Robust Spectral 3D-Bodypart Segmentation Along Time

Fabio Cuzzolin, Diana Mateus, Edmond Boyer, and Radu Horaud

INRIA Rhone-Alpes, 655 avenue de l'Europe, 38334 Montbonnot, France
Fabio.Cuzzolin@inrialpes.fr, Diana.Mateus@inrialpes.fr,
Edmond.Boyer@inrialpes.fr, Radu.Horaud@inrialpes.fr

Abstract. In this paper we present a novel tool for body-part segmentation and tracking in the context of multiple camera systems. Our goal is to produce robust motion cues over time sequences, as required by human motion analysis applications. Given time sequences of 3D body shapes, body-parts are consistently identified over time without any supervision or *a priori* knowledge. The approach first maps shape representations of a moving body to an embedding space using locally linear embedding. While this map is updated at each time step, the shape of the embedded body remains stable. Robust clustering of body parts can then be performed in the embedding space by k-wise clustering, and temporal consistency is achieved by propagation of cluster centroids. The contribution with respect to methods proposed in the literature is a totally unsupervised spectral approach that takes advantage of temporal correlation to consistently segment body-parts over time. Comparisons on real data are run with direct segmentation in 3D by EM clustering and ISOMAP-based clustering: the way different approaches cope with topology transitions is discussed.

1 Introduction

Human motion analysis is an important topic in computer vision with many applications in surveillance, human machine interface and animation, among others. Such analysis, when based on image observations, relies on the ability to extract body motion information from images. This problem has received a considerable attention from the community over the last decades [1]. The existing approaches mainly differ in the amount and type of information that is known in advance. *Model based* approaches, e.g. [2,3,4], assume a known, often kinematic, model for the human body and recover parameters of this model in a joint space using image evidence. However, the joint space is generally high dimensional, making the search for model parameters difficult without adequate initializations, an issue sometimes solved by stochastic sampling. *Learning based* approaches, e.g. [5,6,7,8] directly relate visual information to learned body configurations, without the need for an intermediate model. While solving the initialization issue, these approaches are anyway limited by the classes of examples used for training.

A. Elgammal et al. (Eds.): Human Motion 2007, LNCS 4814, pp. 196–211, 2007.

In opposition, approaches have been proposed that directly infer body poses for markerless motion capture from multiple image cues, in particular volume sequences [9,10,11]. This includes *skeletonization* methods that recover the intrinsic articulated structure of 3D shapes, either directly in 3D, e.g. [12], or in an embedded space, e.g. [13,14]. A nice feature with embeddings is the ability to map 3D shapes onto low-dimensional manifolds in higher dimensional spaces, thus naturally revealing the intrinsic structure of an articulated shape. A critical issue, though, is the presence of topological ambiguities raised by self contacts (joint hands, for instance). This was noticed by Sundaresan and Chellappa [14] who used an *a priori* graphical body model to resolve these ambiguities.

In this work, we propose an approach that segment body-parts in 3D body shape sequences. We consider this an intermediate step towards a robust human motion analysis framework without any *a priori* information nor learned examples, as demonstrated in this paper. Our approach uses spectral embedding to map shapes onto low-dimensional manifolds which are then clustered into body-parts. One crucial innovation is that we take advantage of the correlation between information along time sequences to segment in a consistent way. Our goal is to guarantee robustness, in particular to topological ambiguities that occur over time. Recent attempts to extend nonlinear reduction to spatio-temporal data [15,16] provide indeed elegant solutions to enforce temporal relationships when embedding time sequences. Unfortunately, such relationships are not easily identified with the dense shape representations of moving bodies that we consider. Instead, we propose to enforce temporal consistency through clustering in the embedded space, where clusters are remarkably stable under articulated motions and there propagated over time. Although several spectral embedding methods could be, in principle, considered for that purpose [17,18], we chose *Local Linear Embedding* [19] (LLE) which exhibits better performances in our specific scenario. LLE is fast, maps a shape to a low-dimensional manifold in the embedded space and is already partially robust to topology changes as it depends on the local structure of the data.

The rest of the paper is organized as follows: after motivating the choice of clustering in time in an embedding space (Section 2), we present each step of the algorithm in Section 3: the use of *k-wise* clustering to segment the embedded cloud (3.1), starting from detected branch terminations (3.2); how to propagate cluster seeds over time to ensure consistency (3.3) and merge/split them according to the current topology of the body (3.4). In an extensive experimental section (5) we present results on unsupervised segmentation over a large number of dense voxelset sequences, we compare them with segmentation in 3D by EM clustering, show how to learn the parameters of the algorithm from the data, and discuss the way different approaches cope with topology transitions.

2 Motivation: Clustering After Locally Linear Embedding

Embedding techniques are interesting for clustering purposes because of their characteristic of "amplifying" the separation between different parts of the same

3D shape. This is true in particular for Locally linear Embedding (LLE) [19], but also holds for other spectral methods like Laplacian Eigenmaps.

