devised to embrace supervision by the users to achieve accurate results.

The full algorithm consists of four stages; see Figure 1. First, each shape is over-segmented into primitive patches independently. Then, we extract feature vectors for each patch, followed with the k-nearest-neighbors search in feature space to construct a graph where each node denotes a primitive patch and each edge reflects a relationship between two nodes. After that, the GCN model is used to dynamically update the graph representation while making a prediction for the unlabeled nodes of the graph. Since prediction accuracy relies on the labeled samples, our method employs an active learning strategy to query the most informative samples for labeling. We iterate the last two steps till close to error-free results are achieved. Each of the stages is now described in turn.

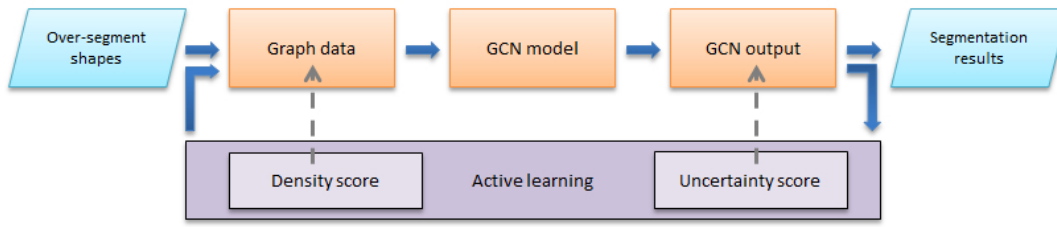


Figure 1: The pipeline of our method. First, each shape is over-segmented into primitive patches. Then, a graph is constructed where each node denotes a primitive patch. After that, the GCN model is employed to dynamically update the graph representation to make label predictions for the unlabeled nodes of the graph. In addition, we further suggest an active learning strategy to query the most informative samples for labeling in order to expand the initial training set of the GCN. We iterate the GCN and active learning until we achieve near error-free results.

Over-segmentation

Inspired by the super-pixel method in image analysis and by previous work on shape (co)segmentation [1-5], we first decompose each shape into primitive patches to generate an over-segmentation. To this end, we employ normalized cuts to compute the primitive patches for each shape. In our experiments, the number of patches per shape is set to 30. Figure 2 gives an example of our over-segmentation results.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

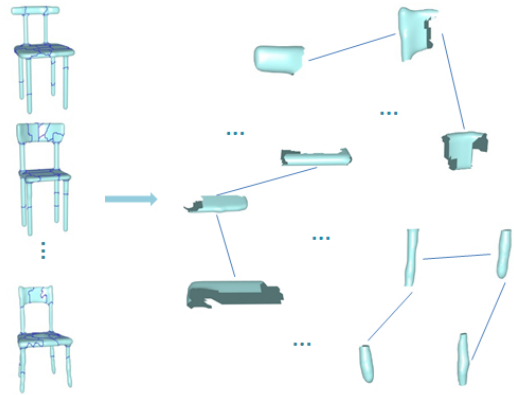


Figure 2: An example of shape over-segmentation and the graph construction. Given a category with 20 shapes, we over-segment each shape to 30 patches (left), resulting into 600 patches for the category. By extracting features defined on the patches, a graph is constructed where each node denotes a patch, and each edge represents a KNN relationship between nodes in feature space (right).

Graph Construction

Since our method is graph-based, we first need to construct a graph based on the patches. Specifically, aiming for characterizing the primitive patches of shapes, we prefer to collect a set of shape descriptors by exploiting geometric features. Following previous approaches[1-5], we select five robust and informative shape descriptors: Shape Diameter Function (SDF), Conformal Factor (CF), Shape Contexts (SC), Average Geodesic Distance (AGD), and the Geodesic distance to the Base of the shape (GB). For descriptors with multi-scale properties (such as SC), we simply sample and concatenate them. These features capture different properties of each triangle. Then, for each feature descriptor, we compute a histogram to capture its distribution for all triangles in each patch. We then accomplish our graph construction by using both the information from the feature vectors of the patches and the relationships between the patches.

