3D Articulated Shape Segmentation Using Motion Information

Emre Kalafatlar and Yücel Yemez
Multimedia, Vision and Graphics Laboratory, Koç University, Istanbul, Turkey
{ekalafatlar,yyemez}@ku.edu.tr

Abstract

We present a method for segmentation of articulated 3D shapes by incorporating the motion information obtained from time-varying models. We assume that the articulated shape is given in the form of a mesh sequence with fixed connectivity so that the inter-frame vertex correspondences, hence the vertex movements, are known a priori. We use different postures of an articulated shape in multiple frames to constitute an affinity matrix which encodes both temporal and spatial similarities between surface points. The shape is then decomposed into segments in spectral domain based on the affinity matrix using a standard K-means clustering algorithm. The performance of the proposed segmentation method is demonstrated on the mesh sequence of a human actor. [1]

1 Introduction

3D shape segmentation is a key step for processing and understanding of digitally acquired mesh models of 3D objects and has numerous applications in computer graphics and vision such as skeleton extraction and model deformation, mesh morphing, gesture recognition, shape retrieval, compression and collision detection [4]. While most of the existing 3D shape segmentation methods consider only static meshes, dynamic shape models of moving objects are becoming more and more commonplace with the recent advances in 3D acquisition techniques. A typical example of this trend is the use of fixed connectivity mesh sequences to represent human actors with articulated motion. In this paper, we present an automatic method to segment an articulated shape given in the form of a dynamic mesh sequence by incorporating motion information so as to further improve the segmentation results obtained by static segmentation methods.

Many algorithms have been proposed to decompose a static 3D mesh into meaningful segments, as surveyed and comparatively evaluated by Chen et al. in [4]. The method of random walks for example, as proposed by Lai et al. [9], first asks for manual selection of a user specified number of seed faces (or automatically selects them), and then assigns each non-seed face to the seed for which it has the highest probability to reach after a random walk on the dual graph. The method of fitting primitives, which is an effective segmentation method proposed by Antenne et al. [2], hierarchically clusters faces by merging them according to the best fitting primitive shape at each iteration. Another hierarchical decomposition method, suggested by Golovinskiy et al. [5], uses a randomized approach to find minimum cuts. Shlafman et al. [15] have described a decomposition algorithm for morphing polyhedral surfaces, which applies K-means clustering to the matrix of pairwise face distances composed of angular and Euclidean distances. Katz et al. [8] have improved this idea by replacing Euclidean distances with geodesic distances between adjacent faces and differentiating the relative weight of convex and concave dihedral angles as well as by avoiding jaggy boundaries with the use of fuzzy clustering instead of K-means. Liu et al. [11] have computationally simplified the problem using spectral clustering, where the basic idea is to embed the data into a lower dimensional space by using the same distance matrix as Katz et al. [8] and then to perform K-means clustering. All these methods apply to static meshes. However, in a number of cases, especially for articulated shapes, depending on the posture of the model, it becomes almost impossible to extract sufficient information to identify the cut regions. When the benchmark results obtained in [4] are investigated for example, one can observe various segmentation failures. In Figure 1, we see an example of how the algorithms compared in [4] may fail to locate some parts of the given surface mesh, e.g. the elbows on the arms, due to the straight stance of the arm and the resulting smooth surface from wrist to shoulder.

In this work, we propose to incorporate the mo-

[1] This work has been supported by TUBITAK under the project EEEAG-109E274.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.877

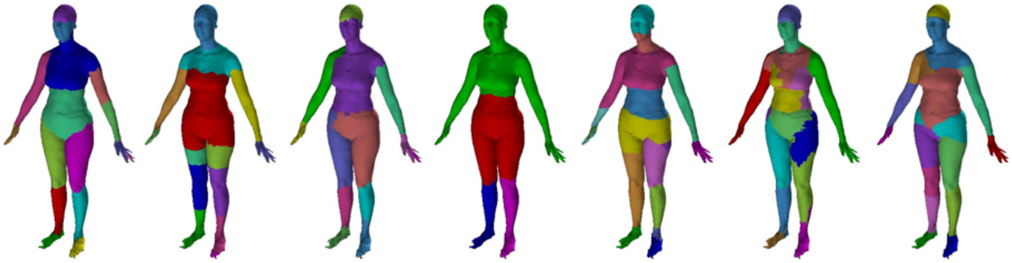

Figure 1. Segmentation results for a static human model extracted from [4]. From left to right, Randomized Cuts [5], Shape Diameter Function [14], Normalized Cuts [5], Core Extraction [7], Random Walks [9], Fitting Primitives [2] and K-Means [15].

