Computers & Graphics ■ (■■■) ■■■-■■■



1

Contents lists available at ScienceDirect

Computers & Graphics



journal homepage: www.elsevier.com/locate/cag

CAD/Graphics 2013

Interactive shape co-segmentation via label propagation

Q1 Zizhao Wu^a, Ruyang Shou^a, Yunhai Wang^b, Xinguo Liu^a

02 ^a State Key Lab of CAD & CG, Zhejiang University, China ^b Shenzhen VisuCA Key Lab/SIAT, China

ARTICLE INFO

Article history: Received 5 August 2013 Received in revised form 8 November 2013 Accepted 8 November 2013

Keywords: Shape co-segmentation Interactive segmentation Label propagation

1. Introduction

In recent years, there have been increasing interests in shape co-analysis, i.e., simultaneously analyzing a set of shapes. One of the most fundamental problems in this field is co-segmentation. Different from the traditional segmentation tools which treat shapes individually, co-segmentation approaches process shapes from an input set in a batch, and generate segmentations carrying consistent semantics across the shapes. The consistent segmentation has demonstrated great utility in modeling [1,2], shape retrieval [3,4], texturing [5], etc.

Previous attempts for solving this problem can be classified into three categories as supervised, semi-supervised and unsupervised. The supervised ones [5,6] take advantages of manually labeled training sets to generate consistent segmentation results. However, the accuracy of the results relies on the training sets, and not surprisingly, the training process is tedious and time consuming. The unsupervised methods [7,8] generally build their approaches on the patch-level. These methods have superior performance, but the results hinge upon the in-sample data.

Recently, Wang et al. [9] presented a semi-supervised learning method with the aid of constrained clustering, where the user can actively assist in the co-segmentation process by assigning pairwise constraints like must-link and cannot-link. This approach can generate error-free results with a sparse set of constraints. However, as some authors [10] mentioned, pairwise constraints are not expressive to the users. In addition, their approach is a transductive algorithm which does not handle with the out-of-sample data, i.e., given a new datum, it needs performing the algorithm over the whole pipeline, which is ineffective.

In this paper, we address the above issues by introducing an interactive shape co-segmentation method. Our motivation drives from label propagation which propagates labels through the

0097-8493/\$ - see front matter © 2013 Published by Elsevier Ltd. http://dx.doi.org/10.1016/j.cag.2013.11.009

10.1016/j.cag.2013.11.009

ABSTRACT

In this paper, we present an interactive approach for shape co-segmentation via label propagation. Our intuitive approach is able to produce error-free results and is very effective at handling out-of-sample data. Specifically, we start by over-segmenting a set of shapes into primitive patches. Then, we allow the users to assign labels to some patches and propagate the label information from these patches to the unlabeled ones. We iterate the last two steps until the error-free consistent segmentations are obtained. Additionally, we provide an inductive extension of our framework, which effectively addresses the outof-sample data. The experimental results demonstrate the effectiveness of our approach.

© 2013 Published by Elsevier Ltd.

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

dataset along high density areas defined by unlabeled data. Our method allows the users to participate in the co-segmentation procedure, and is built upon the patch-level, which guarantees the high speed. Specifically, starting from over-segmenting a set of shapes into primitive patches, we allow the users to assign labels to some patches, and then propagate the labels from these patches to the unlabeled ones. We iterate the last two steps until the errorfree consistent segmentations are obtained.

In addition, as mentioned previously, when building their approaches on the patch-level, state-of-the-art methods [7–9] are effective in dealing with in-sample dataset in their respective problem domains, but all these methods have not explored the out-of-sample data. We investigate the out-of-sample issue by introducing an inductive extension of our pipeline, where the new datum can be labeled effectively.

Comparing with the state-of-the-art algorithms, our approach is featured as follows:

- Intuitive: We provide an intuitive user interface. For the users to directly assign labels is more expressive than the pairwise constraints.
- Inductive: We introduce an inductive extension of our algorithm to deal with out-of-sample data.
- *Error-free*: We can achieve error-free results depending on the input dataset and the labels given by the users.
- Efficient: Our approach is graph-based, but requires no extra eigen-decomposition, which is different from the unsupervised methods [7,8].