Specifically, let p_i denote the i-th patch and $h_{k,i}$ be the histogram for p_i of the k-th descriptor, $\mathrm{k} = 1,2,\ldots,5$. We concatenate the histograms of each patch to form the feature vector $x_i = [h_{1,i}, h_{2,i}, \ldots, h_{5,i}]$ for the i-th patch. After that, we compute the dissimilarities between pairs of patches based on the feature vectors $\mathrm{X} = (x_1, x_2, \ldots, x_n)$ defined on the patches, where n is the total number of primitive patches. The dissimilarity between p_i and p_j is defined as $\mathrm{d}(p_i, p_j) = \mathrm{EMD}(x_i, x_j)$, where $\mathrm{EMD}(\mathrm{x}, y)$ is the earth-mover's distance between x and y. By applying the commonly used Gaussian kernel function on the distance, we obtain the similarity distance between the p_i and p_j:

$$a_{i,j} = \exp(-\mathrm{d}(p_i, p_j)/2\sigma^2). \qquad (1)$$

These constitute the elements of the similarity matrix A defined for all pairs of patches. We then optimize A through a k-nearest-neighbor search based on the assumption that the local similarities are more reliable than the more distant ones. In our implementation, we set the number of bins to be 50 for each of the histograms and σ is set to be the mean of all distances.

Our graph is constructed based on the feature set X and the similarity matrix A. Without loss of generality, let our graph be represented as $\mathrm{G}(\mathrm{V}, \mathrm{E})$ with n nodes $v_i \in \mathrm{V}$ and edges $(\mathrm{v}_i, v_j) \in \mathrm{E}$, where V is the set of nodes and E is the set of edges of G. We associate the feature set X to V, seen as node attributes, and the similarity matrix A is associated to the edge set E to complete the graph definition.

GCN for Shape Co-segmentation

Having defined the graph representation, we now introduce the GCN algorithm for semi-supervised classification of the nodes in the graph. The GCN combines graph structures and vertex features, where the features of unlabeled nodes are mixed with those of nearby labeled nodes, and propagated over the graph through multiple layers with a layer-wise propagation rule. The prediction of labels for all unlabeled nodes is also achieved in the network, which depicts the co-segmentation results for the input shapes. This is explained in detail below.

Active Learning on GCN

Since the training of the GCN still requires considerable amount of labeled data to make an accurate prediction, we propose an active learning strategy to help to extend the training samples of the network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

Our active learning method selects the most informative samples for querying, and iteratively appends them to the labeled set. Two popular active learning query criteria are adopted here: uncertainty and representativeness. These are computed by calculating the information entropy score and the information density score, respectively. We then continue to train the GCN with the expanded label set, using the pre-trained GCN as initialization.

As the process progresses, the GCN will generate more and more accurate predictions as more informative labeled nodes are provided for training. This is elaborated upon below.

GRAPH CONVOLUTIONAL NETWORK

Following the GCN definition of [8], a layer-wise propagation rule in the convolutional layers can be defined as:

$$H^{(l+1)} = \emptyset\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \qquad (2)$$

where $\tilde{A} = A + I$ is the adjacency matrix of the graph G with added self-connections via the identity matrix I, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and W^l is a layer-specific trainable weight matrix. $\emptyset(\cdot)$ denotes an activation function, which we define as the RELU function: $\mathrm{RELU}(\cdot) = \max(0,\cdot)$. $H^l \in R^{N\times D}$ is the matrix of activations in the l-th layer with the initial status $H^0 = X$ defined in the input layer.

Based on the convolutional operations, the GCN model is applied for semi-supervised classification as illustrated in Figure 3. Similarly to [8], we use a two-layer GCN which applies a softmax classifier for semi-supervised node classification on the graph:

$$Z = f(X,A) = \mathrm{softmax}\left(\hat{A}\mathrm{RELU}\left(\hat{A}XW^{(0)}\right)W^l\right), \qquad (3)$$

where $W^{(0)}$ is the input-to-hidden weight matrix, and W^l is the hidden-to-output matrix, $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, and $\mathrm{softmax}(x_i) = \exp(x_i)/z$ with $z = \sum_i \exp(x_i)$. The loss function is defined as the cross-entropy error over all labeled samples:

$$L = -\sum_{l\in y_l}\sum_{f=l}^{F} Y_{lf}\ln Z_{lf}, \qquad (4)$$

where y_l is the set of indices of labeled nodes. As the entire network is designed to be differentiable end-to-end, the parameters $W^{(0)}$ and W^l can be trained with gradient-based optimization, and the class labels for unlabeled nodes can also be obtained based on the classifier of the GCN.

