DOI: 10.1111/cgf.12436
Eurographics Symposium on Geometry Processing 2014
Thomas Funkhouser and Shi-Min Hu
(Guest Editors)

Volume 33 (2014), Number 5

Transductive 3D Shape Segmentation using Sparse Reconstruction

Weiwei Xu[1] Zhouxu Shi[2] Mingliang Xu[2] Kun Zhou[3] Jingdong Wang[4] Bin Zhou[2]
Jinrong Wang[1] Zhenming Yuan[1]

[1]Hangzhou Normal University [2]Zhengzhou University [3]Zhejiang University [4] Microsoft Research

Abstract

We propose a transductive shape segmentation algorithm, which can transfer prior segmentation results in database to new shapes without explicitly specification of prior category information. Our method first partitions an input shape into a set of segmentations as a data preparation, and then a linear integer programming algorithm is used to select segments from them to form the final optimal segmentation. The key idea is to maximize the segment similarity between the segments in the input shape and the segments in database, where the segment similarity is computed through sparse reconstruction error. The segment-level similarity enables to handle a large amount of shapes with significant topology or shape variations with a small set of segmented example shapes. Experimental results show that our algorithm can generate high quality segmentation and semantic labeling results in the Princeton segmentation benchmark.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Shape Analysis—Shape SegmentationSparsity

1. Introduction

Shape or mesh segmentation aims to partition the shape into functional or semantic segments/parts as a compact representation to analyze shape structures. Exploiting the structure information as guidance, the efficiency of geometry processing algorithms in diverse areas, such as 3D modeling [FKS*04, SFCH12], skeleton extraction [KT03], compression [KG00], collision detection [LWTH01], parametrization [ZSGS04] and shape retrieval [GCO06], can be effectively improved. For example, Shapira et al. [SSS*10] developed a contextual part analogy method with hierarchical segmentation, which supports highly discriminative 3D shape part retrieval.

A large amount of research efforts have been devoted to improve the accuracy of shape segmentation algorithms. Various types of low-level geometry properties, such as geodesic distance, concavity, shape diameter function, are investigated in graph-theoretic or variational framework to produce high-quality segmentation results. They can well approximate the three key factors, similarity, proximity and good continuation, in visual grouping [KT03, SSCO08, ZZWC12]. However, it is still difficult for them to obtain satisfactory segmentation results for all kinds of shapes with such low-level geometry properties, since prior shape similarities, important in human segmentation procedure, are not considered.

Recent shape segmentation researches advocate the integration of prior information to further enhance the accuracy of segmentation results. Based on the 3D segmentation benchmark [CGF09], Kalogerakis et al. [KHS10] learned discriminative models, for a specific category of shapes, to segment the shape into parts with seman-tic labels. Co-segmentation techniques can segment a set of shapes within same category in a consistent way [SvKK*11, HKG11], where the shape category constraint can be viewed as a weak prior information. Despite significant research progress in such shape segmentation algorithms, it is still not easy to generalize them to handle fast growing 3D shape data. Given a new shape, its category should be classified or provided by the user in co-segmentation techniques. The re-run of co-segmentation algorithm with the new shape is also computationally expensive. More-over, there exist shapes, for instance, sphinx, which are difficult to be classified into any category. Similarly, the learning algorithm in [KHS10] also requires category information.

In this paper, we propose a transductive shape segmenta-

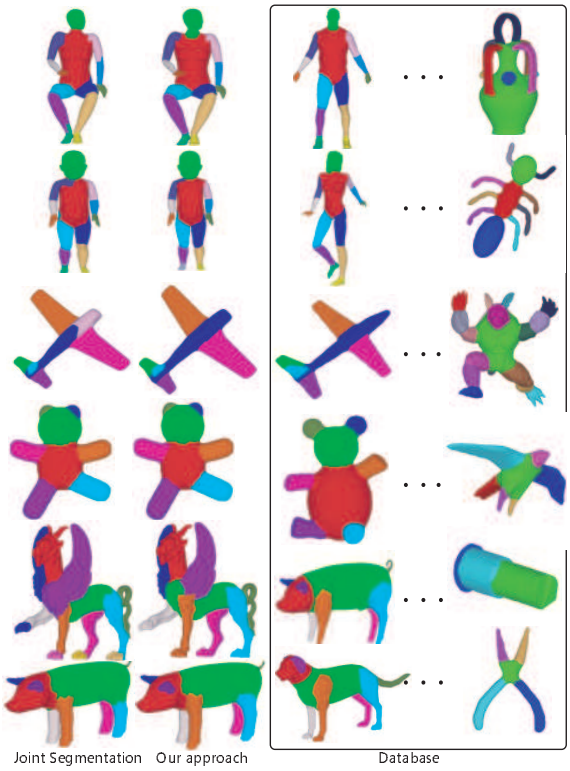

Figure 1: Segmentation results. Left Column: The segmentation results by joint segmentation algorithms in [HKG11]. Middle column: Our transductive segmentation result via a segment similarity maximization. Right Column: The selected example segmentation results in benchmark to show our results are consistent with the database. For each input shape, we run both algorithms using its top 10 shape query results from the segmentation database in [CGF09]. The segment similarity induced by L1 sparse reconstruction can result in more consistent segmentation results with the example segmentations.

tion algorithm, which can transfer prior segmentation results in the segmentation benchmark [CGF09] or from state-of-the-art segmentation algorithms to new shapes without explicit specification of category information. Its key idea is to maximize the segment similarity between the segments in the input shape and the segments in database, where the segment similarity is computed through sparse reconstruction error. The input shape is pre-partitioned into a set of segmentations as a data preparation, and a linear integer programming algorithm is developed to select the segments from them to generate the final segmentation via similarity maximization. There are two distinctive features of our algorithm:

