Manifold Learning and its Applications:
Papers from the AAAI Fall Symposium (FS-09-04)

Mesh Segmentation Using Laplacian Eigenvectors and Gaussian Mixtures

Avinash Sharma, Radu Horaud, David Knossow and **Etienne von Lavante**
PERCEPTION group, INRIA Grenoble Rhone-Alpes
655, avenue de l'Europe
38334 Saint Ismier Cedex, France
avinash.sharma@inrialpes.fr

Abstract

In this paper a new completely unsupervised mesh segmentation algorithm is proposed, which is based on the PCA interpretation of the Laplacian eigenvectors of the mesh and on parametric clustering using Gaussian mixtures. We analyse the geometric properties of these vectors and we devise a practical method that combines single-vector analysis with multiple-vector analysis. We attempt to characterize the projection of the graph onto each one of its eigenvectors based on PCA properties of the eigenvectors. We devise an unsupervised probabilistic method, based on *one-dimensional* Gaussian mixture modeling with model selection, to reveal the structure of each eigenvector. Based on this structure, we select a subset of eigenvectors among the set of the smallest non-null eigenvectors and we embed the mesh into the isometric space spanned by this selection of eigenvectors. The final clustering is performed via unsupervised classification based on learning a *multi-dimensional* Gaussian mixture model of the embedded graph.

1. Introduction

Spectral clustering methods use graph representations of the data and solve for graph partitioning within the context of spectral graph theory. Early spectral approaches recursively compute the normalized cut (Shi and Malik 2000) over the graph using the first non-null Laplacian eigenvector (also known as the Fiedler vector (Chung 1997)) and are referred to as *spectral bi-partitioning (SB) methods*. It has bee noticed that this does not guarantee *good* clusters as the normalized cut is computed recursively irrespective of the global structure of the data (Belkin and Niyogi 2003).

Recent spectral approaches use the k smallest non-null eigenvectors of the Laplacian matrix (or one of its variants) to optimally embed the graph onto a k dimensional subspace (Belkin and Niyogi 2003), and to cluster the embedded graph nodes into k groups. Various grouping strategies may be used, such as direct extensions of SB to multiple eigenvectors, i.e., greedy ordering heuristic (Alpert, Kahng, and Yao 1999) or K-means (Ng, Jordan, and Weiss 2002; Belkin and Niyogi 2003; Yu and Shi 2003). In (Zelnik-manor and Perona 2004; Fischer and Poland 2005; Nadler and Galun 2006) a number of limitations of spectral clustering are analysed, thus focusing on the problems of noise, density and scale variations in the data.

However the use of these spectral clustering algorithms cannot be generalized to any type of graphs. In the case of *sparse* graphs with *uniform* connectivity, there is no obvious optimal graph partitioning solution, namely the extraction of a number of strongly connected components that are only weakly interconnected. Indeed, the Laplacian matrices of such graphs cannot be viewed as *slightly perturbed* matrices of the ideal case (between-cluster similarity is exactly 0) because of the lack of a meaningful eigengap (Luxburg 2007). As a consequence, the estimation of the dimension of the embedding (and hence of the number of clusters) based on eigenvalue analysis (Chung 1997; Ng, Jordan, and Weiss 2002; Luxburg 2007) has several drawbacks when one deals with sparse graphs whose vertex connectivity is almost uniform across the graph. First, there is no eigengap in such cases and therefore it is not straightforward to estimate the dimension of the spectral embedding in a completely unsupervised way. Second, the eigenvalues of any large semi-definite positive symmetric matrix are estimated only approximately; this means that it is not easy to study the eigenvalues' multiplicities (which play a crucial role in the analysis of the Laplacian eigenvectors (Biyikoglu, Leydold, and Stadler 2007)) and that ordering the eigenvectors based on these estimated eigenvalues is not reliable (Mateus et al. 2008). This has dramatic consequences if one seeks some form of repeatability when clustering similar but not identical sets of data.