The remainder of this paper is organized as follows. We review the related work in Section 2. We present the details of the proposed algorithm in Section 3. We show some experimental

results on benchmark datasets in Section 4, followed by conclusions and future work in Section 5.

2. Related work

In this section, we provide a brief review of the existing work on shape co-segmentation, interactive segmentation and label propagation.

Shape co-segmentation: Shape co-segmentation refers to simultaneously segmenting a set of shapes into meaningful parts and building their correspondence. The existing co-segmentation methods can be classified into three categories: the unsupervised, the supervised and the semi-supervised.

In the unsupervised setting, the early work reported by Golovinskiy and Funkhouser [11] builds reliable correspondences across segments of shapes using rigid shape alignment. However, their approach cannot handle shapes with large variations. Xu et al. [12] factor out the scale variation in the shape segments by clustering the shapes into different styles, depending on the scales of the shape parts. Still, their approaches are limited to the shapes that can be properly aligned.

To overcome this limitation, Huang et al. [13] introduce an optimization strategy for simultaneously optimizing the saliency of each segmentation as well as consistency between segmentations. However, due to the computational complexity, this approach does not scale well for large datasets. Sidi et al. [7] present a descriptorbased method that employs multiple feature descriptors to measure the similarities of the segments and poses co-segmentation as a clustering problem in a concatenated descriptor space. Because the descriptors are independent of the pose and location of the shapes, this method can handle shapes with rich variations in part composition and geometry. Instead of concatenating the different feature descriptors into one vector, Hu et al. [8] propose a feature fusion method to co-segment a set of shapes via subspace clustering. However, these unsupervised techniques hinge upon the in-sample data.

Kalogerakis et al. [5] present a supervised learning method to simultaneously segment and label shapes. Their approach needs prior knowledge learned from the training dataset, and has demonstrated a labeling high accuracy on a broad class of shapes. van Kaick et al. [6] optimize the previous method by incorporating the prior knowledge to train a classifier. However, the above supervised methods require a substantial number of manually labeled training shapes, and the training set has a large impact on the segmentation performance.

Very recently, Wang et al. [9] propose a semi-supervised method where the user can actively assist in the learning process by interactively providing inputs. The input consists of a sparse set of pairwise constraints, which are marked as must-link and cannot-link constraints. The authors show that a sparse set of constraints can quickly converge toward an error-free result. However, the pairwise constraints are not clearly expressed to the users. In addition, their approach is a transductive algorithm that is ineffective at handling out-of-sample data.

Interactive segmentation: Interactive shape segmentation approaches are simple and intuitively help users express their intentions. Consequently, they have received significant attention [14].

Many interactive techniques have been proposed. Some of them require the user to specify a few points on the desired cutting contour and then employ the geometric snake [15], scissoring [16,17], graph cut [18] or some other method [19] to find the final cutting boundaries. These methods are called boundary-based approaches. In the last few years, a series of region-based approaches [20–22] have been proposed, which take regional information as the input and

require a much smaller amount of user effort to complete the labeling process for all of the unlabeled faces of a shape.

69 In this paper, rather than segmenting an individual shape, we 70 present an interactive region-based technique to simultaneously 71 segment a set of shapes in a consistent manner.

67

68

80

81

82

83

84

85

86

87

88

89

90

91 92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

72 Label propagation: Label propagation was first introduced by Zhu and Ghahramani [23]. This technique propagates the labels through Q3 73 dense unlabeled regions and locates data with properties that are 74 similar to those of the labeled data. Their approach is graph-based, 75 which can be constructed straightforwardly by computing pairwise 76 similarities among all of the data. Due to its simplicity and robustness, 77 it has been used in processes such as patch labeling [24], image 78 segmentation [25], and image annotation [26]. 79

Some authors [27,28] have tried to optimize the original label propagation. Among them, Wang and Zhang [29] propose approximating the graph with a set of overlapped linear neighborhood patches (LNPs) and computing the edge weights in each patch using the neighborhood linear projection. Our work is directly inspired by the LNP. We apply this algorithm to our interactive shape cosegmentation setting.