- Prior shape similarity is measured at segment-level. This enables us to handle shape with significant topology or shape variations with a small set of example segmented shapes. As shown in 3D shape synthesis techniques [KCKK12, XZCOC12], segment combination can create a large amount of shapes to save modeling efforts. Moreover, it enables us to combine the segments from different categories in the segmentation (see Figure 6).
- Sparse reconstruction is used to compute the similarity score. That is, a segment is deemed to be similar to the segments in the database if its sparse reconstruction error is low. It can not only tolerate the possible false-positive results from 3D shape query but also ease the system implementation (see Figure 4). Since similar segments should have similar shape descriptors, they usually form a cluster in feature space, which can be covered by a linear subspace. Therefore, we use linear combination of shape descriptors as the reconstruction method (Eq. 1).

We have tested our algorithm on a variety of 3D shapes and achieved high-quality segmentation result, and the classification ability of sparse reconstruction method is also used to transfer segment labels in database to input shapes. Moreover, the ability to obtain segmentation results via linear programming results in a fast algorithm. We can optimize for the segmentation results within 1 seconds for a new shape with around 16000 triangles. However, it takes minutes by joint segmentation algorithm in [HKG11], since a multi-class quadric programming problem of thousands of variables needs to be solved for a set of shapes.

2. Related work

There are numerous research efforts on shape or mesh segmentation. Please see [Sha08, AKM*06, CGF09] for a comprehensive survey. Meshes can be partitioned according to geometric properties computed from a single mesh, such as con-cavity, skeleton or primitive shapes [MW99, KT03, L04, SKS06, LLL07, LGQ08, LHMR08, LW08]. Shape deformation property can also be used in mesh segmentation. Huang et al. [HWAG09] applied modal analysis technique in Mechanics to decompose the shape into parts with rigid motion. A common technical difficulty in this line of research works is that no single geometry feature is suitable to all cases of mesh segmentation. In contrast, Golovinskiy et al. [GF08] proposed to segment the shape into a large amount of segmentations with different geometry properties, algorithms and parameters as a base for shape analysis. Wang et al. [WGW*13] presented to extend knowledge in image segmentation to 3D shapes via projective analysis to tackle the challenge in collecting high-quality 3D models.

A recent trend in shape segmentation research is extended from a single shape to a set of shapes [HFL12, WAvK*12, WWS*13]. Golovinskiy et al. [GF09] showed that consistent segmentation constraint from a set of shapes can result in better individual shape segmentation. This algorithm is then improved in co-segmentation algorithms [SvKK*11,HKG11] to handle shapes with more variations. A distinctive feature of co-segmentation algorithms is that part correspondence between shapes are derived automat-

ically. In contrast, with prior shape segmentation and part correspondence data, Kalogerakis et al. [KHS10] learned a Conditional Random Field models for shapes in same category, such as vase, human, to simultaneously segment and label a new shape from same category. Our method is designed based on the fast development of shape segmentation algorithms, and we target to transfer high quality segmentation results from these algorithms to new shapes. Combined with State-of-the-art 3D shape query methods [FMK*03, BBGO11], our method is of great potential to handle heterogenous, large scale shape data set.

Our work is also inspired by the application of sparsity constraints in image segmentation and annotation algorithms. Sparse model selection technique has been used in image segmentation to effectively single out relevant prior models [CYH13]. Zhang et al. [ZZD*11] proposed a sparse shape representation to pick up priors from database for deformable object segmentation, and patch-level sparse representation are explored in [TWC*13] for brain MRI image segmentation. For segmentation algorithms based on graph theory, sparsity are also used to improve the edge affinity function for accurate image segmentation results [CLW*11]. In image annotation, sparse reconstruction is used to transfer region tags or labels [LCY*09], and the annotation accuracy can be enhanced through the context constraint modeled by structural sparsity [YWSZ13]. However, with our best knowledge, there does not exist research works on generalizing such sparsity-based 2D segmentation algorithms to 3D shape segmentation. Moreover, we optimize the segmentation by treating the sparse reconstruction error as a segmentation evaluation score.

Due to its robustness to outliers and ability to discern noise and feature, L1 norm has also been used in geometry processing for point cloud data denoising, skeleton extraction and polycube generation [ASGCO10, HWCO*13, WYL*14, HJS*14]. A L0 norm minimization algorithm was developed by He et al. [HS13] for triangular mesh denoising. Our work focuses on 3D shape segmentation, which can be viewed as a new application of sparsity in geometry processing.

3. Overview

The three stages of our algorithm, initial segmentation, shape query and final segmentation generation, are illustrated in Figure 2. The initial segmentation and shape query can be viewed as segmentation data preparation, and their generated segments are the input to a linear integer programming algorithm for final segmentation generation. Sparse reconstruction plays a key role to guide the algorithm to choose the initial segments similar to example segments in database.

Initial segmentation: Our initial segmentation generation stage is similar to [HWAG09]. We first perform normalized cut to partition the input shape into around 50 patches using concavity as affinity measure [SM00], and then group them into large number of segments with random cuts method [GF08]. The initial number of segments k in a segmentation is estimated from the shape query result. For each different segment numbers k_q in the query result, we compute a range $\{0.5k_q, 1.5k_q\}$. Afterwards, for each initial target segment number in this range, we perform 80 randomized segmentations. We perform segment pruning to reduce the number of segments in the final optimization (see Sec. 4.1).

Shape query: Given an input shape, shape query is to retrieve its similar shapes in database. In current implementation, we adopt pose-oblivious signature in [GSCO07] as a global descriptor in measuring the shape similarity. The shapes ranked in top 10 are used as database shapes, and their segmentation information is used to derive the input shape segmentation. State-of-the-art shape query algorithms can be easily integrated into our algorithm to enhance the query accuracy, which should also improve the final segmentation quality.