Empirically it has been observed by us that more than one perceptually prominent cluster of a graph is projected onto a single eigenvector, contrary to the assumption of the ideal case, where a single eigenvector represents an indicator vector for an individual cluster. Therefore, a clustering algorithm based on these geometric properties, should combine only those eigenvectors that best reveal the graph's clusters.

In this paper we use the geometric properties of the Laplacian eigenvectors and we devise a new spectral clustering algorithm well suited for graphs with uniform connectivity, such as meshes; more precisely, we attempt to character-

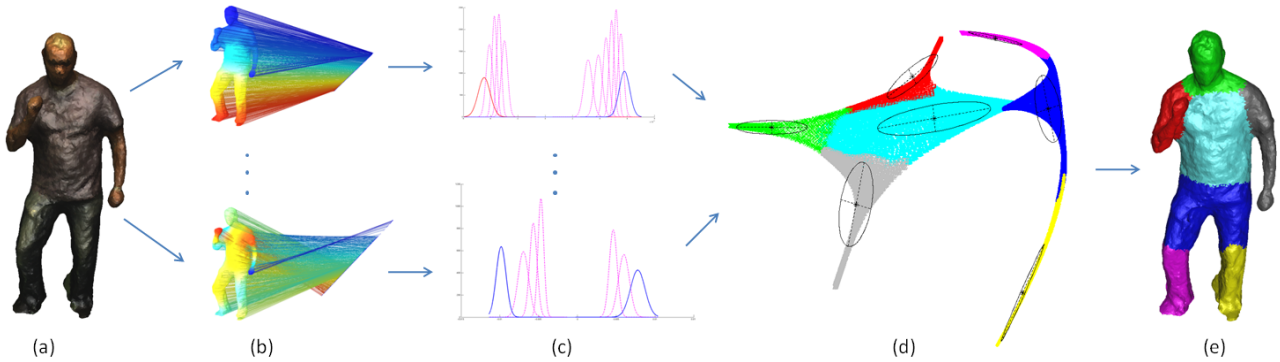


Figure 1: A mesh with approximately 17,000 vertices and with an average of six edges per vertex is (a) projected onto ten eigenvectors (b) corresponding to the ten smallest eigenvalues of the normalized Laplacian. Once the vertices with zero eigenfunction values are removed (see section 3below) we fit 1D Gaussian mixtures with an optimal BIC criterion (c). The eigenvector values associated with the left most (negative values) and right most (positive values) components of these Gaussian mixtures are selected as potential cluster seeds on the extrema of the mesh. Whenever such a cluster corresponds to a connected component, the corresponding eigenvector is selected and considered for the embedding. In this example the method selected 5 eigenvectors. A 5-dimensional Gaussian mixture with 7 components is fitted to this embedding (d) thus segmenting the mesh into 7 clusters (e).

ize the one-dimensional (1D) *projections* of a graph onto its eigenvectors to build a practical algorithm based on the interpretation of the eigenvectors as the *principal components of the graph* (Fouss et al. 2007). We devise an unsupervised probabilistic method, based on 1D Gaussian mixtures with model selection, to reveal the structure of the graph projections on its eigenvectors. Based on this, we show that we are able to select a subset of the set of eigenvectors corresponding the the smallest eigenvalues of the Laplacian. These selected eigenvectors are then used to embed the graph. We show how this eigenvector-by-eigenvector analysis allows to initialize the clustering that is carried out either with a non-parametric method (hierarchical clustering, K-means) or with Gaussian mixtures. The advantage of the latter is that it can be combined with the Bayesian information criterion (BIC) (Fraley and Raftery 2002) to estimate the optimal number of clusters, when this number cannot be provided by eigenvalue analysis, i.e., the existence of an eigengap.

We apply our method to meshes of complex shapes such as articulated bodies with several protrusions, e.g., figure 1. We seek natural segmentations of these shapes such that clusters correspond to body parts. We observe that our algorithm provides perceptually meaningful clustering and that finer body-part details (hands, legs, thighs, hips, torso, head, etc.) correspond to finer segmentations, i.e. a simple increase in the number of clusters.