LLE is an unsupervised learning algorithm which computes d-dimensional embeddings Y of sets of input points $X = \{x_i, i = 1, .., N\}$ living in a nonlinear manifold, while preserving their local structure (i.e. the distances between each point and its k neighbors). Groups of local neighborhoods belonging to a same part of the shape are "redistributed" by the algorithm along distinct chains. Figure 1-middle shows how legs and arms of a human body are much well

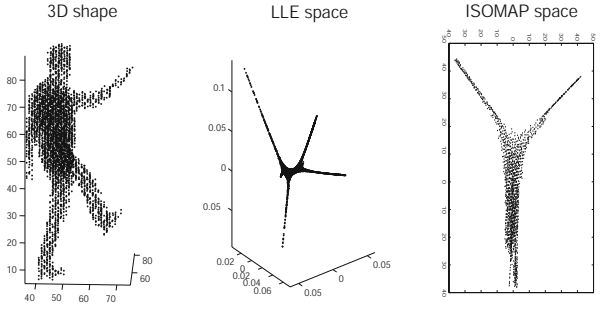

Fig. 1. How LLE (middle) and ISOMAP (right) map the same 3D cloud (left) for the same number of neighbors $k = 13$. Arms (lower appendices) are indistinguishable in the ISOMAP space.

separated in the LLE embedding space than in 3D (left). This is much less true, however, for methods (like ISOMAP [18], right) based on geodesic distances. Clustering in the embedding space is then typically easier than in 3D.

LLE, in particular, has some edges over other embedding schemes: for obvious reasons it is less sensitive (with respect to geodesic-based embeddings) to *changes in the topology* of the moving body (as we show in Section 5.4). It is computationally less expensive than ISOMAP. Finally, as we will show in Section 3.1, the lower dimension of the embedded clouds it generates makes them suitable to be clustered in a more robust way using *k-wise clustering* [23].

2.1 Clustering Along Time and Pose-Invariance

When clustering sequences, though, we need the segmentation obtained at different time instants to be *consistent*. A desirable cue, in this sense, is the fact that some embedding schemes (like ISOMAP) are inherently *pose-invariant* under articulated motion (as while the articulated body evolves geodesic distances between pairs of points do not change). This is not true, in a strict sense, for LLE. However, as its embedding depends only on the local structure of the input dataset, it is reasonable to conjecture that under articulated motion the shape of the LLE embedded cloud would exhibit remarkable stability. An articulated object is formed by a number of rigid bodies linked by a small number of joints:

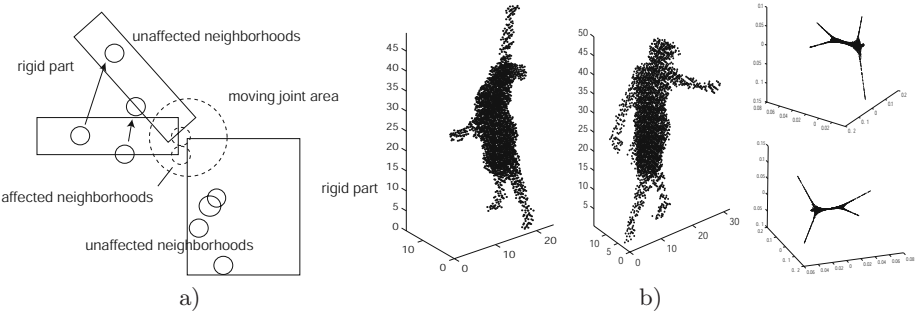

Fig. 2. a) The number of neighborhoods affected by articulated motion is relatively small. b) Some anecdotal evidence on the stability of locally linear embedding under articulated motion. Different poses (left, middle) of the same articulated body are mapped to the same embedded cloud (right), for a large interval of parameter values.

Clearly all local neighborhoods incident on a rigid part are preserved along the motion, while only the few neighborhoods interested by the evolving joint(s) are affected (Figure 2-a).

The validity of this claim depends of course on the number of points N in the dataset, the neighborhood size k (as smaller ks reduce the number of neighborhoods with non-empty intersection with moving joints), and last but not least the number of evolving joints.

As an example let us consider two different poses of the same articulated body, for instance a dancer performing a ballet, represented as voxelsets (Figure 2-b, middle and left). Figure 2-b right shows the related embedded clouds obtained through LLE for $d = 3$ and $k = 10$. Their similarity is apparent. Analogous results can be obtained for a wide range of values of the critical parameter k.

Stability and other desirable properties are actually shared by other embedding schemes like, for instance, Laplacian Eigenmaps [17]. In the following we will make reference in particular to LLE.

3 Approach

3.1 K-Wise Clustering in the Embedded Shape

Let us first focus on the problem of segmenting an embedded cloud at a given time instant. As we mentioned above (and as Figures 1 and 2 confirm) for $d = 3$ the embedded cloud is (for a wide interval of values of k) a tree-like one-dimensional curve. It is then natural to look for clusters formed by sets of roughly collinear embedded points. With a few exceptions, clustering algorithms (like k-means [20]) are based on the assumption that a pairwise measure of distance between data-points is available. As every pair of data-points trivially defines a line, however, there does not exist a useful measure of similarity between such pairs. It is instead possible to define measures of similarity over *triplets* of points to indicate how close they are to being collinear (Figure 3-a) [21,22]).