Final segmentation generation: The final segmentation is generated by choosing segments from the initial segments to form a complete segmentation. We pose it as a linear binary integer programming problem, where an integer variable $x \in \{0;1\}$ is assigned to each initial segment and its value 1 indicates a selection of the associated segment into the final segmentation. The objective function of the linear programming algorithm is designed to minimize the sparse reconstruction errors for the selected initial segments.

4. Segmentation Algorithm

In this section, we describe how to evaluate the segment similarity through sparse reconstruction and the linear binary integer program- ming algorithm for the final segmentation generation.

Notation: The set of patches generated in normalized cut is denoted by $\mathcal{P} = \{P_i | i = 1..m\}$, and a segmentation of a shape $\mathcal{M}$ denoted by $\mathcal{S} = \{s_i | i = 1..n\}$, where s_i represents a segment in the segmentation. Each segment, in our setting, consists of adjacent patches from the patch set $\mathcal{P}$. A segmentation $\mathcal{S}$ is deemed to be valid if each patch P_i belongs to one and only one segment.

4.1. Segment Evaluation

The goal of segment evaluation is to compute the segment similarity between an initial segment and the example segments in the database. The evaluation score is then used to guide the optimization algorithm to choose the similar initial segments to form a valid segmentation. Thus, the segmentation information of the example shapes in database can be transferred via the efforts to maximize the similarity. This design is to simulate human segmentation procedures

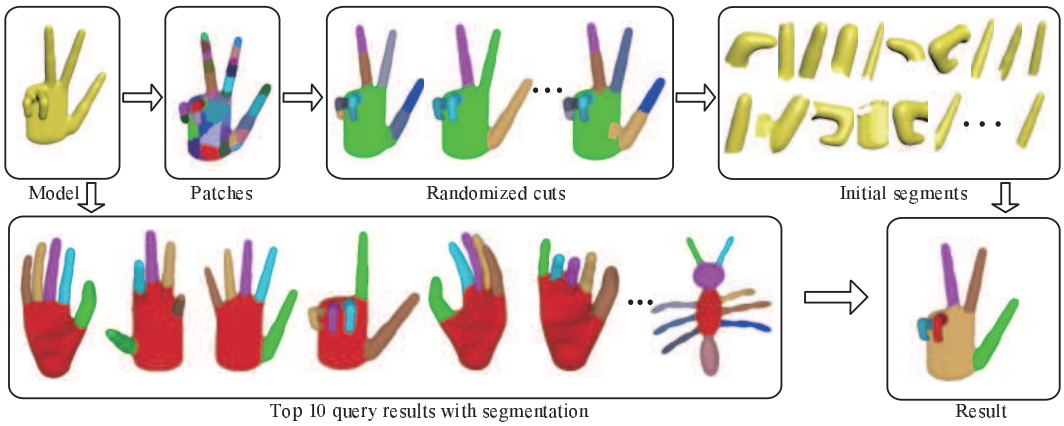

Figure 2: The pipeline of our transduction shape segmentation algorithm.

where the prior model plays an important role. According to perceptual grouping research in Psychology, the human segmentation procedure actually involves object analogy [LeG03]. If a given shape or part of the shape is similar to the recorded shapes in mind, human tends to perform visual grouping according to the prior similarity.

We pose the problem of computing segment similarity as a sparse reconstruction problem. Specifically, given an initial segment, we perform sparse reconstruction with $L1$ norm to choose sparse segments in database that can well reconstruct the initial segment, which is measured based on shape descriptor distance. For an input segment s_i , its reconstruction score is optimized according to the formula:

$$E_i = \|\mathbf{Y}_i - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_1 \quad (1)$$

where Y_i is a column vector which is the shape descriptor computed for the segment s_i, and the columns of $\mathbf{X}$ represent the shape descriptors computed for the segments of shapes in the database. Moreover, the non-zero coefficients can encode the segment correspondences which is also exploited in image annotation applications.

Shape descriptor: The shape descriptor computed for each segment is the combination of shape distribution (SD) [OFCD02], shape diameter function (SDF) [SSCO08], and context statistics, which forms a 79-dimension feature vector. The SD and SDF are normalized histograms of the distance and shape diameter values computed at sampled points on the segment surface to achieve reliable feature vector. Each histogram is of dimension 36. The context statistics is used to measure the relative size of the oriented bounding box of a segment to the whole shape. In our current implementation, they are 7 values: (a) global ratio: the length, width, height and diagonal length of the segment oriented bounding box normalized by the longest geodesic distance of the mesh. (b) the ratio of length to width, height to width and length to height for the segment bounding box.

Segments pruning: Since we generate a large number of initial segments using random cuts for input shapes, it will induce heavy computational demand if we compute $L1$ reconstruction for each segment. Thus, we first filter out segments with $L2$ reconstruction error, which is much faster since the factorization of the matrix $\mathbf{X}^T\mathbf{X}$ in linear least squares is only needed to be performed once. We retain top 20% segments to generate the final segmentation according to $L2$ reconstruction error. Similar to [HKG11], we also include segmentations with highest minimum cut cost for each target segment number k to ensure the selected segments can form a complete segmentation.

4.2. Segmentation by Linear Binary Integer Programming

In this section, we describe a linear binary integer programming algorithm to select segments to form a segmentation with prior segmentation information in the database. The formulation is inspired by the joint segmentation algorithm in [HKG11]: We also assign an $0-1$ indicator variable x_i for each initial segment s_i , and set up constraints to form the segmentation. However, since we are targeting to segment one input shape with the prior segmentations, the overall problem can be formulated into a linear binary integer programming problem, which can be efficiently solved.