In practice and unlike traditional discrete geometry approaches to mesh processing (which use cotangent weights (Pinkall and Polthier 1993; Lévy 2006; Reuter et al. 2009)) we use a Gaussian kernel: each mesh vertex describes a 3D point lying onto a surface and each graph edge connects two nearby vertices. The results shown in this paper use real data. Indeed, multiple camera technologies available today are able to process 2D data (pixels) and to provide 3D meshes (Franco and Boyer 2009), (Zaharescu and Horaud 2009). Real-data meshes, such as the ones used in this paper, are more challenging than meshes obtained with simulated data (as is often the case in computer graphics and computer-aided design): the discrete set of vertices is not regularly distributed over the surface, there is a large vertex-distribution variability within the same class of perceptually similar shapes or when the shape is deforming itself, and there are meaningful topological changes.

The organization of the paper is as follows. Section 2. briefly summarizes a few mathematical properties of graph Laplacian and discusses properties of the Laplacian eigenvectors along with a strategy to characterize these properties. In section 3. we propose a clustering algorithm. Section 4. describes mesh segmentation results before concluding the paper in section 5.

2. Properties of Laplacian Eigenvectors

We consider an undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with a node set $\mathcal{V} = \{V_1, \ldots, V_N\}$ and an edge set $\mathcal{E} = \{E_{ij}\}$. We consider the *graph Laplacian*: $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where the entries of matrix $\mathbf{W}$ are given by $W_{ij} = \exp(-d_{ij}^2)/\sigma^2$ whenever there is an edge E_{ij} and 0 otherwise, and by the diagonal matrix $\mathbf{D}$ with $D_{ii} = \sum_{j=1}^{N} W_{ij}$. The volume of the graph is defined by $\mathrm{Vol}(\mathcal{G}) = \mathrm{Trace}\,(\mathbf{D})$ and d_{ij} denotes the distance between two graph vertices. The value of σ is estimated automatically for each graph dataset based on the median value of the d_{ij} distances.

$\mathbf{L}$ is a semi-definite positive symmetric matrix. Hence, the geometric multiplicity of any eigenvalue is equal to the algebraic multiplicity of that eigenvalue. We will simply refer to *eigenvalue multiplicity*. The eigenvalues of $\mathbf{L}$ are $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$. Without loss of generality we