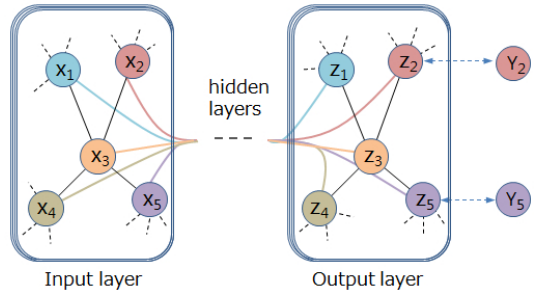


Figure 3. The Graph Convolutional Network (GCN) for semi-supervised learning. The input to the GCN is a graph with its feature representation. The GCN dynamically updates the hidden representation of each node by employing a layer-wise propagation rule, and finally infers the class labels for all unlabeled nodes in the output layer of the network.

ACTIVE LEARNING ON THE GCN

Similar to other kinds of convolutional neural networks, GCNs also need a set of labeled data for training and model selection. One way to boost the performance of a learning approach is via active learning, which selects the most informative unlabeled samples for querying. Drawing on inspiration from recent work on active learning [16,17], we chose two widely adopted active learning query strategies to actively select the samples for labeling: uncertainty sampling and density sampling (also called representativeness sampling).

Uncertainty sampling aims to choose the most uncertain instance to label. We employ information entropy as our uncertainty measure. The information entropy of a node n_i is defined as:

$$\emptyset_u(n_i) = -\sum_{c=1}^{C} P(Y_i c = 1|G,L,X)\log P(Y_i c = 1|G,L,X), \qquad (5)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

where $P(Y_i c = 1|G, L, X)$ is the probability of node n_i belonging to class c computed by the GCN. A visualization of the uncertainty score is shown on an example in Figure 4(a).

(a) Uncertainty criterion (b) Density criterion

Figure 4. Our active learning criteria: uncertainty criterion (a) and density criterion (b). The larger the embedding area of a sample, the higher its score. The squares denote the labeled samples. t-SNE [20] was applied to visualize the learned feature embedding. Our active learning method selects the most informative unlabeled samples for querying by combining the two sampling criteria.

Since uncertainty sampling may suffer from the problem of selecting outliers, density sampling is an effective solution to overcome this. The density of a sample is evaluated based on how many samples it is similar to: we compute the density score for each node, and the nodes with a high density score are selected.

To obtain the density score, we first employ the traditional cosine measure to estimate the similarity between two nodes:

$$\cos(x_i, x_j) = x_i \cdot x_j / (\|x_i\| \cdot \|x_j\|), \qquad (6)$$

where x_i and x_j are the feature vectors of nodes n_i and n_j, respectively. Practically, it would be inefficient to exhaustively calculate similarities between all pairs of samples as the unlabeled corpus is often very large. Similarity to [16], we utilize the K-Nearest-Neighbor-based density (KNN-density) to measure the density value by estimating the average similarity between each node and its k most similar nodes. Mathematically, given a set of k most similar examples $S(n_i) = \{s_1, s_2, \cdots, s_k\}$ of an unlabeled sample n_i, the average similarity, or the density value, can be computed as:

$$\emptyset_d(n_i) = \sum_{s_i \in S(n_i)} \cos(n_i, s_i) / K. \qquad (7)$$

A visualization of the density score is shown on an example in Figure 4(b).

We combine the uncertainty and density scores to form our active learning criterion, which is defined to be their product:

$$\emptyset(n_i) = \emptyset_u(n_i) * \emptyset_d(n_i). \qquad (8)$$

The motivation behind this model is to use the density score to adjust the uncertainty score of an unlabeled sample n_i. This means that our active learning model favors sample with high uncertainty and high density at each step of our learning cycle.