In general the problem of clustering points based on similarity between k-tuples of points is called *k-wise clustering*. An interesting approach to k-wise clustering has been proposed in [23]. Consider a set of datapoints $V = \{v_i, i = 1, ..., N\}$.

K-Wise Clustering Algorithm

1. In the first step an *affinity hypergraph* is built. A weighted undirected hypergraph H is a pair (V, h), where V is the set of vertices of H, and subsets z of V of size k are called hyperedges. The function h associates nonnegative weights $h(z)$ with each hyperedge (k-tuple) $z = \{v_{j_1}, ..., v_{j_k}\}$, and measures the affinity of each hyperedge.
2. Then, a weighted graph $G = (V, g)$ that approximates the hypergraph H is constructed by constrained least square optimization, based on the assumption that $h(z) = \sum_{v_i, v_j \in z; i<j} g(v_i, v_j)$, i.e. the weight of each hyperedge is the arithmetic mean of the weights of the edges of G incident on it (*clique averaging*).
3. Finally, to partition the approximating graph G into k parts a spectral clustering algorithm is adopted that uses the first k eigenvectors of the normalized Laplacian of the graph and performs k-means clustering on the resulting k-dimensional embedding [24,25].

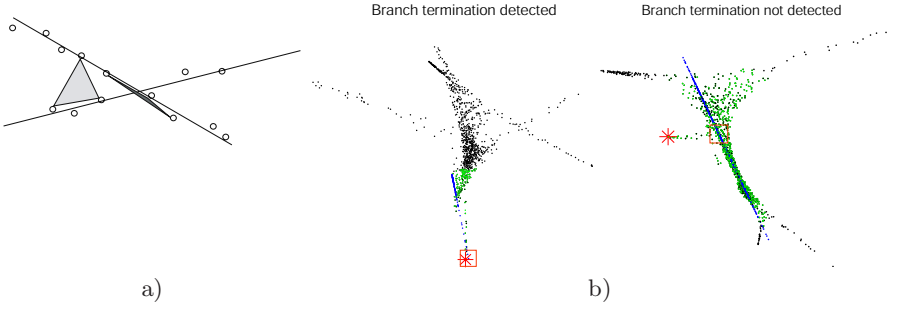

Fig. 3. a) 3-wise (k-lines) clustering. Areas of triangles defined by triads of points measure their collinearity. The smaller the area (dark triangle) the greater the collinearity. b) Termination (left) and internal (right) points of the embedded cloud are characterized by the fact that their projection (red square) on the line (in blue) interpolating their neighborhoods (in green) is an extremum of the interval of all projections.

In our case, the hypergraph to approximate has as set of vertices the embedded cloud $V = Y$, and hyperedges formed by d elements (for a d-dimensional embedding space). Specifically, these hyperedges are triads of points for $d = 3$. A natural choice for the affinity of these triads is then the area of the triangle they form (the volume of the $d-1$-dimensional hyperedge in the general case).

The application of the k-wise clustering algorithm to the embedded cloud yields a segmentation in the embedding space that can be trivially remapped

back to the original 3D space, using the ordering of the data-points. Notice that, as step 3. of the k-wise clustering algorithm involves standard k-means, and the latter requires a set of initial seeds to start clustering from, seeds are required to initialize the overall algorithm as well. We will address this issue in Section 3.3. It remains to decide the number of clusters for a given embedded shape.

3.2 Branch Detection and Number of Clusters

The fact that embedded clouds typically appear as one-dimensional strings formed by a number of branches (corresponding to the extremities of the moving body) provides us with a simple method to estimate, at each given time instant, the "correct" number of clusters (Figure 3-b).

Each point of the embedded cloud is tested to decide whether or not it is a branch termination. This test is performed by finding its nearest neighbors (for a certain threshold distance which can be empirically learned from the data), plotted in green. The best interpolating line for all neighbors is then found (in blue), and all neighbors projected on it. A point of the embedded cloud (red star in Figure 3-b) is a branch termination if the projection of all its neighbors on the interpolating line lay on one side of its own projection (red square) like in Figure 3-b-left, it is not when the projection has neighbors on both sides (Figure 3-b-right).

This algorithm proves to work extremely well on embedded clouds generated through LLE. It becomes then possible to detect transitions in the topology of the moving body when they happen, and modify the number of clusters accordingly. We will return on this in Section 3.4.

3.3 Temporal Consistency and Seed Propagation

When considering entire sequences of 3D clouds we need to ensure the *temporal consistency* of the segmentation: in normal situations (no topology changes due to contact of different body-parts) the cloud has to be decomposed into the "same" groups in all instants of the sequence. We propose a propagation scheme in which centroid clusters at time t are used to generate initial seeds for clustering at time $t+1$ (Figure 4). Let n be the number of clusters.