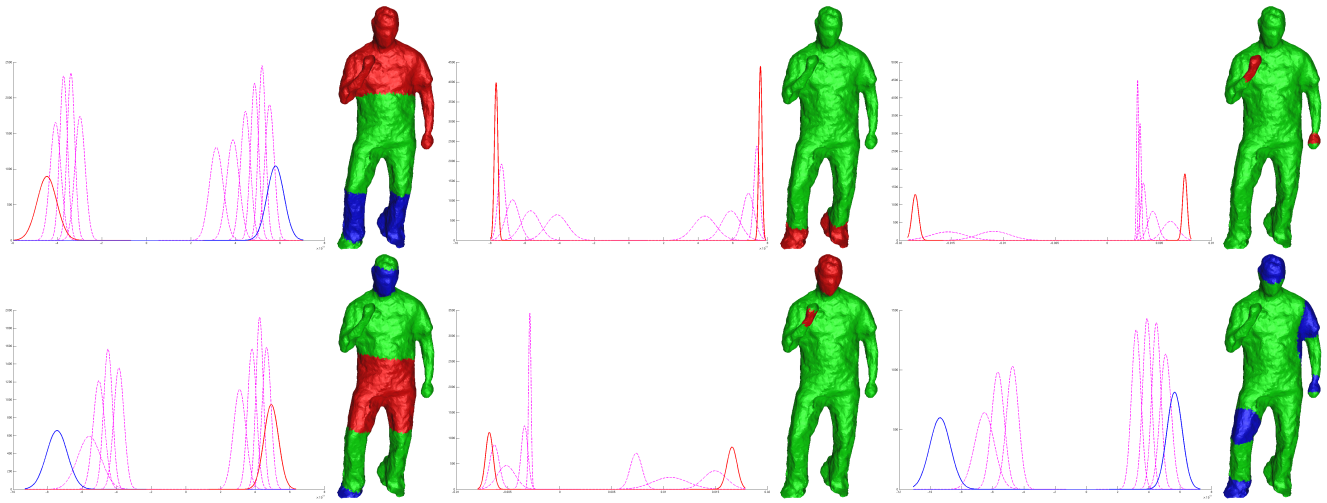


Figure 2: 1D Gaussian clusters along the left- and right-side of Laplacian eigenvectors $\boldsymbol{U}_2$ to $\boldsymbol{U}_7$. The leftmost and the rightmost clusters of each eigenvector are either represented with red color, if the corresponding vertex set belongs to a single connected component, or with blue color, if the corresponding vertex set belongs to several connected components. In this case the algorithm selected eigenvectors $\boldsymbol{U}_2$ to $\boldsymbol{U}_6$, while $\boldsymbol{U}_7$ was left out.

will only consider the case of connected graphs, i.e., λ_1 has multiplicity equal to one and $\lambda_2 > 0$.

The null space of $\mathbf{L}$ is the constant unit eigenvector $\boldsymbol{U}_1 = (1/\sqrt{N} \dots 1/\sqrt{N})^\top$. The remaining eigenvectors are denoted by $\boldsymbol{U}_2, \dots, \boldsymbol{U}_N$. All the eigenvectors are normalized and mutually orthogonal: $\boldsymbol{U}_i \boldsymbol{U}_j = \delta_{ij}$. Let $\mathbf{L}_t = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the *truncated* eigendecomposition where the null eigenvalue and constant unit eigenvector were removed. Hence $\mathbf{U} = [\boldsymbol{U}_2 \dots \boldsymbol{U}_N]$ is a $N \times (N-1)$ matrix and $\mathbf{\Lambda} = \text{Diag}[\lambda_2 \dots \lambda_N]$ is a $(N-1) \times (N-1)$ diagonal matrix. We also denote by $\mathbf{L}^+$ the pseudo-inverse of the Laplacian that satisfies the Moore-Penrose conditions (Golub and Van Loan 1989). The *truncated* eigendecomposition of the pseudo-inverse is $\mathbf{L}_t^+ = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^\top$.

We summarize a number of useful properties of the Laplacian eigenvectors. From above, one can immediately notice that for any eigenvector $\boldsymbol{U}_i = (U_i(V_1) \dots U_i(V_N))^\top$, $2 \leq i \leq N$ we have:

$$\sum_{j=1}^{N} U_i(V_j) = 0 \tag{1}$$

as well as:

$$-1 < U_i(V_j) < 1 \tag{2}$$

The above notation, i.e., $U_i(V_j)$ is preferred to the matrix-entry notation U_{ij} because it emphasizes the fact that an eigenvector is an eigenfunction of $\mathbf{L}$ that maps the vertices of the graph onto a bounded interval on the real line: $\boldsymbol{U}_i : \mathcal{V} \rightarrow (-1; +1)$.

2.1 PCA Interpretation of Laplacian Eigenvectors

Another important concept that will be used by our algorithm is the principal component analysis (PCA) of a graph (Fouss et al. 2007). We denote by $\boldsymbol{X}_j$ a row of matrix $\mathbf{U}$ and hence matrix $\mathbf{X} = \mathbf{U}^\top$ has as columns the Euclidean coordinates of the N graph vertices. From (1) we obtain that the embedded representation is centered irrespective of the choice of the eigenvectors used for the embedding:

$$\sum_{j=1}^{N} \boldsymbol{X}_j = \mathbf{0} \tag{3}$$

The vertex covariance matrix is $\mathbf{X}\mathbf{X}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}$. Another possible embedding is $\mathbf{Y} = \mathbf{\Lambda}^{-1/2}\mathbf{X}$ and the vertex covariance matrix is in this case: $\mathbf{Y}\mathbf{Y}^\top = \mathbf{\Lambda}^{-1/2}\mathbf{X}\mathbf{X}^\top\mathbf{\Lambda}^{-1/2} = \mathbf{\Lambda}^{-1}$.