RESULTS AND COMPARISON

We now present the results obtained with our active learning approach, compare our method to prior art, and discuss relevant implementation details.

Results and Accuracy

In this section, we conduct experiments on the COSEG dataset [11] and the PSB dataset [19]. Both of these are widely used datasets for evaluating mesh segmentation algorithms.

As in Yi et al. [13], we also employ the amount of supervision and accuracy to evaluate our method. The accuracy is computed as:

$$Accuracy = \sum_{t \in T} a_t g_t(l_t) / \sum_{t \in T} a_t \ , \qquad (9)$$

where a_t is the area of the triangle $t \in T$, l_t is the predicted segmentation part, and $g_t(l_t)$ denotes the l_t-th component of g_t, which equals to 1 if the prediction is correct.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

We used our system to iteratively annotate the COSEG benchmark dataset. Our system selects the most informative sample for querying the annotation at each step of the iterative learning process, and the GCN model then propagates the label information to unlabeled samples while learning their high-level representations. This process proceeds until we obtain close to error-free results. Figure 5 shows an example in which the results improve as more training samples are provided.

(a) Feature embedding with 8 samples (b) Feature embedding with 16 samples.

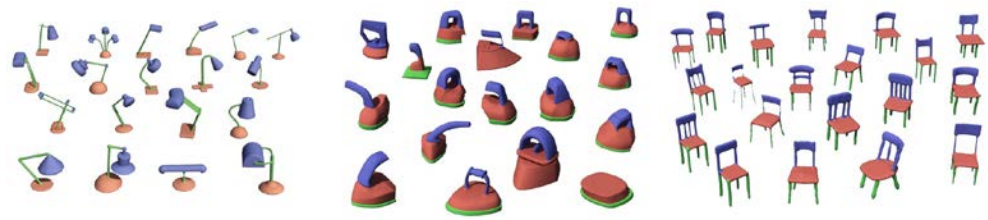
Figure 5. A visualization of the learning procedure of our GCN algorithm. Again, we adopted the t-SNE [20] approach to visualize the learned features of our graph. The squares denote the labeled samples.

Based on our extensive experiment, we report the human annotation and the verification (accuracy, precision and recall) scores of our algorithm in Table 1. As can be seen in this table, our approach is able to obtain above 95% average accuracy among several categories of shapes. Our co-segmentation results are shown in Figure 6.

Table 1. A summary of the annotation and verification scores for the varied COSEG dataset, where Num. denotes the number of shapes, Anno. denotes the number of annotations, and Acc., Pre., and Rec. stand for the accuracy, the precision, and the recall scores of the evaluation results, respectively.

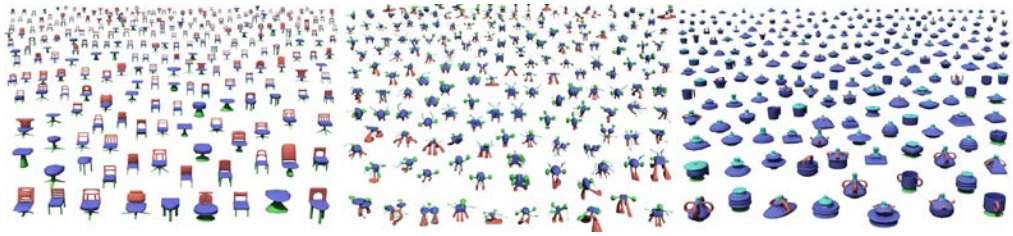
Category	Num.	Anno.	Acc.	Pre.	Rec.
Candelabra	28	21	98.73	97.72	95.67
Chairs	20	16	98.21	97.78	96.95
Goblets	12	16	98.54	98.36	97.56
Guitars	44	26	95.54	96.10	90.24
Lamps	20	23	95.68	95.34	96.21
Vases	28	18	93.35	91.68	90.46
Irons	18	25	94.64	92.35	92.85
Large chairs	400	95	95.56	92.68	89.43
Large vases	300	102	94.66	92.35	92.31
Large tele-aliens	200	97	96.64	95.88	91.52