Energy Function: We minimize the following energy function to select segments from the set of initial segments:

$$\min \sum_i x_i E_i - \alpha \sum_i x_i \quad (2)$$

The first term is the sum of reconstruction cost E_{s_i} where the indicator variables x_{s_i} is 1 . The second term indicates the number of non-zero indicator variables, which simulates the reconstruction error normalization. Otherwise, the algorithm may prefer the segmentation result with few segments due to the fact that the difference between reconstruction errors sometimes may not be enough to push the algorithm to correct segment numbers. α is a weight whose value is default 1.5 in our implementation.

Segmentation Constraint: We require the selected subset

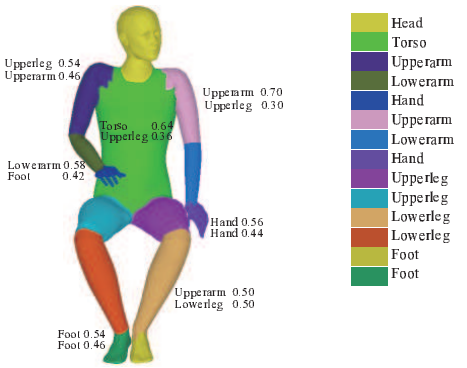

Figure 3: Labeling result for a human model. The label names and top 2 reconstruction weights obtained in segmentation are displayed on or near the corresponding segments. Their final labeling results are shown on the right legend.

of segments forms a valid segmentation of the input shape. As formulated in [Huang et al. 2011], the constraint can be written into the follow-ing linear equation:

$$\sum_{i \in cover(P_j)} x_i = 1;\ \forall P_j \in \mathcal{P} \tag{3}$$

where $cover(P_j)$ indicates the set of segments covering the patch P_j. Assembling this equation at each patch yields:

$$\mathbf{Ax} = 1 \tag{4}$$

The dimension of matrix A in the above equation is $m \times n$, where m is the number of patches, and n is the number of initial segments. $\mathbf{x}$ is the stacked vector of indicator variables x_i

Number constraint: This constraint is for user to control the number of segments in the final segmentation. It is useful in specific geometry processing algorithms. For example, the user might wan-t to control the number of segments so as to control the number of bones in skeleton extraction. It can be formulated into a linear constraint as well:

$$\sum_i x_i = k \tag{5}$$

where k is the target number of segments. If this constraint is selected, the second term in Eq.2 is constant and thus ignored in the optimization.

Putting all them together, we get the following optimization problem:

$$\begin{aligned} &\min_{x_i} \sum_i x_i E_i - \alpha \sum_i x_i \\ st: \quad &\mathbf{Ax} = 1 \\ &\sum_i x_i = k \end{aligned} \tag{6}$$

Note that the number constraint is enabled only according to user requirement. We adopt CVX package solve this linear binary integer programming problem [GB08].

4.3. Labeling

Given the segmentation result obtained by Eq.2, the labeling procedure aims to estimate its segment labels through the sparse reconstruction weight β computed via Eq.1. A straightforward solution might be to select the segment label with the database segment corresponding to its largest reconstruction weight for each segment s_i. However, this solution might be confused by segments of similar shape but with different label. For example, in a human body shape, the upperarm might be confused to be upperleg, since these two segments are geometrically similar (see Fig. 3). Thus, we solve the labeling problem with integer programming to select from 2 labels of the largest 2 reconstruction weights for each segment.

Let $\mathcal{S} = \{s_i | i = 1..n\}$ denote the segmentation result and $\mathcal{L} = \{l_i | i = 1..m\}$ denote the set of labels collected from the top 10 query results. For each segment s_i, we associate it with a set of indicator variables $\{x_i^j, j = 1..m\}$. x_i^j is 1 if the segment s_i is assigned with label l_j, otherwise 0. We can obtain the labeling result using a quadric integer program:

$$\begin{aligned} \min \quad & E_L = \sum_i \sum_j x_i^j W_i^j + \sum_{adj(s_i,s_j)} \sum_{p=1}^{m} \sum_{q=1}^{m} x_i^p x_j^q W_{ij}^{pq} \\ s.t. \quad & \sum_j x_i^j = 1, i = 1..n, \\ & x_i^j \in \{0,1\} \end{aligned} \tag{7}$$

The set of coefficients W_i^j represents the labeling costs if the segment s_i is assigned label j. Suppose the largest 2 reconstruction weights are β_{j_1} and β_{j_2} and they are from two database segments with label l_{j_1} and l_{j_2}. We thus set $W_i^{l_{j1}} = \beta_{j_1}/(\beta_{j_1} + \beta_{j_2})$ and $W_i^{l_{j1}} = \beta_{j_2}/(\beta_{j_1} + \beta_{j_2})$. The rest coefficients W_i^j are set to be a large value $1e3$ in our implementation, which means that we prefer the label of the segment s_i to be selected through segments corresponding its top 2 reconstruction weights. W_{ij}^{pq} is used to penalize the unreasonable label pair of two adjacent segment pairs. For example, it is not expected if a torso is connected to a lower leg in a human body shape. Thus, W_{ij}^{pq} is set to be a penalty value $1e4$ if label p and label q should not be adjacent to each other, otherwise W_{ij}^{pq} is 0.