Therefore, the eigenvectors of $\mathbf{L}_t$, $\boldsymbol{U}_2, \boldsymbol{U}_3, \dots, \boldsymbol{U}_{p+1}$ associated with its p smallest eigenvalues $\lambda_2, \lambda_3, \dots, \lambda_{p+1}$ can be interpreted as the p principal components of the embedded coordinates of the graph's vertices $\{\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_i, \dots, \boldsymbol{Y}_N\}$. Hence, the eigenvectors of the Laplacian matrix $\mathbf{L}$ associated with its smallest non-null eigenvalues can be interpreted as the axes of maximal variance of the embedded graph.

2.2 A Heuristic for Eigenvector Selection

The intuition behind the PCA of a graph is that two different connected components that are farther away from each other should project onto the positive and negative extremities of one of the eigenvectors. It is intuitive to select only those eigenvectors which have a 1D cluster of vertices either at its positive or at its negative extremity. Notice that not *all* of the smallest eigenvectors of a graph feature significant clusters at their extremities. This suggests that spectral clustering may include some form of eigenvector selection based on the availability of 1D clusters at their extremities.

3. The Proposed Clustering Algorithm

PCA interpretation of the Laplacian embedding suggest that the projection of a graph onto each one of its eigenvectors could provide interesting information about the structure of the graph associated with the data, as discussed above. Indeed, connected components project as one-dimensional clusters along the eigenvectors; Moreover, these connected components are more likely to project towards the extremities of the eigenvectors rather than towards their centers, i.e, at the lower and upper bounds of the open interval $]-1;+1[$. This suggests that one can detect 1D clusters along the eigenvectors, select the leftmost and rightmost ones as initialization for the graph clustering, as it can be seen in Figure 2.

Traditionally, the latter is performed by K-means which uses as many clusters as the number of smallest eigenvectors. In our algorithm, we select a set of q eigenvectors which have identifiable 1D clusters at their extremities and we embed the graph in the isometric space spanned by these q eigenvectors. Then we perform clustering in this q-dimensional space. An open issue with our algorithm is how to choose the number of clusters. For this purpose we combine GMM clustering with an optimality criterion for model selection, i.e., the Bayesian information criterion (BIC). For practical reasons, we remove the vertices with nearly zero eigenfunction value while performing 1-D Gaussian analysis. This allows GMM to capture small scale clusters along the extremities of an eigenvector.

Consequently, we suggest an algorithm that first performs 1D clustering along the p smallest eigenvectors. Second, it detects clusters found at the extremities of these vectors. Third, it performs a simple connectivity analysis in order to determine whether each one of these extremity-clusters belong to a connected component.This allows us to select a subset of $q \leq p$ eigenvectors that are well suited to embed the graph. Finally we fit a q-dimensional Gaussian mixture to the embedded data and we apply an optimality criterion for model selection.

This yields the algorithm outlined below.

Clustering algorithm:

1. Compute the first p non null eigenvectors of the graph Laplacian, $[\boldsymbol{U}_2, \ldots, \boldsymbol{U}_{p+1}]$.
2. For each eigenvector $\boldsymbol{U}_i$:
   (a) Remove the vertices with eigenfunction value close to zero.
   (b) Perform 1D clustering using GMM with optimal model selection.
   (c) Choose the outer clusters, i.e., the leftmost one and the rightmost ones.
3. Perform connectivity analysis of all the vertex sets associated with the outer clusters, thus providing an embedding of size $q \leq p$.
4. Embed the graph in the space spanned by the selected eigenvectors and fit a q-dimensional GMM. Select the number of clusters based on BIC.

The main differences between the proposed algorithm and standard spectral clustering are the followings. The size of the embedding is not governed any more by the detection of an eigengap. Instead, 1D GMM allows a completely unsupervised selection of a set of eigenvectors well suited to embed the graph. K-means is replaced with GMM with model selection.