Seed Propagation Algorithm

1. The embedded cloud $Y(t)$ at time t is clustered using d-wise clustering (Section 3.1, Figure 4-bottom-left) using the current seeds $c'_j(t)$ (the branch terminations of $Y(0)$ if $t = 0$, computed as in step 4 for $t > 0$);
2. For each centroid $c_j(t)$ $j = 1, ..., n$, of these clusters, the original datapoint $x_{i_j}(t)$ (*3D cluster centroid*) whose embedding $y_{i_j}(t)$ is the closest neighbor of $c_j(t)$ is found (Figure 4-top-left):

$$i_j(t) = \arg \min_{i=1,...,N} \|y_i(t) - c_j(t)\|^2. \quad (1)$$

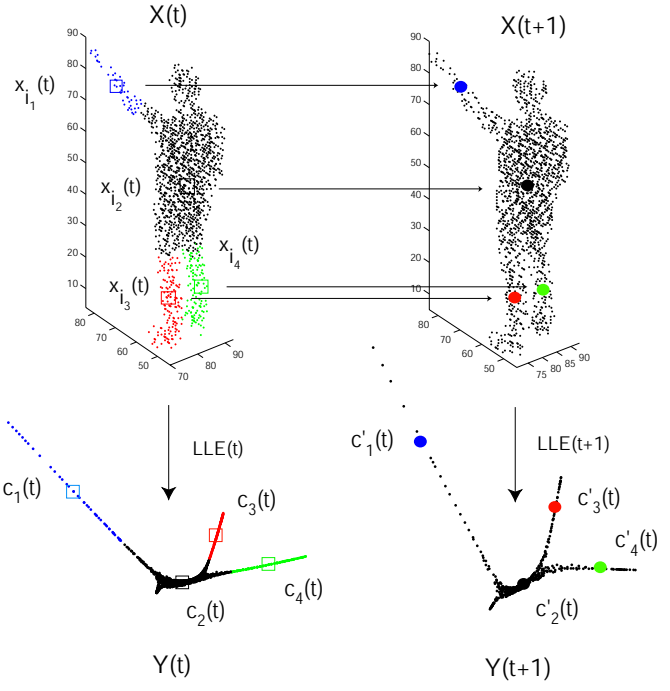

Fig. 4. Seed propagation for consistent clustering along time in the embedding space. The anti-images of centroids at time t are added to the 3D cloud at time $t+1$. Their embeddings $c'_j(t+1)$ are the seeds from which to start clustering the new embedded cloud $Y(t+1)$.

3. At time $t+1$, the dataset of 3D input points $X(t+1) = \{x_i(t+1), i = 1, ..., N(t+1)\}$ at time $t+1$ is augmented with the positions of the old 3D centroids at time t, yielding a new dataset (Figure 4-top-right)

$$X'(t+1) = X(t+1) \cup \{x_{i_j}(t), j = 1, ..., n\}. \tag{2}$$

4. LLE is applied to the extended dataset $X'(t+1)$, obtaining (Figure 4-bottom-right) [1]

$$Y(t+1) \cup \{c'_j(t+1), j = 1, ..., n\}. \tag{3}$$

5. The images $c'_j(t+1)$ of the *old* 3D centroids $x_{i_j}(t)$ in the *new* embedded space will then be used as seeds to start the k-wise clustering of the new embedded cloud $Y(t+1)$.

3.4 Topology Changes and Dynamic Clustering

The question of how to initialize the seeds for $t = 0$ naturally arises. Besides (even though working in an embedding space helps to dramatically reduce the

[1] Embeddings $c'_j(t+1)$ of the old 3D centroids $x_{i_j}(t)$ in the new embedded cloud can also be computed by *out of sample extension* [26].

problem of segmenting body-parts which get close to each other) moments in which different parts of the articulated body come to contact still have important effects on the shape of the embedded cloud. In fact, in an unsupervised context in which we do not possess prior knowledge about the number of rigid parts which form the body or the way they are arranged, there is no reason to distinguish adjacent body-parts. It is instead more sensible to adapt the number of clusters to the number of actually distinguishable parts.

The branch detection algorithm of Section 3.2 provides a tool to initialize the clustering machinery and implement the necessary change in the number and location of clusters when a topology change occurs.

Clusters' Merging/ Splitting Algorithm

1. At each time instant t all branch terminations of the embedded cloud $Y(t)$ are detected; if $t = 0$ they are used as seeds for k-wise clustering.
2. Otherwise ($t > 0$) standard k-means is performed on $Y(t)$ using branch terminations as seeds, yielding a rough partition of the embedded cloud into distinct branches.
3. Propagated seeds $c'_j(t)$ in the same partition are merged.
4. For each partition of $Y(t)$ not containing any old seed a new seed is defined as the related branch termination.

The third situation takes place when previously separated body-parts get too close to be distinguished: it makes then sense to merge the corresponding clusters. 4. embodies the opposite event in which a body-part which was previously impossible to distinguish becomes well separated, requiring then the introduction of a new cluster. This way clusters merge and/or split according to topological changes in the moving articulated body.