We linearize the objective function so that it can be efficiently solved by CVX package [GB08]. We thus associate each pair, $x_i^p x_j^q$, a latent indicator variable z_{ji}^{pq}, and change the objective function into:

$$\begin{aligned} \min \quad & E_L = \sum_i \sum_j x_i^j W_i^j + \sum_{adj(s_i,s_j)} \sum_{p=1}^{m} \sum_{q=1}^{m} z_{ij}^{pq} W_{ij}^{pq} \\ s.t. \quad & \sum_j x_i^j = 1, i = 1..n, \\ & z_{ij}^{pq} \le x_i^p, z_{ij}^{pq} \le x_j^q, \sum_{p=1}^{m} \sum_{q=1}^{m} z_{ij}^{pq} = 1, \forall\ adj(s_i,s_j), \\ & x_i^j \in \{0,1\},\ \ z_{ij}^{pq} \in \{0,1\}. \end{aligned} \tag{8}$$

The constraints on z_{ij}^{pq} ensure its compatibility with $x_i^p x_j^q$. This linear integer programming problem can be efficiently solved within 0.2 seconds by CVX package.

5. Experiments

We have implemented our algorithm on a desktop PC with Intel I5 CPU in single thread. The running performance of our algorithm varies with the mesh size, such as the number of triangles and vertices, and the number of segments required in the segmentation. Overall, for a mesh with about 16000 triangles, the time is about 30 seconds for the initial segmentation generation, 0.5 seconds for the sparse reconstruction of each segment, and 0.1 seconds for the linear integer programming. The computation time can be significantly improved by parallel implementation.

Dataset: We employ the dataset used in [KHS10], as it provides both segmentation and labeling information. This dataset is adapted from the Princeton benchmark [CGF09], which has 380 meshes of 19 categories. We pre-compute shape descriptors for each segment in the database to be used in the segmentation algorithm.

Segmentation results: Similar to [KHS10], we also test our segmentation results with respect to the human segmentation results in [CGF09]. When segmenting a 3D model in the database, we remove that model from the top 10 query results if it is present, like leave-one-out validation, to fully test our algorithm (Fig.1). Figure 7 illustrates the statistics of all measures developed in Chen et al. [CGF09] for 380 models in the benchmark. The four scores of our segmentation algorithms are better than geometry property based segmentation algorithms, such as Random cuts and Shape diameter function. It is not surprising since our algorithm can be viewed as to choose the segments from such segmentation algorithms to maximize the similarity score. In terms of rand index and consistency error, our results are better then SB3, which is the segmentation algorithm in [KHS10] with 3 training models for each category, and slightly worse than learning with 19 training models each category, i.e. SB19 (We use the scores of rand index and consistency error for SB3 and SB19 provided in [KHS10]). However, such a downgrade of segmentation scores does not influence the visual quality as demonstrated in the experiments. Considering the saving of computational load in training time, our algorithm can be a reasonable choice for the task of shape segmentation.

Robustness: We first test the robustness of L1 reconstruction against L2, as illustrated in Fig. 4a. For most models, L1 reconstruction plus linear programming produce segmentation results more consistent with the examples, for instance, the leg of the eye-glasses and the tail of the airplane. In Fig. 4b, we compare the similarity computed by L1 reconstruction and Euclidean distance of shape features, where the closest Euclidean distance between an input segment and example segments is used as the similarity measure. Similarly,

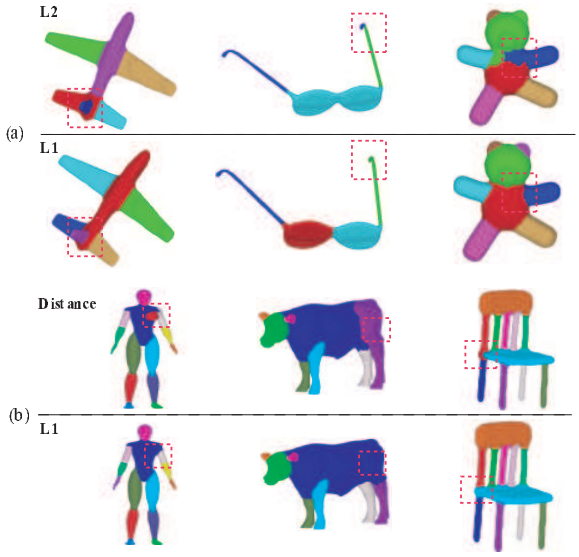

Figure 4: (a) The segmentation results with L1 and L2 reconstruction using the linear programming algorithm. Top row: Results using L2 reconstruction . Bottom row: Results using L1 Reconstruction. (b) The segmentation results with L1 reconstruction and the closest shape feature Euclidean distance induced segment similarity. Top row: Results using Euclidean distance. Bottom row: Results using L1 Reconstruction. L1 norm is able to pick up segments more similar to those in database.

the distance based similarity results in inconsistent segmentation results for human, bull and chair models. It demonstrates that the similarity induced by L1 regularization can encourage the algorithm to pick up segments more similar to those in database, which is critical to the success of transductive segmentation. Fig. 5 illustrates that sparse reconstruction is robust to outliers in the database. We intentionally select six models as the database and test our algorithm with five meshes. The results show that our algorithm can correctly select the segments for a test mesh to form a segmentation similar to the segmentation of its most similar shapes in the database. This property is particularly useful when we start with a small database, since, under this situation, it is of high probability that the top query results contain shapes that are not similar to the input shape.