3. Algorithm

3.1. Overview

Define a set of shapes $S = \{s_1, s_2, ..., s_N\}$, where s_i represents the *i*-th shape and N is the total number of shapes. Our algorithm simultaneously produces segments of the set of shapes S and builds their correspondences across these segments.

The pipeline of our approach is illustrated in Fig. 2. First, the algorithm pre-processes the set of shapes by partitioning the dataset into primitive patches and building a graph that represents the geometric similarities across them. Then, the user interactively labels some patches, which are used as initial seeds that guide the iterative propagation to find labels for the others.

Our algorithm is an iterative approach. Each iteration includes two steps: user interaction and label propagation based on the user input. These steps repeat until satisfactory results are obtained. Additionally, we apply an extension to the pipeline to handle out-of-sample data.

We discuss the preprocessing step in the next section, the label propagation in Section 3.3, and the inductive extension in Section 3.4.

3.2. Preprocessing

In this step, we start by over-segmenting the input shapes, where 112 normalized cuts [30] are employed to decompose each shape s_i into 113 primitive patches. In our settings, the number of patches per shape is 114 set to 30. Let $P = \{p_1, p_2, ..., p_M\}$ be the set of patches from all of the 115 shapes; it is clear that M=30 N. Fig. 1 gives an example of our over-116 117 segmentation results.

Our approach associates the representation of relations between 118 the patches with graphs. We represent this graph in matrix form, i.e., 119 by constructing an affinity matrix W whose entries $w_{i,i}$ carry the 120 similarities of p_i and p_i . Thus, to measure the similarities among 121 patches, we first choose five robust and discriminative shape descrip-122 123 tors to extract extrinsic geometric information about the patches; shapes can be informatively represented based on these data. These 124 widely recognized descriptors are the Shape Diameter Function (SDF) 125 [31], the Conformal Factor (CF) [32], the Shape Contexts (SCs) [33,34], 126 the Average Geodesic Distance (AGD) [35], and the geodesic distance 127 to the base of the shape (GB) [7]. The descriptors are all defined on the 128 mesh faces, so no additional conversions are required to make them 129 130 mutually representationally compatible. Then, to describe the patches 131 using the descriptors of the faces within them, we incorporate 132 histograms. Specifically, for patch p_i , we first build for each descriptor

2

1

2

3

4

5

6

7

8

9

10

11 12

13

14 15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46 47

48

49

50

51

52

53

54

55

56

57

ARTICLE IN PRESS



Fig. 1. The co-segmentation results of our algorithm for a set of images of lamps. The left figure shows the labels assigned by the users to the primitive patches of the given set of shapes. The right figure shows the co-segmentation results after label propagation.



Fig. 2. The pipeline of the proposed method. Given a set of shapes, we first over-segment them to primitive patches, and based on which we construct the neighborhood graph. Then, the users are required to label some patches in the light of their intention. Finally, we propagate the label information of the labeled patches to the unlabeled ones, and obtain the co-segmentation results. The user can iterate the last two steps until satisfactory results have been achieved. We also provide an inductive extension of our approach to handle the out-of-sample data.

 d_k the histogram $h_{k,i}$, counting the distribution of data from the faces $f \in p_i$. Then, we concatenate the histograms of all of descriptors into a sequence, that is, $h_i = \{h_{1,i}, h_{2,i}, ..., h_{K,i}\}$. In our case, K=5. The joined histograms h_i embed the discriminative geometric information that we want for the patches; we can use the histograms to evaluate the patch affinities.

After computing the measurement for each patch, we use the neighborhood information of each patch to construct the affinity matrix *W*. This process is based on the assumption that the local similarities are more reliable than the distant ones, which has been widely accepted in many research communities [36].