4 Algorithm

It is time to summarize our approach for unsupervised robust segmentation of parts of moving articulated bodies in a consistent way along a sequence (by assembling the separate algorithms we described in Sections 3.1, 3.3, 3.4). For each time instant t:

1. The current dataset ($X(t) = \{x_i(t), i = 1, ..., N(t)\}$ for $t = 0$, $X'(t) = X(t) \cup \{x_{i_j}(t-1)\}$ for $t > 0$, Figure 5-a) is mapped to an embedding space of dimension d yielding $Y(t) = \{y_i(t), i = 1, ..., N(t)\} = LLE(X(t))$.
2. All branch terminations of the embedded cloud $Y(t)$ are detected (Section 3.2): the natural number of clusters $n(t)$ for time t is then set to the number of branches (plus one for the torso), Figure 5-b.
3. The embedded cloud $Y(t)$ is clustered into $n(t)$ groups by d-wise clustering (Section 3.1) starting from $n(t)$ seeds (Figure 5-c):

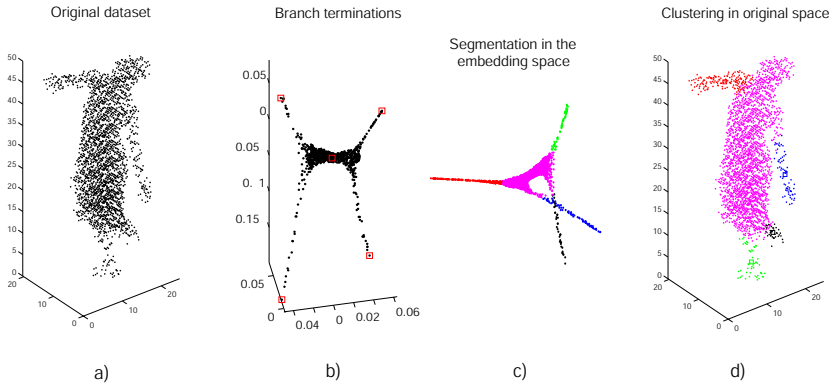

Fig. 5. Graphical illustration of the segmentation algorithm

- if $t = 0$, we use all branch terminations as seeds;
- if $t > 0$, the seeds are derived from the old centroids $\{c'_j(t), j = 1, ..., n(t-1)\}$ after the splitting/merging procedure exposed in Section 3.4.

4. This yields a new set of centroids $\{c_j(t), j = 1, .., n(t)\}$.
5. The labeling of the embedded points induces a segmentation in the original 3D shape (Figure 5-d).
6. All cluster centroids $c_j(t)$ are remapped to 3D (3.3), the corresponding 3D centroids $x_{i_j}(t)$ are added to the new dataset $X(t+1)$ at time $t+1$ (Figure 4).

5 Experiments

To test our spectral approach to body-part segmentation, we analyzed its performance on a large number of sequences acquired through our acquisition system composed by 8 synchronized cameras. Silhouettes were processed in order to compute first their visual hull, and the moving 3D articulated body was finally rendered as a uniformly sampled voxelset (Figure 6). We applied the algorithm to several different high-resolution sequences, one of which a 200-frame-long sequence capturing a dancer who moves and swirls all around the scene.

Figure 6 illustrates some typical results of the dynamic segmentation algorithm of Section 4, for a number of sequences. It can be appreciated that segmentation along time turns out to be pretty consistent, yielding very smooth cluster trajectories, non only in situations where body-parts are well separated (Figure 6-a,-b) but also during walking gaits (c) or even extremely complicated motions like the dance performed in (Figure 6-d,-e). In particular, in the "danceuse" case several topology transitions take place, due to the presence of a moving scarf around the waist of the woman, and numerous contacts between legs and arms during the dancer's performance. The algorithm segments the sequence in subsequences with constant topology (e.g. 6-d,-e), within which clusters are smoothly tracked. Another example of such a temporal segmentation in a different sequence is illustrated in 6-f).