4. Mesh Segmentation

We illustrate the method developed in this paper with 3D meshes of articulated objects. Such a mesh corresponds to graphs that are sparse and with regular local connectivity. Clusters should correspond to object parts, protrusions (hands, legs, head) as well as torso and hips.

We obtain results on meshes extracted from real data, which is more challenging compared to most of the synthetic meshes in terms of topology variations. We compare our results with the standard spectral clustering approach presented in (Ng, Jordan, and Weiss 2002). Our algorithm was not able to find any optimal BIC value for deciding the number of components *i.e.* k in selected subspace, *e.g.*, Figure 3. We tried different values for k and here we show results with $k = 5, 7, 9, 11, 13$.

In the case of standard spectral clustering implementation we use the k-dimensional subspace while finding k segments. In our approach we consider a set of $p = 10$ eigenvectors for selecting a subset of eigenvectors, independent of value of k. The average dimension of our selected subspace was 5. In the examples shown here the coloring of indentified segments is not consistent over the individual images. However, we can still easily observe the consistency of results *w.r.t.* segment boundaries.

Figure 4 compares our results with the standard approach. Each column of the figure represents segmentation for an human articulated mesh with with different values of k. The first row of the figure corresponds to segmentations obtained by the standard spectral clustering approach, which applies K-means to the embedding defined by the first k non-null eigenvectors. The second row presents the results of our approach. In column Figure 4(a),(b) the standard approach completely failed in assigning distinct parts with same label color, while in columns (c-d) the segmentation results by our approach are more consistent and perceptually meaningful.

Figure 5 shows a comparison of the segmentation results by the standard and our approach at $k = 11$. Here, we can see that our approach segments the mesh in perceptually meaningful parts (working with 5-dimensional embeddings in most of the examples shown here) as compared to the standard approach (which uses k dimensional space). As

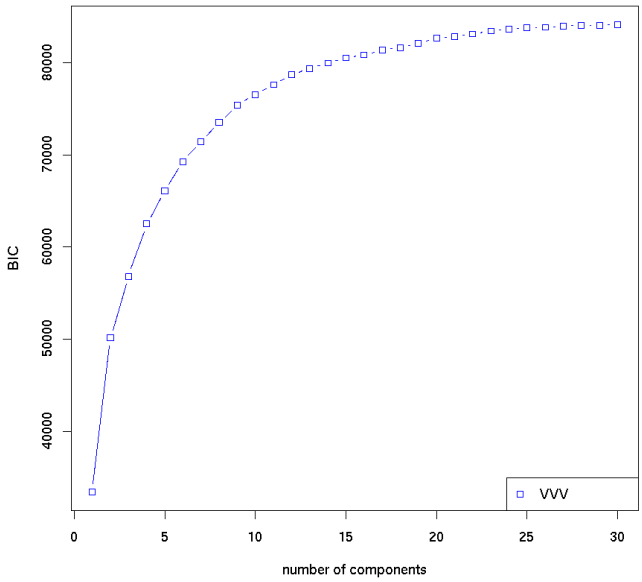


Figure 3: BIC plot as a function of number of components (k) for the mesh shown in figure 4.

(a) k=5 (b) k=7 (c) k=9 (d) k=11 (e) k=13 (f) k=15

Figure 4: Segmentation results obtained with the graphs of an articulated shape and with different values of k. Colors encode different clusters. But the coloring is not the same for all the segmentations. The top row shows the segmentation obtained with standard spectral clustering. The bottom row shows the results obtained with our algorithm.