tion information, whenever available, to circumvent the problems encountered in segmentation of static meshes such as the one demonstrated above. To this effect, we describe a segmentation method that is applicable to mesh sequences with articulated motion. By analyzing multiple postures of an articulated shape and positions of different parts of the mesh in multiple frames, we deduce information about the semantic relations between different parts of the shape. We assume that semantically different parts of an articulated object exhibit different rigid motions. So far this idea has been exploited for mesh segmentation only a few times, such as in [10] and [1]. Our approach differs from existing methods in that we use spatial information in the form of angular distance and temporal information both at the same time for dynamic mesh segmentation.

2 Algorithm

Because of the high computational cost and memory requirement of analyzing multiple frames for a time varying model, as a first step we downsample the vertices of the mesh sequence, which is of fixed connectivity, to a desired number of base vertices. The base vertices, computed by using the Dijkstra's shortest paths algorithm as suggested in [6], are enforced to be distributed equally on the surface of the mesh (see Figure 2). Every base vertex has its own base area, called a patch, including a number of non-base vertices. In order to determine the likelihood of two base vertices to be in the same cluster, we constitute a distance matrix $D(i,j)$ based on the matrix used by Katz et al. [8] as described in the sequel.

We denote the geodesic distance between two base vertices b_i and b_j by $D_{\mathrm{g}}(i,j)$ and the angular distance by $D_{\mathrm{a}}(i,j)$. The angular distance $D_{\mathrm{a}}(i,j)$ is calculated as follows:

$$D_{\mathrm{a}}(i,j) = \frac{1}{L}\sum_{l} \eta(1-\cos\theta(i,j,l)) \qquad (1)$$

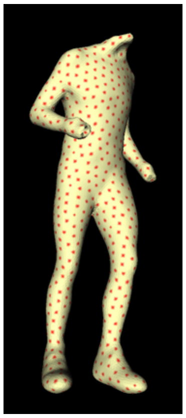

Figure 2. Base vertices computed over the surface of a given mesh model.

where $\theta(i,j,l)$ is the angle between the normals of the base vertices b_i and b_j in frame l, and L is total number of frames in the sequence. The normal of a base vertex is computed as the average of the normals of the vertices within the patch of that base vertex. Since a concave angle indicates a higher probability of belonging to a cut region, $\eta = 1$ is used for concave angles whereas $\eta = 0.01$ is used for convex angles.

In order to incorporate the motion information, we also introduce a motion distance term, denoted by $D_{\mathrm{m}}(i,j)$, which holds the variance of pairwise Euclidian distances of the base vertices over the mesh sequence (note that, in the case of articulated motion, the Euclidean distance between two surface points belonging to the same segment is expected to remain almost constant):

$$D_{\mathrm{m}}(i,j) = \mathrm{Var}\left(\frac{d_{\mathrm{E}}(i,j,l)}{\tilde{d}_{\mathrm{E}}(i,j)}\right) \qquad (2)$$

where $d_{\mathrm{E}}(i,j,l)$ is the Euclidean distance between b_i and b_j in frame l, and $\tilde{d}_{\mathrm{E}}(i,j)$ denotes its average over all frames. During variance calculation, only some equal number of smallest and largest Euclidian distances are taken into account so as to eliminate the frames in which the pairs of base vertices stay relatively stationary. Note that we assume the inter-frame vertex correspondences are given a priori such as in the form of fixed connectivity. Moreover, prior to calculating the variances as well as the angular and geodesic terms, pairwise distance values are normalized in order to emphasize relative distance variations instead of absolute changes. Given above definitions, the overall distance matrix is then given by the following weighted summation:

$$D(i,j) = \alpha\frac{D_{\mathrm{g}}(i,j)}{\tilde{D}_{\mathrm{g}}} + \beta\frac{D_{\mathrm{a}}(i,j)}{\tilde{D}_{\mathrm{a}}} + \gamma\frac{D_{\mathrm{m}}(i,j)}{\tilde{D}_{\mathrm{m}}} \quad (3)$$