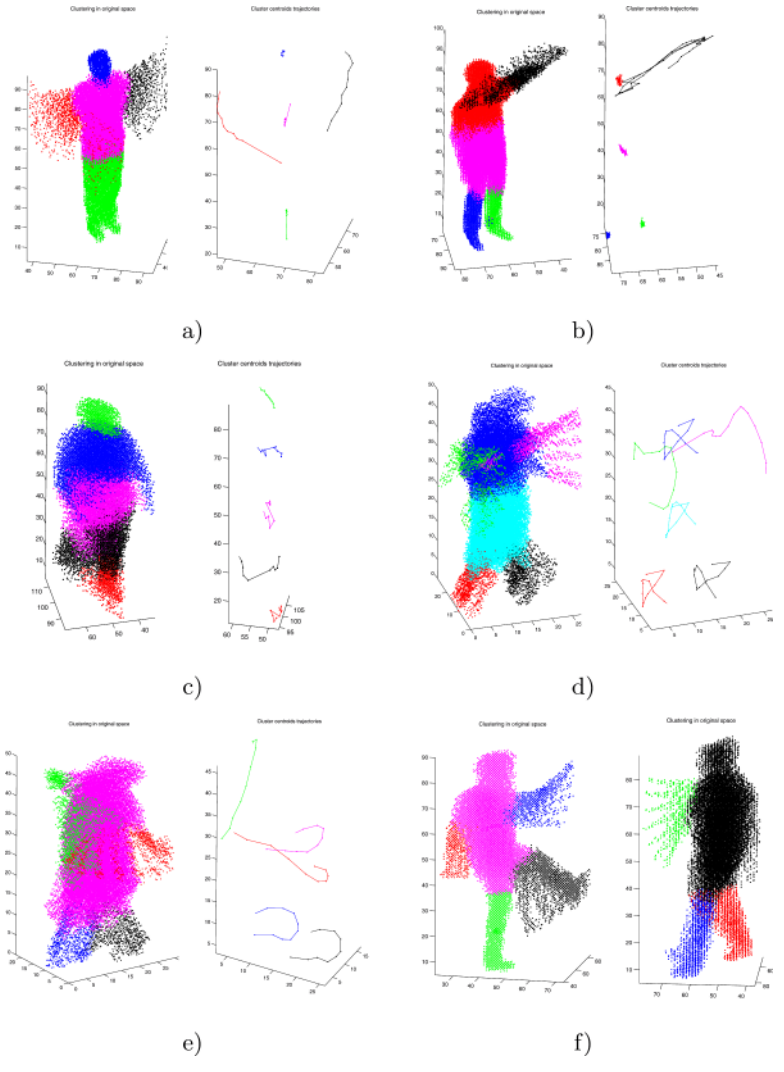

Fig. 6. Examples of the results produced by the dynamic segmentation algorithm of Section 4 on a number of sequences. a) a "fly" sequence of 11 frames. b) "arm-waving" sequence of length 50. c) "walking" motion, length 10. d) a subsequence of "danceuse", 16 frames. e) another "danceuse", length 10. Both whole clusters' evolution and their centroids' trajectories are shown. f) Topology transition management: after 11 frames in which arms are spread out (left) the left arm gets in contact with the torso: the algorithm adapts the number of clusters accordingly and proceeds to segment in a smooth way for other 7 frames (right, from a different viewpoint).

5.1 A Comparison with Direct EM Clustering in 3D

In order to show the advantages of our methodology over direct clustering in 3D we compared these results with those of a scheme similar to that of Section 4 in which old seeds and weights are passed to the next frame in order to make the segmentation coherent along time, but k-means or k-wise clustering in the embedding space are replaced by straightforward EM clustering of the original 3D shape. The idea is to model the probability density of the data as a convex combination of (typically Gaussian) components (which can be seen as clusters) $f(y) = \sum_j w(j) f_j(y)$, $\sum_j w(j) = 1$, $f_j(y) \sim N(\mu_j, \Sigma_j)$ whose parameters are estimated through the EM algorithm [27]. Figure 7 shows the resulting segmentation

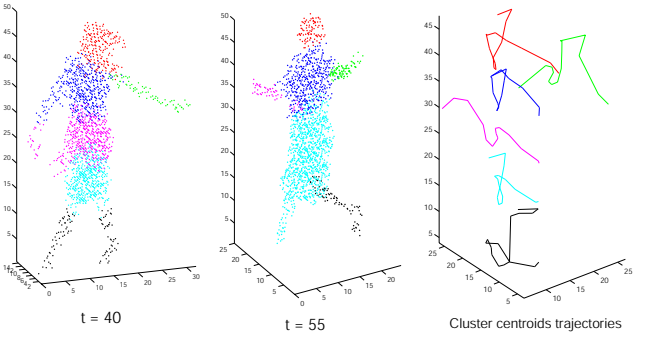

Fig. 7. Behavior of dynamic segmentation based on EM clustering. Even in presence of seed propagation, clusterings in different instants of a sequence (here $t = 40$ and $t = 55$ of "danceuse") are not consistent. This is confirmed by apparent irregularities in all centroids' trajectories (right).

on the same sequence of Figure 6-d, for some sample frames $t = 40, 45, 50, 55$, as an example of its typical performance. Besides spanning different body-parts at the same time (since clustering is based on the original Euclidean distance), clusters evolve inconsistently along time. A comparison of the related cluster trajectories with Figure 6-d highlights this irregular behavior.

5.2 Dimension of the Embedding Space

The quality of the segmentation depends on the stability of the embedded shape, which is in turn affected by the parameters of the embedding, like the dimension of the embedding space (i.e. the number of eigenvectors we selected after SVD of the affinity matrix M [19]). A better segmentation performance can be in general observed when we set as dimension of the embedding space a number similar to the expected number of clusters. Figure 8 shows, for a given frame ($t = 59$ of the "dancer" sequence) and an equal number of neighbors $k = 20$, the different segmentations we obtain for $d = 3$ (top) and $d = 4$ (bottom). Even though the scarf is clearly visible in the 3D cloud, a three-dimensional embedding space