According to [36,29], we use the linear neighborhoods to reconstruct each point from its neighbors, which can be formulated using the following objective function:

$$f = \sum_{i} \left\| h_{i} - \sum_{h_{j} \in \Omega(h_{i})} W_{i,j} h_{j} \right\|^{2},$$
(1)

where $\Omega(h_i)$ represents the neighborhood set of h_i and $w_{i,j}$ is the linear weight of h_j to h_i , which constitutes the elements of W. We add two additional constraints, i.e., $\sum_i w_{i,j} = 1$ and $w_{i,j} \ge 0$, to normalize the weights of each point. This objective function is a standard locally linear embedding (LLE), and can be easily resolved through an eigen-decomposition problem [36].

3.3. Labeling and propagation

Our algorithm allows the user to actively assign some labels to a small subset of patches as the seeds for propagation. In this section, we discuss the iterative process, which repeatedly asks the user to specify some seeding labels and provides the user with the segmentation results via label propagation until satisfactory results are achieved. For convenience, we denote the labeled patches as $X_l \subseteq P$ and the unlabeled ones as X_u . We also define $L = \{l_1, l_2, ..., l_C\}$ as the label set and $y_i \in L$ as the label of p_i .

The goal of this step is to propagate the labeled set X_l to the unlabeled set X_u based on the affinity matrix W. Let F be the $M \times C$ labeling matrix that associates each patching p_i , $i \leq M$ with each label l_j , $j \leq C$, of which the element $f_{ij} \in F$ denotes the probability that p_i corresponds to the *j*-th label. After propagation, each

unlabeled patch has a probability of belonging to each label. We assign to y_i the label that is the maximum in the set $\{f_{i,1}, f_{i,2}, ..., f_{i,C}\}$, i.e., $y_i = \arg \max_{j \le C} f_{i,j}$.

We use an iterative convergence method to propagate labels to the unlabeled patches. In each iteration, the patches both absorb the label information from their neighbors and retain some fractions of their own initial label stations. This procedure repeats until none of the patches' labels changes. Let t be a time stamp and F^t be the labeling function at time t. The label propagation can be written as

$$F^{t+1} = \alpha W F^t + (1 - \alpha) F^0,$$
 (2)

where $\alpha \in (0, 1)$ is the weight, based on the label information inherited from the neighbors and the initial possibilities. F^0 is constructed according to X_l and X_{u} ; that is, $f_{i,j}^0 = 1$ if p_i is labeled as l_j by the results of a previous iteration or assignments from a user which can overwrite the former if applicable, otherwise $f_{i,j}^0 = 0$. Our algorithm iterates Eq. (2) until convergence. Let F^* be the limit of F^t . Finally, we output the labels Y with $y_i = \arg \max_j f_{i,j}^*$ for the unlabeled patches. This procedure is summarized as Algorithm 1. Figs. 3 and 4 illustrate the label propagation process.

The convergence analysis of the label propagation algorithm is presented in Appendix A, where we refer to the LNP [29].

3.4. Inductive extension

To enhance the scalability of our algorithm and handle out-ofsample computations, we propose an inductive extension. Given a new input shape, we first preprocess it using the method mentioned in Section 3.2, obtaining small patches with the corresponding histograms. Denoting h_k as the histogram of patch p_k of the new shape, we then assign the weights to its linear neighborhoods by minimizing the following objective function:

$$\arg\min_{w_{k,i}}\sum_{i}\left\|h_{k}-\sum_{h_{i}\in\Omega(h_{k})}w_{k,i}h_{i}\right\|^{2},$$
128
129
130

where $\Omega(h_k)$ denotes the linear neighborhoods of h_k . After computing the weights, we can directly calculate y_k through the

Z. Wu et al. / Computers & Graphics ■ (■■■) ■■■–■■■



Fig. 3. The labeling results on the Gaussian dataset. (a) The original dataset. (b) Two data points are labeled. (c) Label propagation is performed on the dataset. (d) Different choices of labeling points.



following interpolation:

$$y_k = \underset{j \le C}{\arg\max} \sum_{i:h_i \in \Omega(h_k)} \sum_{j=1}^{c} w_{k,j} f_{i,j},$$
(3)

here y_k is the final label for p_k .