where α, β and γ, $\alpha+\beta+\gamma=1$, are weights used to adjust the relative importance of the three different distance values and are currently set manually to find an optimum blend. The factors $\tilde{D}_{\mathrm{g}}$ and $\tilde{D}_{\mathrm{a}}$ are the average values computed over the entries of the corresponding distance matrices for adjacent base vertices whereas $\tilde{D}_{\mathrm{m}}$ is the average value computed over all entries of $\tilde{D}_{\mathrm{m}}$.

In order to achieve clustering on the distance matrix, there exist various schemes in the literature, among which we favor spectral clustering for its simplicity and computational efficiency [12]. Our clustering implementation is quite similar to the one used in [11]. As a starting point, we construct an $N \times N$ affinity matrix by transforming the distance matrix via the following exponential kernel:

$$W(i,j) = \begin{cases} e^{-D(i,j)/2\sigma^2} & \text{if } A(i,j)=1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where N is the number of the vertices of the mesh sequence and $A(i,j)$ is the adjacency matrix that encodes the adjacency of the base vertices. To properly set the parameter σ, we again stick to [11] and calculate its value with a slight modification using $(1/10N^2)\sum_{i,j} D(i,j)$. The next step is to embed the shape coordinates into a lower K-dimensional space [12, 11]. For this purpose we calculate the normalized symmetric Laplacian $\mathbf{L}$ of the affinity matrix $\mathbf{W}$ by $\mathbf{L} = \mathbf{\Lambda}^{-1/2}\mathbf{W}\mathbf{\Lambda}^{-1/2}$, where $\mathbf{\Lambda}$ is a diagonal matrix whose i^{th} diagonal element is the sum of the i^{th} row of $\mathbf{W}$. Let $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K$ be the eigenvectors of $\mathbf{L}$, corresponding to the largest K eigenvalues. We then compute the $N \times K$ matrix $\mathbf{V}$ containing the eigenvectors as columns. The rows of $\mathbf{V}$, when normalized so as to have unit norm, provide us with the K dimensional points in the spectral domain.

Polarization Theorem [3] states that the projection of data points to a lower rank K amplifies the unevenness in the point distribution on the K dimensional sphere. This implies that the points of high affinity will be grouped together while the points of low affinity will move apart. Therefore, clustering of the data embedded on a lower dimensional space will be simple enough to obtain by employing a K-means clustering technique.

The initialization of cluster means is an important problem for all K-means clustering techniques. Our algorithm uses the number of clusters, K, as an input parameter to set manually. Given the number of clusters K, we initially find K vertices scattered as evenly as on the mesh surface by utilizing the same method used to find the base vertices. A standard K-means clustering is then performed on the points in spectral domain, using these K vertices as initial cluster centers.

3 Results

In order to demonstrate the performance of our algorithm, we have used the “Jumping Man” mesh sequence with fixed connectivity, consisting of 220 frames that exhibit the real jumping act of a human actor [13]. To show the improvement obtained by incorporating the motion information, we have performed two segmentation experiments. In the first one, we have performed static segmentation only on a single frame of the sequence. In other words we set the parameter γ to zero and make use of the angular distances which belong only to a single frame so as to completely discard the motion information. In the second experiment, we have performed the segmentation on the whole mesh sequence as proposed. The parameters α, β and γ are set manually to obtain the optimum performance. In Figure 3, we display the results of the two segmentation experiments. The left column shows the segmentation result with $\alpha = 0.15$, $\beta = 0.85$ and $\gamma = 0$, hence including no motion information. The middle column displays the result of the segmentation process with $\alpha = 0.15$, $\beta = 0.55$ and $\gamma = 0.3$. The rightmost column is the ground truth segmentation that we have obtained by labeling the model based on the manual segmentation boundaries as provided in [4] for human models. Note that the ground-truth segmentation looks jaggy due to the patchwork structure. We observe that, with the inclusion of the motion information, the segment boundaries move towards the real joints, bringing the segmentation closer to the ground truth.