Categories combination: As our algorithm forms segmentation at segment level, it possesses the ability to combine segments from shapes of different categories to obtain the segmentation result. This enables the algorithm to segment cross-category shapes. Fig. 6 illustrates the segmentation result of the sphinx and a man with handcraft model, which can be viewed as a combination of human, animal or vehicle category. For the sphinx model, the number of segments is set to be 9 using the number constraint. Since the pose-oblivious signature is global, which is challenging for

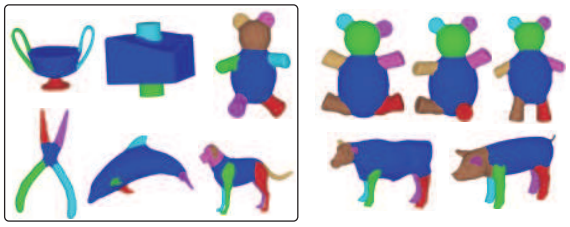

Figure 5: Robustness to outliers. Left: The designed database. Right: Transductive segmentation results with this database. Even though there are only two shapes in database similar to the shapes to be segmented, our L1 sparse reconstruction based segmentation algorithm can still generate similar segmentation results.

Human	81.1%	Cup	96.0%	Glasses	92.7%
Airplane	86.9%	Ant	87.8%	Chair	92.2%
Octopus	97.4%	Table	91.3%	Teddy	92.3%
Hand	82.2%	Plier	82.7%	Fish	87.4%
Bird	82.1%	Armadillo	69.8%	Bust	62.3%
Mech	92.9%	Bearing	96.2%	Vase	81.5%
Fourleg	71.9%	Average	85.6%		

Table 1: Labeling: Recognition rate

it to query by parts, we thus manually select a set of models from the dataset to be the database for the sphinx model. Afterwards, our algorithm can successfully segment out the face and limbs, while the hierarchical fuzzy cut algorithm in [KT03] fails to extract semantic parts. In contrast, for the man with handcraft model, the database examples are from the shape retrieval results. Even though the results are not similar to the input model, our algorithm still correctly captures the basic structure of the man, which mimics the segment structure of the animals in the examples.

Labeling accuracy: The recognition rate of our algorithm is listed in Table 1. The semantic labeling results for the dataset are obtained by the labeling algorithm in Sec. 4.3 after segmentation, and we count recognition rate with the classification error equation in [KHS10]:

$$E = \left(\sum_i a_i I(c_i, c_i^*)\right) / \left(\sum_i a_i\right) \quad (9)$$

where a_i is the area of triangle i in the dataset, c_i its ground truth label, and c_i^* our labeling result. I is a boolean function to count the correctly labeled triangles. Without training, the recognition rate of our algorithm is close to the result of the well-trained model SB6, which is 89.4%, in [KHS10]. Our recognition results should be able to be improved by employing sophisticated contextual features [SSS*10].

Limitation: Our segmentation results are influence by the randomcuts algorithm in initial segmentation. For an animal model, if there is no neck segment in the initial segments generated by randomcuts using concavity as the affinity measure, the neck segment can not be transferred from

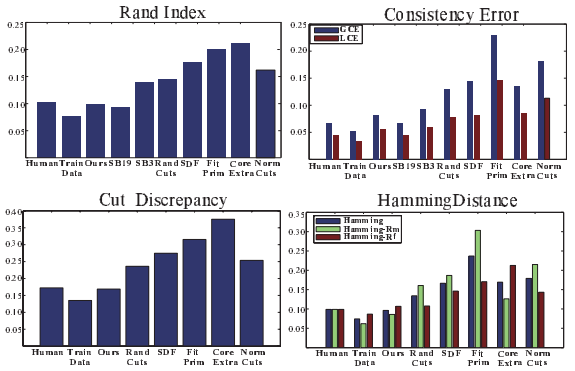

Figure 7: Segmentation evaluations using the measures in [CGF09]. Our approach represents the evaluation scores of our algorithm. Other segmentation algorithms: SB3, SB19 - Learning to segment algorithm in [KHS10], 3 and 19 training models per category. Randcuts-Random cuts. SDF - Shape diameter function. FitPrim - Fitting primitives [AKM*06]. CoreExtra - Core extraction [KLT05]. NormCuts - [GF08].

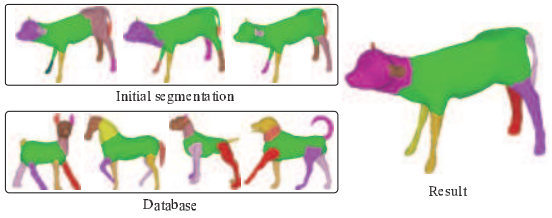

Figure 8: Failure case. Without neck segments in the initial segmentations, our algorithm can not segment out the neck region from an input animal model.

the database to the new input shape, as illustrated in Fig.8. In addition, the number of segments is still difficult to be automatically determined by the energy function in Eq. 2 for complicated models. For example, the number of segments is set to be 9 to obtain the final result in Fig. 6 for the Sphinx model. We speculate that it is because that the shape descriptors are not distinctive enough to eliminate all non-meaningful segments.

6. Conclusion

We have developed a transductive segmentation algorithm which aims to transfer the segmentation in database to an input new shape. Sparse reconstruction error is used as an efficient way to measure the segment similarity and guide the linear binary integer programming algorithm to select the similar segments to form the final valid segmentation. Combined with state-of-the-art 3D shape query method, our algorithm has potential to handle large scale shape dataset due to its simplicity and ability to handle heterogenous 3D shape data.