Algorithm 1. Label propagation algorithm.

Input: The set of over-segmented patches P; The set of labels $L = \{l_1, l_2, ..., l_C\}; X_l \subseteq P$, where patches are assigned to labels in *L* and $X_u = P - X_l$; The balancing parameter $\alpha \in (0, 1)$.

Output: The labels of patches $Y = \{y_1, y_2, ..., y_n\}$. 1. Employ descriptors to measure the patch and then construct the linear neighborhood graph W by solving Eq. (1) based on the histograms of the patches.

2. Construct the labeling matrix F^0 based on the label assignment of X_l .

3. Iterate Eq. (2) until convergence. Let F^* be the limit of F^t . Output the labels Y with $y_i = \arg \min_i f_{i,i}^*$ for the unlabeled patch $X_i \subseteq P$.

4. Iterate Steps 2 and 3 by interactively assigning labels to the patches until satisfactory results are obtained.

4. Experimental results

We evaluate our approach on the shape co-segmentation benchmark (COSEG) [9]. We set $\alpha = 0.5$ in all our experiments. Fig. 5 shows the visual results of our approach, which show that we can obtain close to error-free results with the help of user interaction and label propagation. More specifically, we use the 7, 15, 6, 12, 10, 21, 12 label assignments on the Lamps, Candelabra, Goblets, Irons, Guitars, Vases and Chairs sets of COSEG to generate the results. Note that other alternative assignments may produce the same results; the user can add as many constraints as desired to refine the results, depending on the user's intention.

It would be challenging for our algorithm to label large datasets because a small modification of the labeling patches may

substantially change the final results. In our experiments, we asked three users to label these large datasets until satisfactory results were gained. After obtaining these results, we choose the results with minimal labeling effort for each set. In our results, we have taken 113 assignments, 63 assignments and 97 assignments for the Large chairs, the Large vases and the Large tele-aliens sets, respectively. We note that it is a good choice to divide the large set into several smaller sets and then label these sets separately. Taking the large tele-aliens as an example, we divided the set into four groups; each group has 50 shapes. Then, we performed our algorithm to obtain the error-free results for each group. As a result, we produced 17, 19, 25, and 16 assignments, and the total number of assignments is 92, which is more effective than considering the large set as a whole.

Fig. 6 illustrates some co-segmentation results of the unsupervised approaches [7,8] on the COSEG. These results contain many mislabeled patches because the geometry alone cannot fully convey the semantics of parts. Instead, by incorporating the user's guidance, our approach can effectively alleviate this problem.

We also demonstrate the effectiveness of label propagation in Figs. 3 and 4. Given a dataset, by assigning some labels on some points, our approach propagates the label information from the labeled points to the unlabeled points. The labeling status of the whole dataset highly depends on the initial labeling assignments.

User study: We conducted a user study to compare our method and the method of Wang et al. [9]. Ten participants were invited to perform the experiments on several small datasets (Candelabra, Irons) and large datasets (Large vases, Large tele-aliens) from the COSEG. We required the participants to use the two interfaces of these semi-supervised leaning approaches for each dataset, where some fixed numbers for the constraints are given. Then, we measured the accuracy of co-segmentation results for each method under the given constraints. The method with a higher accuracy was considered to be more effective. In our setting, we used the average accuracy as the final score of a dataset for each method, and the users were given four numbers (5, 10, 15, 20) for the small datasets and four numbers (20, 40, 60, 80) for the large datasets. Finally, because the users wanted to achieve satisfying results easily and quickly, we also required the users to judge which method is more intuitive.