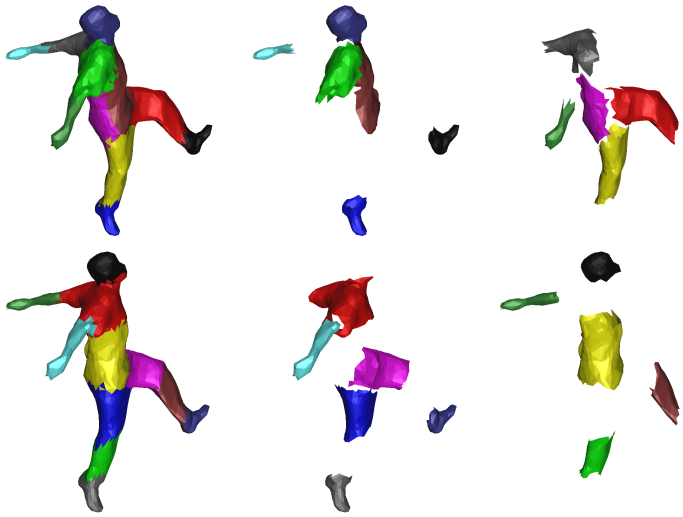
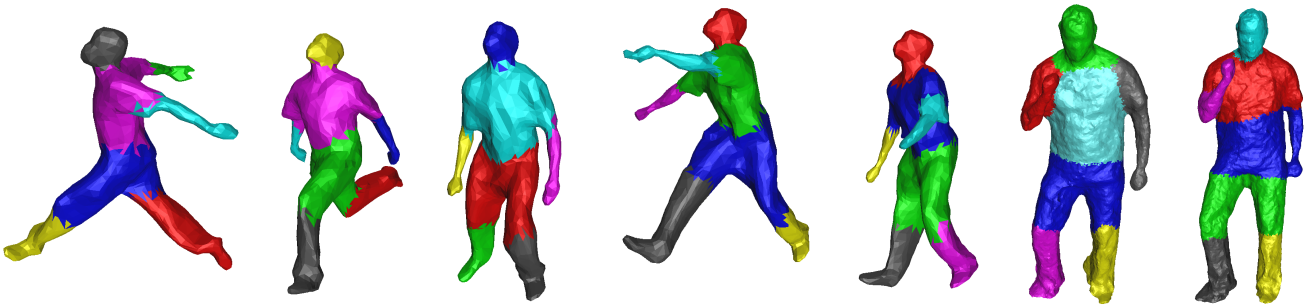


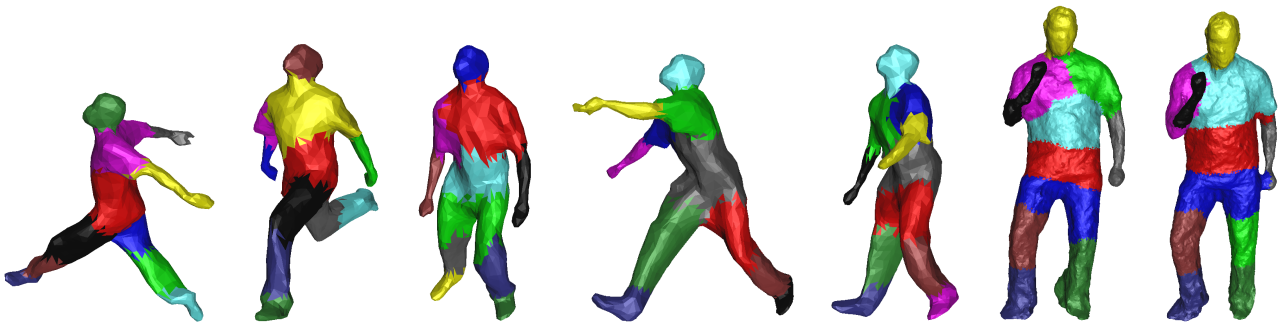
Figure 5: This figure shows how standard and our clustering method splits an articulated mesh. In the first row the segmentation obtained by standard approach segments the torso vertically while our approach in bottom row segments the shape into perceptually meaningful body parts. Here $k = 11$

.



(a) (b) (c) (d) (e) (f) (g)

Figure 6: Segmentation results obtain with our approach over different articulated real meshes with $k = 7$. Here the different colors are used to represent different parts of the mesh. But this coloring is not part-wise consistent over different poses and subjects.