(a) Goblets (b) Guitars (c) Candelabra



(d) Lamps (e) Irons (f) Chairs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

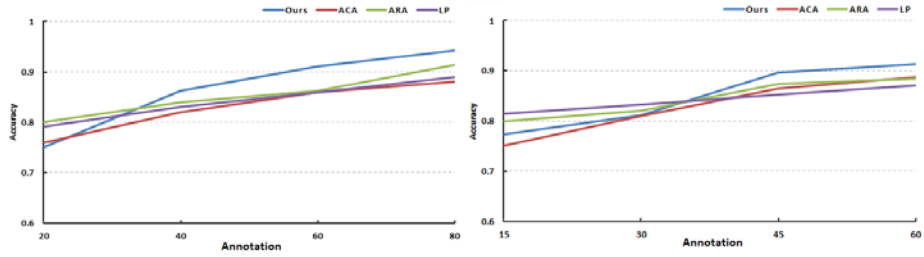


(g) Large chairs (h) Large vases (i) Large tele-aliens

Figure 6: Our co-segmentation results on several representative categories from the Shape COSEG dataset. Corresponding segments are shown in the same color.

Comparison

We compare our approach to several state-of-the-art active learning techniques. These include the active co-analysis (ACA) method proposed in [1], the label propagation (LP) method [2], and the active region annotation (ARA) method proposed in [13]. We consider four different label rates, 5%, 10%, 15%, and 20%, per class in the training set in our evaluation. We note that ACA achieves the segmentation results based on a clustering strategy with pairwise constraints, which is designed to iteratively query users to provide `must-link' and `cannot-link' relationships between over-segment patches. In contrast, our approach requires users to label directly on patches, which is more time-efficient. Our quantitative experimental results are shown in Figure 7.



(a) Large chairs (b) Large vases

Figure 7: We compare our approach to the active co-analysis (ACA) method [1], the label propagation (LP) method [2], and the active region annotation (ARA) method [13] and on the large chairs dataset and the large vases dataset. The x-axis is human annotation and the y-axis is the measured accuracy score. Note that our method generates more and more accurate results than the other methods as more training data is given.

Comparing to ARA, we note that ARA takes advantage of global shape-to-shape similarities, local feature similarities, and point-to-point correspondences in order to learn similarities between patches. Instead, due to the strong ability of representation learning of the GCN, our method requires fewer handcrafted settings, which makes our approach more general and more widely applicable. Our evaluation results suggest that our approach has superior performance over ARA on several categories of shapes, as shown in Figure 7.

We also compare our method to the semi-supervised labeling method of Wu et al. [2]. We test their method on the COSEG dataset, and used the average accuracy score to make a comparison. We note that our approach achieves superior performance than theirs; see Figure 7. This was to be expected as our method iteratively selects the best informative samples for querying rather than labeling them blindly. Our approach thus greatly reduces redundancy among labeled samples.

Finally, we compare our approach to several supervised learning algorithms [7,10,11], which range from a classical learning based approach [10] to deep learning based approaches [7,11]. For each category, N (N=6, N=12, N=19) shapes are randomly selected for training and the remaining meshes are used for testing. Theoretically, either the supervised approach or the semi-supervised approach can produce close to error-free results once sufficient training data is given. As a result, the larger the training data, the higher the accuracy score. Compared to supervised learning algorithms, our approach outperforms them on most of the categories, while requiring less training data.

Taken the N=12 case as an example, where the training data is approximately 60% of all data for a category, our method requires less than 5% (30 annotated patches over 600 patches of a category) training data while still achieving comparable results, as summarized in Table 2. Additionally, the supervised learning based approaches require several hours to train the classifier (8 hours, 4 hours, and 8 hours for [10,11], and [7], respectively). In contrast, our method spends only several seconds on training a graph with 600 nodes (or 1200 nodes for 60 patches) for 200 epochs of a category. This is a strong

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

advantage of our approach when compared to supervised learning based algorithms.

Table 2. Segmentation accuracy on the PSB Dataset, which contains 19 categories (20 models per category), summarizing a comparison of our approach to those of [10,11], and [7]. SB12 denotes 12 shapes were chosen for training, and the remaining shapes were used for testing in each category. Note that our method achieves competitive results.