ARTICLE IN PRESS



Fig. 5. The results of our co-segmentation on the COSEG dataset [9]. Corresponding segments are shown in the same color. (a) Candelabra (15 assignments), (b) Goblets (6 assignments), (c) Irons (12 assignments), (d) Guitars (10 assignments), (e) Vases (21 assignments), (f) Chairs (12 assignments), (g) Large chairs (113 assignments), (h) Large vases (63 assignments), and (i) Large tele-aliens (97 assignments). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



Fig. 6. This figure shows some co-segmentation results for the COSEG dataset [9] by applying the unsupervised approaches [7,8]. (a) Sidi et al. [7], (b) Hu et al. [8], and (c) Hu et al. [8].

We used the ground truth segmentation in [9] and their accuracy measure as our quality measure:

Q5

7 users thought that our interface was more intuitive, while the other 3 users stated that our method was inconvenient due to the lack of active learning.

$$Accuracy(l,t) = \sum_{i} a_i \delta(l_i = t_i) / \sum_{i} a_i,$$
(4)

where a_i is the area of the face *i*, *l* is the label computed by the cosegmentation, *t* is the ground-truth labeling, and $\delta(x = y)$ is 1 if and only if *x* equals *y*. We average the accuracies for all of the shapes in the set as the accuracy of the co-segmentation method.

Fig. 7 shows the average co-segmentation accuracies achieved by the participants in our experiments. We can see that our system helps the users to perform better than the other method [9]. In addition, we provide much faster feedback when updating the constraints, which we discuss later. As for the survey results,

Inductive evaluation: We tested the inductive extension of our algorithm on the Large chair dataset, where we randomly selected 200 shapes as in-sample data and used the remaining samples as out-of-sample data. After we obtained the error-free results of the in-sample data, the inductive extension was conducted to predict the labels of the out-of-sample data. Fig. 8 shows an example where a new datum has been pre-processed and labeled. As a result, we achieved an average of 88.7% accuracy for the out-of-sample shapes, which is associated with O(n) time complexity (less than 1s for each shape in our case). In comparison, the method of Wang et al. [9] requires $O(n^3)$ (20 min). This result demonstrates the effectiveness of our inductive algorithm.



Fig. 7. The accuracies of the co-segmentation results and their corresponding number of label assignments. (a) User study on some small datasets, where about 7 s is needed for one update of the labels in these cases. (b) User study on some large datasets, where about 8 min is needed for one update of the labels in these cases.



Fig. 8. Given an out-of-sample data, we can fast obtain its labels through interpolation of the inductive extension of our algorithm.

Performance: We implemented our approach in C++ and MATLAB and evaluated the performance on a 2.83 GHz Intel CoreTM 2 Quad processor with 4 GB of RAM.

The pre-processing of our pipeline requires considerable time, particularly the computation of geodesic distance and shape context. The label propagation is very fast; the time complexity is $O(n^2)$, where *n* is the total number of patches. Consider a small dataset that contains 20 shapes, where each shape is decomposed into 30 patches. Therefore, the linear neighborhood matrix *W* has the dimension of 600 × 600, and our algorithm needs approximately 100 iterations to converge, which takes approximately 7 s for one update of its labels. For a large dataset, we need approximately 8 min to complete an update.

5. Conclusions

We presented a novel algorithm to interactively co-segment a set of shapes via label propagation. Given a set of shapes with a small number of user-defined labels on the over-segmented patches, our method automatically propagates those labels to the remaining unlabeled patches through an iterative procedure. Moreover, we provide an out-of-sample extension of our approach. The experiments demonstrated the effectiveness of the proposed approach. 122

Limitations and future work: There are many limitations in our algorithm, which suggests many avenues for future work. First, our approach suffers from a lack of suggestions when labeling samples

Please cite this article as: Wu Z, et al. Interactive shape co-segmentation via label propagation. Comput Graph (2013), http://dx.doi.org/10.1016/j.cag.2013.11.009

to improve the co-segmentation results with less effort. In the future, we plan to integrate our labeling process with the active learning setting [38]. Second, our approach is graph-based. How to better measure the similarities preserved in the graph plays an important role in the following steps. A more rigorous treatment of this problem is appreciated. Finally, our approach builds upon the patch-level, so the final co-segmentation cuts are relying on the initial over-segmentation boundaries, which encourages us to exploit a more robust over-segmentation method to refine the cutting boundaries for the applications.