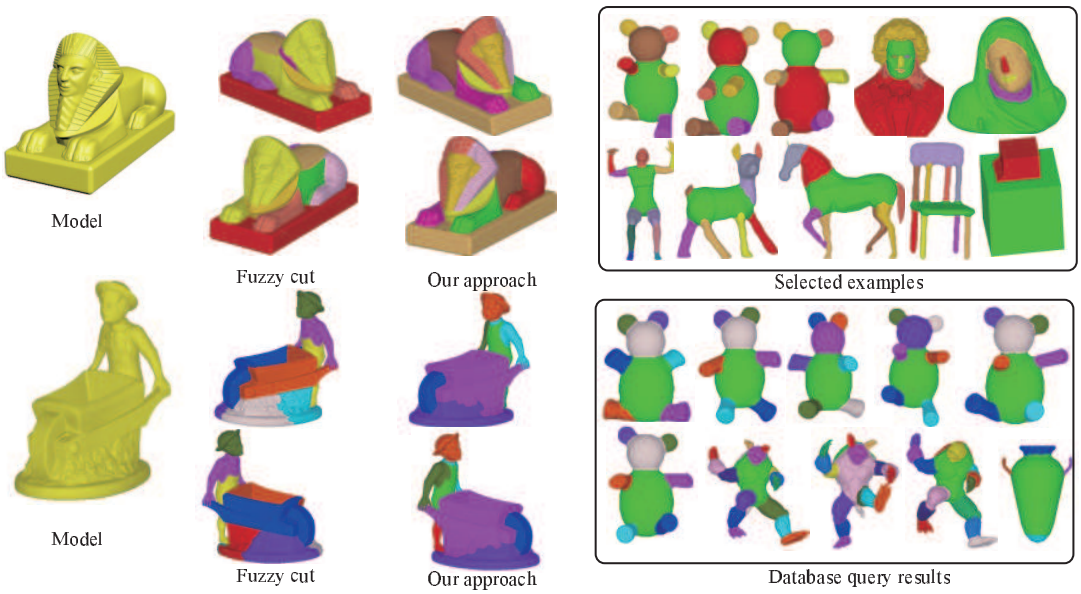

Figure 6: The segmentation result for two models: sphinx and man with handcart . Our algorithm can segment out parts, such as face, body and legs, similar to the segments in the manually selected database.

In future, we plan to investigate how group sparsity constraint can help to improve the segmentation and labeling accuracy. We are also interested in the application of sparsity constraints in various geometry processing algorithms, such as 3D modeling and geometry compression.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments; This work is partially supported by NSFC (No. 61322204, 61379079, 61272392, 61202207 and 61210005). Mingliang Xu is partially supported by NSTS Program (Grant 2013BAH23F01). Zhenming Yuan is partially supported by Provincial NSFC (No. Z12F020027). Weiwei Xu is partially supported by open project of State Key Lab of CAD&CG (A1307).

References

[AKM*06] ATTENE M., KATZ S., MORTARA M., PATANE G., SPAGNUOLO M., TAL A.: Mesh segmentation - a comparative study. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006* (Washington, DC, USA, 2006), SMI '06, IEEE Computer Society, pp. 7–. 2, 7

[ASGCO10] AVRON H., SHARF A., GREIF C., COHEN-OR D.: l1 sparsee reconstruction of sharp point set surfaces. *ACM Trans. Graph. 29*, 5 (Nov. 2010), 135:1–135:12. 3

[BBGO11] BRONSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J., OVSJANIKOV M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30*, 1 (Feb. 2011), 1:1–1:20. 3

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3d mesh segmentation. *ACM Trans. Graph. 28*, 3 (July 2009), 73:1–73:12. 1, 2, 6, 7

[CLW*11] CHENG B., LIU G., WANG J., HUANG Z., YAN S.: Multi-task low-rank affinity pursuit for image segmentation. In *ICCV* (2011), pp. 2439–2446. 3

[CYH13] CHEN F., YU H., HU R.: Shape sparse representation for joint object classification and segmentation. *Image Processing, IEEE Transactions on 22*, 3 (2013), 992–1004. 3

[FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 652–663. 1

[FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A search engine for 3d models. *ACM Trans. Graph. 22*, 1 (Jan. 2003), 83–105. 3

[GB08] GRANT M., BOYD S.: Cvx: Matlab software for disciplined convex programming, 2008. 5

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1 (Jan. 2006), 130–150. 1

[GF08] GOLOVINSKIY A., FUNKHOUSER T.: Randomized cuts for 3d mesh analysis. *ACM Trans. Graph. 27*, 5 (Dec. 2008), 145:1–145:12. 2, 3, 7

[GF09] GOLOVINSKIY A., FUNKHOUSER T. A.: Consistent segmentation of 3d models. *Computers & Graphics 33*, 3 (2009), 262–269. 2

[GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (Mar. 2007), 261–271. 3

[HFL12] HU R., FAN L., LIU L.: Co-segmentation of 3d shapes via subspace clustering. *Comp. Graph. Forum 31*, 5 (Aug. 2012), 1703–1713. 2

[HJS*14] HUANG J., JIANG T., SHI Z., TONG Y., BAO H., DEBRUN M.: L1-based construction of polycube maps from complex shapes. *ACM Trans. Graph.* (2014). 3

[HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint shape segmentation with linear programming, Dec. 2011. 1, 2, 4

[HS13] He L., Schaefer S.: Mesh denoising via l0 minimization. *ACM Trans. Graph. 32*, 4 (July 2013), 64:1–64:8. 3

[HWAG09] Huang Q.-X., Wicke M., Adams B., Guibas L. J.: Shape decomposition using modal analysis. *Comput. Graph. Forum 28*, 2 (2009), 407–416. 2, 3

[HWCO*13] Huang H., Wu S., Cohen-Or D., Gong M., Zhang H., Li G., Chen B.: L1-medial skeleton of point cloud. *ACM Trans. Graph. 32*, 4 (July 2013), 65:1–65:8. 3

[KCKK12] Kalogerakis E., Chaudhuri S., Koller D., Koltun V.: A probabilistic model for component-based shape synthesis. *ACM Trans. Graph. 31*, 4 (July 2012), 55:1–55:11. 2

[KG00] Karni Z., Gotsman C.: Spectral compression of mesh geometry. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 279–286. 1