	[10] SB12	[11] SB12	[7] SB12	Ours ~5%		[10] SB12	[11] SB12	[7] SB12	Ours ~5%
Human	93.20%	91.22%	94.5%	93.32%	Plier	96.20%	96.22%	95.5%	95.42%
Cup	99.60%	99.73%	93.8%	92.75%	Fish	95.60%	95.64%	96.0%	94.85%
Glasses	97.20%	97.60%	96.6%	97.11%	Bird	87.90%	88.35%	88.5%	87.41%
Airplane	96.10%	96.67%	93.0%	94.84%	Armadillo	90.10%	92.27%	92.8%	92.32%
Ant	98.80%	98.80%	98.6%	96.43%	Bust	62.10%	69.84%	68.4%	**76.12%**
Chair	98.40%	98.67%	98.5%	98.21%	Mech	90.50%	95.60%	98.7%	92.57%
Octopus	98.40%	98.79%	98.3%	96.84%	Bearing	86.60%	92.46%	92.3%	**93.67%**
Table	99.30%	99.55%	99.5%	99.26%	Vase	85.80%	89.11%	86.8%	**90.10%**
Teddy	98.10%	98.24%	97.7%	97.78%	Fourleg	86.20%	87.02%	85.0%	**89.46%**
Hand	88.70%	88.71%	84.8%	**90.14%**					

Implementation Details

We use the same hyper-parameters as in [8] to initialize the GCN model, where the initial learning rate is set to 0.01, the maximum number of epochs is set to 200, and the dropout rate is set to 0.5. We follow the work [8] to set the number of convolutional layers to 2, and the hidden layer size is set to 32.

As for computation cost, we note that the computational complexity of the GCN is linear in the number of graph edges, as discussed in [8]. We evaluated the performance of our approach on a computer with Intel Core i7 processor, 8GB of RAM, and Nvidia GeForce GTX1080. Given a graph with 600 nodes (20 shapes, 30 patches per-shape), 3600 edges, our method spends less than 0.1 seconds for each epoch. Note that the computational cost can be further sped up using FastGCN, so the scalability of our method is not an issue.

To evaluate the impact of the graph connectivity on our algorithm, we added noise to our graph by randomly changing the graph connectivity and modifying the number of edges. In our experiment, the convergence rates changed only slightly, but did not significantly affect the results. The connectivity affects the matrix A in (3), but the GCN learns W, which still ensures that the method converges. However, an unreasonable change in the graph connectivity will of course cause the GCN to fail to converge. Regarding the edge count in the graph, the GCN model converges to a better solution by taking into account longer-range information in the graph (achieved by increasing the KNN value), and the computational cost increases linearly.

It is important to emphasize that our pipeline is not very sensitive to the initial patch number of the graph construction, as demonstrated in Figure 8. In our experiment, we separately over-segmented the shapes into 30 patches and 60 patches, and also extracted feature descriptors. As can be seen in the figure, patches belonging to the same part typically have similar values, and the obtained histograms are very close to each other. In our experiment, the patch number only affected the result very slightly, showing that our approach is robust with respect to the parameter specifying the number of patches. Consequently, we recommend 30 as the default value. A smaller value (e.g. 10) will merge some parts which need to remain separate, and although a larger value is in principle better, it will unnecessarily increase computation time without producing significantly better results.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCG.2019.2891634, IEEE Computer Graphics and Applications