Acknowledgment

Many thanks to the anonymous reviewers for their valuable comments. This work was partially supported by NSFC (Nos. 6137 9068 and 61202222), Shenzhen Science and Innovation Program (CXB201104220029A).

Appendix A. Convergence analysis

In this section, we present the convergence analysis of Algorithm 1. We can see from the Algorithm 1 that our approach needs to iterate Eq. (2) until it converges with the given input α , *W* and F^0 . By substituting variable F^t in Eq. (2), we have

$$F^{t+1} = (\alpha W)^t F^0 + (1-\alpha) \sum_{n=0}^t (\alpha W)^n F^0.$$
 (5)

Because we have constrained $\sum_{j} w_{i,j} = 1$ and $w_{i,j} \ge 0$, according to the theorem of *Perron–Frobenius* [37], we can conclude that the spectral radius of *W* is no larger than one. In addition, $0 \le \alpha \le 1$. We have

 $\lim (\alpha W)^t = 0$ and

$$\lim_{t\to\infty}\sum_{n=0}^t (\alpha W)^n = (I - \alpha W)^{-1},$$

where *I* is the identity matrix. From these, Eq. (2) will converge to $F^* = \lim F^t = (1 - \alpha)(I - \alpha W)^{-1}F^0.$ (6)

In other words, we can predict the labeling results in one step.

References

- [1] Funkhouser TA, Kazhdan MM, Shilane P, Min P, Kiefer W, Tal A, et al. Modeling by example. ACM Trans Graph (Proc SIGGRAPH) 2004;23(3):652–63.
- [2] Xu K, Zhang H, Cohen-Or D, Chen B. Fit and diverse: set evolution for inspiring 3D shape galleries. ACM Trans Graph (Proc SIGGRAPH) 2012;31(4):57:1-10.
- [3] Shalom S, Shapira L, Shamir A, Cohen-Or D. Part analogies in sets of objects. In: 3DOR; 2008, p. 33–40.
- [4] Chaudhuri S, Kalogerakis E, Guibas LJ, Koltun V. Probabilistic reasoning for assembly-based 3D modeling. ACM Trans Graph (Proc SIGGRAPH);30(4).
- [5] Kalogerakis E, Hertzmann A, Singh K. Learning 3D mesh segmentation and labeling. ACM Trans Graph (Proc SIGGRAPH) 2010;29(4):1–11.
- [6] van Kaick O, Tagliasacchi A, Sidi O, Zhang H, Cohen-Or D, Wolf L, et al. Prior knowledge for shape correspondence. Comput Graph Forum (Special Issue of Eurographics) 2011;30(2):553–62.