(a) (b) (c) (d) (e) (f) (g)

Figure 7: Segmentation results obtain with our approach over different articulated real meshes with $k = 11$. Notice that as the number of clusters is increased from 7 to 11, we still obtain a segmentation in terms of semantically and perceptually meaningful parts.

shown in Figure 4 and 5, it is quite clear that the segmentation obtained by our multidimensional GMM approach is more consistent than standard spectral clustering based on the analysis presented in previous sections.

In Figure 6, we present the segmentation results with our approach on real meshes with different articulated poses, with subject variation, with no vertex correspondences and with $k = 7$. In the last column 6(g) we have shown a result on a difficult topolog: a mesh having a cycle. Although our approach is not completely robust to such major topological changes, we are still able to get consistent segmentation results except for the parts which are directly involved in cycle.

In Figure 7 we present segmentation results on different articulated real data meshes with $k = 11$. The first five examples involve meshes taken from the so-called flashkick sequence (Starck and Hilton 2007), with an average of 1500 vertices. The last pair of results correspond to the meshes of the Ben data sequence with (approx.) 17000 vertices.

5. Conclusions

A novel spectral clustering algorithm based on a detailed analysis of geometric properties of the Laplacian eigenvectors has been proposed in this paper. More specifically, we devised an unsupervised probabilistic method, based on Gaussian mixtures with model selection, which reveals the structure of each eigenvector. This enables us to select a subset of eigenvectors among the smallest eigenvectors of a graph, to embed the graph in the space spanned by this selection, and to cluster the graph's vertices using a multidimensional GMM with model selection. When applied to graphs corresponding to meshes of articulated objects, such as humans, our method segments the latter into perceptually meaningful parts.

6. Acknowledgments

We are thankful to Dr. Yve Colin de Verdiere and Visesh Chari for the insightful discussions with them and for providing important feedback on our work.

References

Alpert, C. J.; Kahng, A. B.; and Yao, S. 1999. Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics* 90(1-3):3–26.

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15:1373–1396.

Biyikoglu, T.; Leydold, J.; and Stadler, P. F. 2007. *Laplacian Eigenvectors of Graphs*. Springer.

Chung, F. 1997. *Spectral Graph Theory*. American Mathematical Society.

Fischer, I., and Poland, J. 2005. Amplifying the block matrix structure for spectral clustering. In *Technical Report, IDSIA*.

Fouss, F.; Pirotte, A.; Renders, J.; and Saerens, M. 2007. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *Knowledge and Data Engineering* 19(3):355–369.

Fraley, C., and Raftery, A. E. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association* 97:611–631.

Franco, J. S., and Boyer, E. 2009. Efficient polyhedral modeling from silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(3):414–427,.

Golub, G. H., and Van Loan, C. 1989. *Matrix Computations*. The Johns Hopkins University Press.

Lévy, B. 2006. Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *SMI*.

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

Mateus, D.; Horaud, R.; Knossow, D.; Cuzzolin, F.; and Boyer, E. 2008. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *CVPR*.

Nadler, B., and Galun, M. 2006. Fundamental limitations of spectral clustering. In *NIPS*.

Ng, A.; Jordan, M.; and Weiss, Y. 2002. On spectral clustering: analysis and an algorithm. In *NIPS*.

Pinkall, U., and Polthier, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 2:15–36.

Reuter, M.; Biasotti, S.; Giorgi, D.; Patanè, G.; and Spagnuolo, M. 2009. Discrete laplace-beltrami operators for shape analysis and segmentation. *SMI*.

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

Starck, J., and Hilton, A. 2007. Surface capture for performance based animation. *IEEE Computer Graphics and Applications* 27(3):21–31.

Yu, S. X., and Shi, J. 2003. Multiclass spectral clustering. In *ICCV*.

Zaharescu, A., and Horaud, R. 2009. Robust factorization methods using a gaussian/uniform mixture model. *International Journal of Computer Vision* 81(3):240–258.

Zelnik-manor, L., and Perona, P. 2004. Self-tuning spectral clustering. In *NIPS*.