In addition to subjective visual comparison, we have also performed a quantitative evaluation of the segmentation results. To this effect, we compute a segmentation error which is based on the cut discrepancy metric described in [4]. The error is calculated by summing the distances between wrongly segmented base vertices and the closest base vertex to them in their desired segments defined by the ground truth segmentation. This

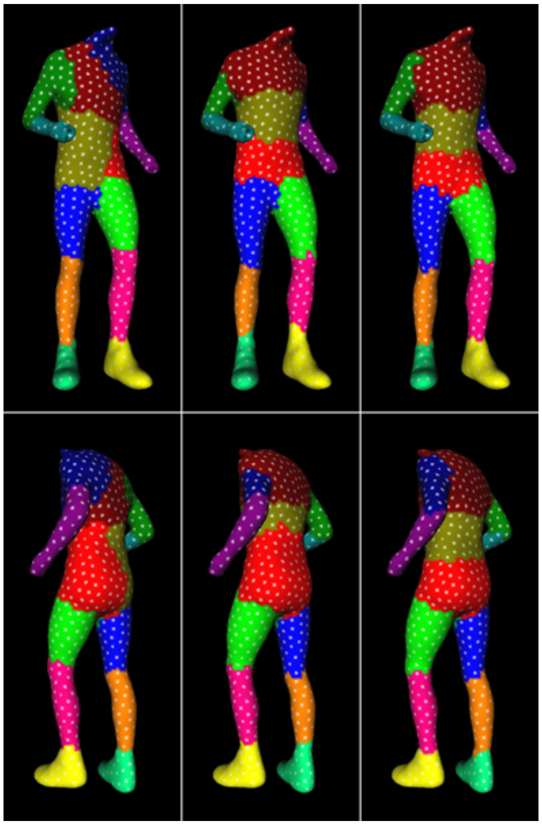

Figure 3. Segmentation results. Left column: Without motion information. Middle column: Using motion information. Right column: Ground truth.

summation is then divided by the average Euclidian distance from each base vertex to the centroid of the mesh in order to rule out different scales of different models. The quantitative evaluation is also in accordance with our visual observation that the proposed algorithm improves the segmentation results obtained by the static approach. We have found that under the given parameter setting, the incorporation of motion information has reduced the segmentation error by almost a factor of 3 and also decreased the number of the wrongly clustered base vertices by %38.

4 Conclusion

We have described a new method to improve the segmentation of time-varying mesh models of articulated 3D shapes. Our experiments have shown that incorporation of motion information significantly enhances the segmentation results obtained via static segmentation and yields a shape segmentation which is in more accordance with human intuition. The current drawback of our algorithm is that, it has a number of parameters which we set manually, including blending ratios of distance values and the number of clusters. Our future plan is to advance the algorithm in order to set these parameters without user intervention and to fully automate the segmentation process.

References

[1] R. Arcila, S. K. Buddha, F. Hétroy, F. Denis, and F. Dupont. A framework for motion-based mesh sequence segmentation. *International Conference on Computer Graphics, Visualization and Computer Vision, WSCG 2010*, Feb. 2010.

[2] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput.*, 22(3):181–193, 2006.

[3] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering, 2003.

[4] X. Chen, A. Golovinskiy, , and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. Graph.*, 28(3), Aug. 2009.

[5] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3d mesh analysis. *ACM Trans. Graph.*, 27(5):1–12, 2008.

[6] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. SIGGRAPH*, pages 203–212, 2001.

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

[8] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *Proc. SIGGRAPH*, pages 954–961, 2003.

[9] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin. Fast mesh segmentation using random walks. In *ACM Symposium on Solid and Physical Modeling*, pages 183–191, 2008.

[10] T.-Y. Lee, Y.-S. Wang, and T.-G. Chen. Segmenting a deforming mesh into near-rigid components. *Vis. Comput.*, 22(9):729–739, 2006.

[11] R. Liu and H. Zhang. Segmentation of 3d meshes through spectral clustering. In *Proc. Pacific Graphics*, pages 298–305, 2004.

[12] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[13] P. Sand, L. McMillan, and J. Popović. Continuous capture of skin deformation. *ACM Trans. Graph.*, 22(3):578–586, 2003.

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

[15] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Computer Graphics Forum*, pages 219–228, 2002.