- [7] 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 Graph (Proc SIGGRAPH Asia) 2011;30(6):126:1–9.
- [8] Hu R, Fan L, Liu L. Co-segmentation of 3D shapes via subspace clustering. Comput Graph Forum (Proc SGP) 2012;31(5):1703–13.
- [9] Wang Y, Asafi S, van Kaick O, Zhang H, Cohen-Or D, Chen B. Active co-analysis of a set of shapes. ACM Trans Graph (Proc SIGGRAPH Asia) 2012;31(6):165–74.
- [10] Wang X, Qian B, Davidson I. Labels vs. pairwise constraints: a unified view of label propagation and constrained spectral clustering. In: ICDM; 2012. p. 1146– 51.
- [11] Golovinskiy A, Funkhouser TA. Consistent segmentation of 3D models. Comput Graph 2009;33(3):262–9.
- [12] Xu K, Li H, Zhang H, Cohen-Or D, Xiong Y, Cheng Z. Style-content separation by anisotropic part scales. ACM Trans Graph (Proc SIGGRAPH Asia) 2010;29 (5):184:1–10.
- [13] Huang Q-X, Koltun V, Guibas LJ. Joint shape segmentation with linear programming. ACM Trans Graph (Proc SIGGRAPH) 2011;30(6):125.
- [14] Shamir A. A survey on mesh segmentation techniques. Comput Graph Forum 2008;27(6):1539–56.
- [15] Lee Y, Lee S. Geometric snakes for triangular meshes. Comput Graph Forum 2002;21(3):229–38.
- [16] Lee Y, Lee S, Shamir A, Cohen-Or D, Seidel H-P. Intelligent mesh scissoring using 3D snakes. In: Pacific conference on computer graphics and applications; 2004. p. 279–87.
- [17] Lee Y, Lee S, Shamir A, Cohen-Or D, Seidel H-P. Mesh scissoring with minima rule and part salience. Comput Aided Geom Des 2005;22(5):444–65.
- [18] Sharf A, Blumenkrants M, Shamir A, Cohen-Or D. Snappaste: an interactive technique for easy mesh composition. Vis Comput 2006;22(9–11):835–44.
- [19] Ji Z, Liu L, Chen Z, Wang G. Easy mesh cutting. Comput Graph Forum 2006;25 (3):283–91.
- [20] Xiao C, Fu H, Tai C-L. Hierarchical aggregation for efficient shape extraction. Vis Comput 2009;25(3):267–78.
- [21] Zheng Y, Tai C-L. Mesh decomposition with cross-boundary brushes. Comput Graph Forum 2010;29(2):527–35.
- [22] Zhang J, Wu C, Cai J, Zheng J, Tai X-C. Mesh snapping: robust interactive mesh cutting using fast geodesic curvature flow. Comput Graph Forum 2010;29 (2):517–26.
- [23] Xiaojin Zhu ZG. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107. Carnegie Mellon University.
- [24] Bishop CM, Ulusoy I. Object recognition via local patch labelling. In: Proceedings 2004 workshop on machine learning; 2005.
- [25] Wang F, Wang X, Li T. Efficient label propagation for interactive image segmentation. In: Fourth international conference on machine learning and applications; 2007. p. 136–41.
- [26] Kang F, Jin R, Sukthankar R. Correlated label propagation with application to multi-label learning. In: Computer vision and pattern recognition (CVPR); 2006. p. 1719–26.
- [27] Zhu X, Ghahramani Z, Lafferty JD. Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML; 2003. p. 912–9.
- [28] Tian Z, Kuang R. Global linear neighborhoods for efficient label propagation. In: SDM; 2012. p. 863–72.
- [29] Wang F, Zhang C. Label propagation through linear neighborhoods. In: ICML; 2006. p. 985–92.
- [30] Golovinskiy A, Funkhouser TA. Randomized cuts for 3D mesh analysis. ACM Trans Graph (Proc SIGGRAPH) 2008;27(5):1–12.
- [31] Shapira L, Shalom S, Shamir A, Cohen-Or D, Zhang H. Contextual part analogies in 3D objects. Int J Comput Vis 2010;89(1–2):309–26.
- [32] Ben-Chen M, Gotsman C. Characterizing shape using conformal factors. In: 3DOR; 2008. p. 1–8.
- [33] Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 2002;24(4):509–22.
- [34] Körtgen M, Park G-J, Novotni M, Klein R. 3D shape matching with 3D shape contexts. In: The 7th central European seminar on computer graphics, vol. 3; 2003. p. 5–17.
- [35] Hilaga M, Shinagawa Y, Komura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3D shapes. In: SIGGRAPH; 2001. p. 203–12.
- [36] Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science 2000;290(5500):2323–6.
- [37] Golub GH, van Loan CF. Matrix computations. 3rd ed.Johns Hopkins University Press; 1996.
 [11]
- [38] Settles B. Active learning literature survey. Computer Sciences Technical Report 1648. University of Wisconsin-Madison; 2009.

107

108

109

112

113

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

57

1

2