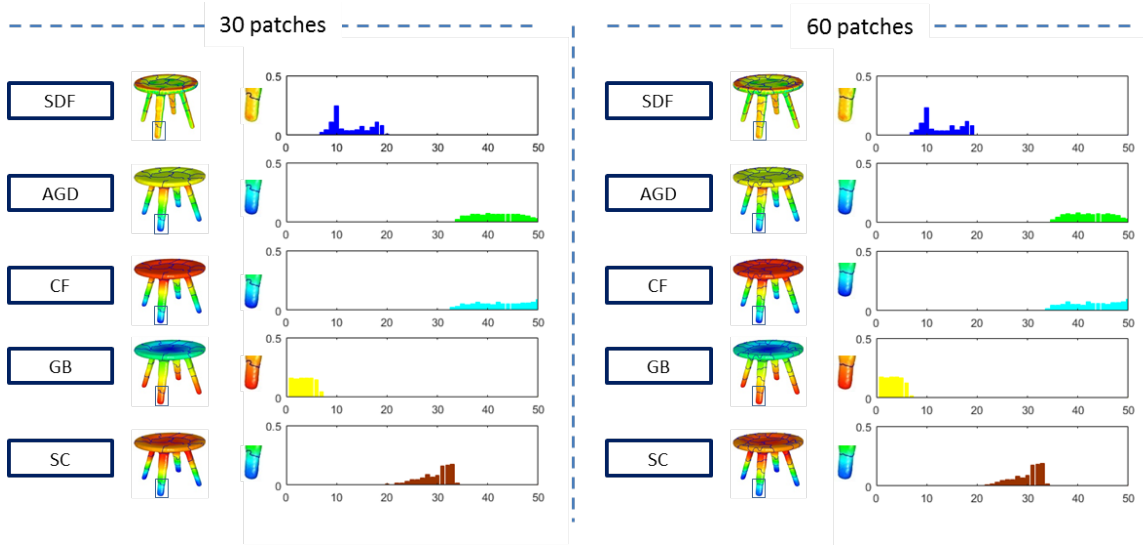


Figure 8: We show the impact of the number of patches (30 patches on the left vs. 60 patched on the right) used in over-segmentation on the feature values. SDF stands for Shape Diameter Function, AGD for Average Geodesic Distance, CF for Conformal Factor, GB for Geodesic distance to the Base of the shape, and SC for Shape Contexts. Note that the histogram values based on all these metrics defined on the patches are robust to the patch number parameter.

CONCLUSION

We have presented an active learning approach for shape co-segmentation. We regard the co-segmentation problem as a graph learning problem, where the GCN approach is leveraged to predict the class labels for the unlabeled samples through a layer-wise propagation rule defined in the network. Additionally, we have introduced an active learning method that queries the most informative samples to extend the training samples of our method. We have validated the effectiveness and utility of our method on experimental results.

Our approach has several limitations which suggest avenues for future work. One such limitation is that our algorithm fails to deal with out-of-sample data, especially in the setting of dynamic graphs. And another limitation has to do with the fact that the similarity distances between the nodes in the graph are measured by means of concatenating multiple features and performing a KNN search. A more rigorous strategy to deal with multiple features could be expected to lead to even better results.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Nos. 61602139, 61402387, 61502129), the Open Project Program of State Key Lab of CAD&CG, Zhejiang University (Nos. A1803, A1817), the Guiding Project of Fujian Province (No. 2018H0037), and the Zhejiang Province science and technology planning project (No. 2018C01030).

REFERENCES

1. Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, B. Chen, "Active co-analysis of a set of shapes," ACM Transactions on Graphics, vol. 31, no. 6, 2012, pp. 165–174.
2. Z. Wu, R. Shou, Y. Wang, X. Liu, "Interactive shape co-segmentation via label propagation," Computers & Graphics, vol. 38, 2014, pp. 248–254.
3. A. Golovinskiy, T. A. Funkhouser, "Consistent segmentation of 3d models," Computers & Graphics, vol. 33, no. 3, 2009, pp. 262–269.
4. O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, D. Cohen-Or, "Unsupervised co-segmentation of a set of shapes via descriptor space spectral clustering," ACM Transactions on Graphics, vol. 30, no.6, 2011, pp. 126:1–126:9.
5. Z. Wu, Y. Wang, R. Shou, B. Chen, X. Liu, "Unsupervised co-segmentation of 3D shapes via affinity aggregation spectral clustering," Computers & Graphics, vol. 37, no. 6, 2013, pp. 628–637.