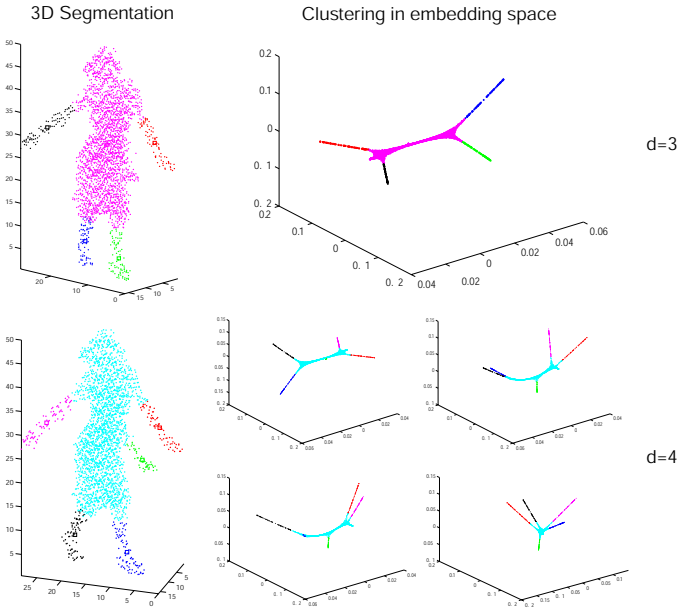

Fig. 8. Different segmentations of the same pose obtained for different dimensions of the embedding space. Top: for $d = 3$ the scarf merges with the torso into the same branch of the embedded cloud (top-right). Bottom: for $d = 4$ a separate branch associated with the scarf (in green) is present in the embedding shape.

fails to reveal its presence as separated branch of the embedded cloud (Figure 8-top-right). Such a new (green) branch appears instead in the four-dimensional embedding space: Figure 8-bottom-right displays the 4 orthogonal projections of the embedded cloud for $d = 4$ (with axes, x, y, z, w) onto its three-dimensional subspaces (x, y, z), (x, y, w), (x, z, w), (y, z, w).

5.3 Estimating the Optimal Number of Neighbors

The most critical parameter for the entire segmentation algorithm is however the number of neighbors k of the LLE step, as it affects the stability of the embedded shape from which the estimation of the number of clusters depends. It can be empirically noticed that, while the embedded shape shows a remarkable stability for some values of k this is not in general the case for arbitrary such values. It is then desirable to *estimate a variable number of neighbors in time*, in order to guarantee the stability of $Y(t)$ and ensure a consistent segmentation along time (Figure 9). For too large values of k some neighborhoods of points in a given body-part can comprise regions of a different body-part (middle). These "anomalous" neighborhoods are characterized by the fact that their farthest elements (as they belong to another, distinct link) are relatively distant from all others elements (which instead lie all on the same rigid part). If we then plot

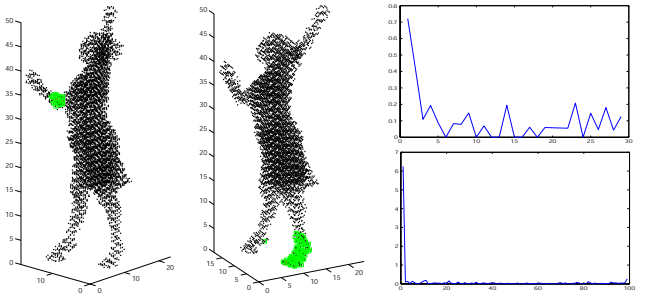

Fig. 9. How to estimate the correct number of neighbors k in the LLE algorithm. Non-admissible values of k are characterized by "anomalous" neighborhoods which span distinct body-parts (middle), in opposition to admissible values (left). The corresponding distance plots are visible to the right.

the distance between the farthest point of the neighborhood and all its fellows we notice a large jump (right-bottom).

This is not the case for neighborhoods which span a single rigid part (left). It is natural to choose as correct k any of those values which yield only "regular" neighborhoods (for instance the average of the interval of admissible values).

5.4 Robustness to Topology Changes

In any case, moments in which different parts of the articulated body come to contact still have important effects on the shape of $\{Y_i\}$. Figure 10 illustrates how the dynamic clustering technique copes with such changes. These events have dramatic consequences on embeddings based on measuring geodesic distances along the body, since new paths appear affecting in general the distance between

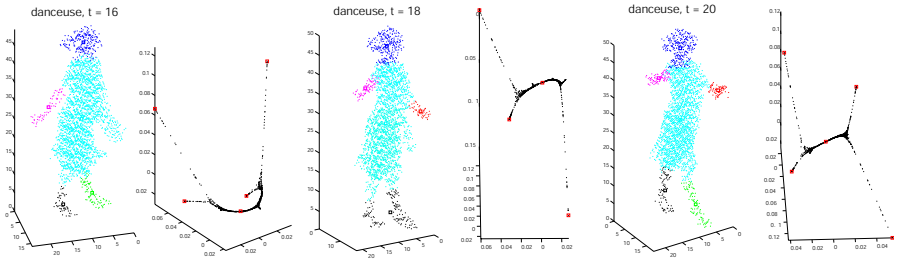

Fig. 10. How the splitting/merging algorithm deals with topology changes in the embedded space. At $t = 16$ the left arm of the dancer touches her scarf (left), and a single cluster covers body, arm, and scarf. Then ($t = 18$) the left arm becomes visible and a new cluster is assigned to it, while the dancer's feet get too close to be distinguished (middle). Finally ($t = 20$) her legs widen again, inducing a separate cluster for each one of them (right).

all pairs of points in the original cloud. Figure 11 shows how ISOMAP (as a representative of all geodesic-based spectral methods) copes with the transitions of Figure 10. Clustering is performed in the ISOMAP space by k-means.