[KHS10] Kalogerakis E., Hertzmann A., Singh K.: Learning 3d mesh segmentation and labeling. *ACM Trans. Graph. 29*, 4 (July 2010), 102:1–102:12. 1, 3, 6, 7

[KLT05] Katz S., Leifman G., Tal A.: Mesh segmentation using feature point and core extraction. *The Visual Computer 21*, 8-10 (2005), 649–658. 7

[KT03] Katz S., Tal A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph. 22*, 3 (July 2003), 954–961. 1, 2, 7

[L04] Liu R., 0002 H. Z.: Segmentation of 3d meshes through spectral clustering. In *Pacific Conference on Computer Graphics and Applications* (2004), IEEE Computer Society, pp. 298–305. 2

[LCY*09] Liu X., Cheng B., Yan S., Tang J., Chua T. S., Jin H.: Label to region by bi-layer sparsity priors. In *Proceedings of the 17th ACM International Conference on Multimedia* (New York, NY, USA, 2009), MM '09, ACM, pp. 115–124. 3

[LeG03] Luo J., en Guo C.: Perceptual grouping of segmented regions in color images. *Pattern Recognition 36*, 12 (2003), 2781 – 2792. 4

[LGQ08] Li X., Gu X., Qin H.: Surface matching using consistent pants decomposition. In *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling* (New York, NY, USA, 2008), SPM '08, ACM, pp. 125–136. 2

[LHMR08] Lai Y.-K., Hu S.-M., Martin R. R., Rosin P. L.: Fast mesh segmentation using random walks. In *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling* (New York, NY, USA, 2008), SPM '08, ACM, pp. 183–191. 2

[LLL07] Lin H. Y. S., Liao H. Y. M., Lin J. C.: Visual salience-guided mesh decomposition. *Trans. Multi. 9*, 1 (Jan. 2007), 46–57. 2

[LW08] Lavoué G., Wolf C.: Markov random fields for improving 3d mesh analysis and segmentation. In *3DOR* (2008), pp. 25–32. 2

[LWTH01] Li X., Woon T. W., Tan T. S., Huang Z.: Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics* (New York, NY, USA, 2001), I3D '01, ACM, pp. 35–42. 1

[MW99] Mangan A. P., Whitaker R. T.: Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (Oct. 1999), 308–321. 2

[OFCD02] Osada R., Funkhouser T., Chazelle B., Dobkin D.: Shape distributions. *ACM Trans. Graph. 21*, 4 (Oct. 2002), 807–832. 4

[SFCH12] Shen C.-H., Fu H., Chen K., Hu S.-M.: Structure recovery by part assembly. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 180:1–180:11. 1

[Sha08] Shamir A.: A survey on mesh segmentation techniques. *Comput. Graph. Forum 27*, 6 (2008), 1539–1556. 2

[SKS06] Simari P., Kalogerakis E., Singh K.: Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2006), SGP '06, Eurographics Association, pp. 111–119. 2

[SM00] Shi J., Malik J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 8 (2000), 888–905. 3

[SSCO08] Shapira L., Shamir A., Cohen-Or D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4 (2008), 249–259. 1, 4

[SSS*10] Shapira L., Shalom S., Shamir A., Cohen-Or D., Zhang H.: Contextual part analogies in 3d objects. *Int. J. Comput. Vision 89*, 2-3 (Sept. 2010), 309–326. 1, 7

[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. Graph. 30*, 6 (Dec. 2011), 126:1–126:10. 1, 2

[TWC*13] Tong T., Wolz R., Coupé P., Hajnal J. V., Rueckert D.: Segmentation of mr images via discriminative dictionary learning and sparse coding: Application to hippocampus labeling. *NeuroImage 76* (2013), 11–23. 3

[WAvK*12] 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. 31*, 6 (Nov. 2012), 165:1–165:10. 2

[WGW*13] Wang Y., Gong M., Wang T., Cohen-Or D., Zhang H., Chen B.: Projective analysis for 3d shape segmentation. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 192:1–192:12. 2

[WWS*13] Wu Z., Wang Y., Shou R., Chen B., Liu X.: Unsupervised co-segmentation of 3d shapes via affinity aggregation spectral clustering. *Computers & Graphics 37*, 6 (2013), 628–637. 2

[WYL*14] Wang R., Yang Z., Liu L., Deng J., Chen F.: Decoupling noises and features via weighted l1-analysis compressed sensing. *ACM Trans. Graph.* (2014). 3

[XZCOC12] Xu K., Zhang H., Cohen-Or D., Chen B.: Fit and diverse: Set evolution for inspiring 3d shape galleries. *ACM Trans. Graph. 31*, 4 (July 2012), 57:1–57:10. 2

[YWSZ13] Yuan Y., Wu F., Shao J., Zhuang Y.: Image annotation by semi-supervised cross-domain learning with group sparsity. *J. Visual Communication and Image Representation 24*, 2 (2013), 95–102. 3

[ZSGS04] Zhou K., Synder J., Guo B., Shum H.-Y.: Isocharts: Stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (New York, NY, USA, 2004), SGP '04, ACM, pp. 45–54. 1

[ZZD*11] Zhang S., Zhan Y., Dewan M., Huang J., Metaxas D. N., Zhou X. S.: Deformable segmentation via sparse shape representation. In *MICCAI (2)* (2011), Fichtinger G., Martel A. L., Peters T. M., (Eds.), vol. 6892 of *Lecture Notes in Computer Science*, Springer, pp. 451–458. 3

[ZZWC12] Zhang J., Zheng J., Wu C., Cai J.: Variational mesh decomposition. *ACM Trans. Graph. 31*, 3 (June 2012), 21:1–21:14. 1