6. P.Wang, Y. Liu, Y. Guo, C. Sun, X. Tong, "O-CNN: octree-based convolutional neural networks for 3d shape analysis," ACM Trans. Graph., vol. 36, no.4, 2017, pp. 72:1–72:11.
7. E. Kalogerakis, M. Avekiou, S. Maji, S. Chaudhuri, "3D Shape Segmentation with Projective Convolutional Networks," CVPR, 2017, pp. 6630-6639.
8. T. N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks," CoRR abs/1609.02907.
9. Z. Shu, C. Qi, S. Xin, C. Hu, L. Wang, Y. Zhang, L. Liu, "Unsupervised 3d shape segmentation and co-segmentation via deep learning," Computer Aided Geometric Design, vol. 43, 2016, pp. 39–52.
10. E. Kalogerakis, A. Hertzmann, K. Singh, "Learning 3D mesh segmentation and labeling," ACM Transactions on Graphics, vol. 29, no. 4, 2010, pp. 102:1-102:12.
11. K. Guo, D. Zou, X. Chen, "3D Mesh Labeling via Deep Convolutional Neural Networks," ACM Transactions on Graphics (Proc.SIGGRAPH), vol. 35, no.1, 2015, pp. 3:1-3:12.
12. C. R. Qi, H. Su, K. Mo, L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," CVPR, 2017, pp. 77-85.
13. L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheer, L. Guibas, "A scalable active framework for region annotation in 3d shape collections," ACM Transactions on Graphics, vol. 35, no. 6, 2016, pp. 210:1-210:12.
14. M. Gori, G. Monfardini, F. Scarselli, "A new model for learning in graph domains," in: Proceedings of the International Joint Conference on Neural Networks, vol.2, 2005, pp. 729–734.
15. X. Wang, J. Wen, S. Alam, Z. Jiang, Y. Wu, "Semi-supervised learning combining transductive support vector machine with active learning," Neurocomputing, vol. 173, 2016, pp. 1288–1298.
16. D. Yu, B. Varadarajan, L. Deng, A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," Computer Speech & Language, vol. 24, no.3, 2010, pp.433–444.
17. J. Zhu, H. Wang, B. K. Tsou, M. Y. Ma, "Active learning with sampling by uncertainty and density for data annotations," IEEE Trans. Audio, Speech & Language Processing, vol. 18, no.6, 2010, pp.1323–1331.
18. H. Cai, V. W. Zheng, K. C. Chang, "Active learning for graph embedding," CoRR abs/1705.05085.
19. X. Chen, A. Golovinskiy, T. Funkhouser, "A benchmark for 3D mesh segmentation," ACM Trans. on Graphics, vol. 28, no.3, 2009, pp. 73:1–73:12.
20. L.J.P. van der Maaten and G.E. Hinton. "Visualizing High-Dimensional Data Using t-SNE," Journal of Machine Learning Research, vol. 9, 2008, pp. 2579-2605.

ABOUT THE AUTHORS

Zizhao Wu (corresponding author) is an assistant professor in the School of Media and Design, Hangzhou Dianzi University, China. His research interests include 3d shape analysis, machine learning, and human-computer interaction. Wu has a PhD in computer science from Zhejiang University. Contact him at wuzizhao@hdu.edu.cn.

Ming Zeng is currently an associate professor in the software school of Xiamen University, China. He obtained his Ph.D. degree in computer science from Zhejiang University in 2013. His research interests mainly lie in computer graphics and computer vision, especially analysis and reconstruction of 3D objects. Contact him at zengming@xmu.edu.cn.

Feiwei Qin is an associate professor in the College of Computer Science and Technology at Hangzhou Dianzi University. His research interests include artificial intelligence, image processing, and CAD. Qin has a PhD in computer science from Zhejiang University. Contact him at qinfeiwei@hdu.edu.cn.

Yigang Wang was born in Songxian, Henan Province. He received the Ph.D in Applied mathematics from Zhejiang University in China. Now he is a professor in School of Media and Arts, Hangzhou Dianzi University. His research interests include Image processing, Computer Graphics and Virtual Reality. Contact him at yigang.wang@hdu.edu.cn.

Jiří Kosinka is an Assistant Professor in the Scientific Visualization and Computer Graphics research group at the Bernoulli Institute of the University of Groningen, the Netherlands. He received his PhD degree from Charles University in Prague in 2006. His research interests include computer graphics, geometric modeling and processing, and computer aided design. Contact him at j.kosinka@rug.nl.