Besides not having the desirable behavior in terms of branch separation of LLE or Laplacian Eigenmaps, geodesic spectral methods prove incapable of handling those situations, which are extremely common in any natural articulated motion. Even though we left the algorithm free to estimate the "best" number of clusters along the sequence, EM proved also incapable of adapting itself to the mutated topology of the body. As a result, clusters would "shift" around the body in a rather unstable way, failing to segment in a consistent way moving body-parts.

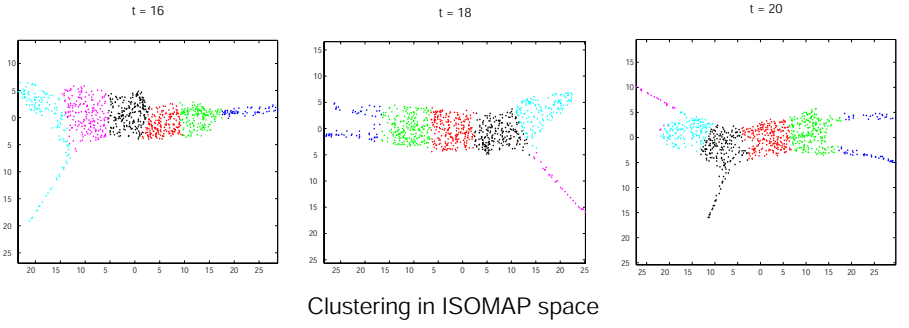

Fig. 11. Behavior of ISOMAP along the sequence of topology transitions of Figure 10. The shape of the embedded cloud (we chose $d = 2$ for sake of readability) changes dramatically, gravely affecting the segmentation in the original 3D space.

6 Conclusions

In this paper we presented a novel dynamic segmentation scheme in which moving articulated bodies are clustered in an embedding space, and clusters propagated in time to ensure temporal consistency. Exploiting some desirable features of LLE, in particular, we proposed a systematic way of estimating the optimal number of clusters in order to merge/split clusters in correspondence of topology transitions. To ensure stability and improve segmentation performance we proposed a method to learn the critical parameter k of the embedding algorithm. We compared the performance of the algorithm with similar propagation schemes based on direct EM clustering in 3D, and k-means clustering in ISOMAP space.

It is natural to imagine this unsupervised segmentation procedure as a building block of more detailed motion analysis, in which kinematic or stick models are fitted to the data based on the obtained segmentation. Even though the reconstructed segments are not in general associated with "natural" body-parts (as we assume no model of the moving body is available) temporal consistency as assured by our propagation scheme guarantees the coherence of the rough stick model which corresponds to those segments. The latter can be seen as a first guess of the underlying kinematic model, which could be later improved by

exploting 3D point matching schemes based on shape alignment in the embedding space [28]. Once obtained trajectories for each point of the cloud we could indeed easily cluster them according to the similarity of the associated motions, distinguish this way distinct rigid links belonging to the same initial segment, eventually achieving a finer bottom-up model of the body.

In a different context, as they are inherently invariant with respect to the direction of the motion, cluster centroids may provide good features to use for action recognition, for instance by feeding them to a classical hidden Markov model. We plan to explore both those opportunities in the near future.

References

1. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision based human motion capture and analysis. Computer Vision and Image Understanding 103(2-3), 90–126 (2006)
2. Hogg, D.: Model based vision: a program to see a walking person. Image and Vision Computing 1(1), 5–20 (1983)
3. Gavrila, D., Davis, L.: 3-D model-based tracking of humans in action: A multi-view approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, USA, pp. 73–80. IEEE Computer Society Press, Los Alamitos (1996)
4. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, USA, vol. 2, pp. 126–133 (2000)
5. Brand, M.: Shadow puppetry. In: Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, vol. 2, pp. 1237–1244 (1999)
6. Grauman, K., Shakhnarovich, G., Darrell, T.: Inferring 3D structure with a statistical image-based shape model. In: Proceedings of the 9th International Conference on Computer Vision, Nice, France, pp. 641–648 (2003)
7. Elgammal, A., Lee, C.: Inferring 3D body pose from silhouettes using activity manifold learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Washington, USA, pp. 681–688. IEEE Computer Society Press, Los Alamitos (2004)
8. Peursum, P., Venkatesh, S., West, G.: Tracking-as-recognition for articulated full-body human motion analysis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA, pp. 1–8. IEEE Computer Society Press, Los Alamitos (2007)
9. Cheung, G., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3D voxel reconstruction of human motions. In: Proceedings of CVPR 2000, pp. 2714–2720 (2000)
10. Mukasa, T., Nobuhara, S., Maki, A., Matsuyama, T.: Finding articulated body in time-series volume data. In: Perales, F.J., Fisher, R.B. (eds.) AMDO 2006. LNCS, vol. 4069, pp. 395–404. Springer, Heidelberg (2006)
11. de Aguiar, E., Theobalt, C., Magnor, M., Theisel, H., Seidel, H.P.: M3: Marker-free model reconstruction and motion tracking from 3D voxel data. In: Cohen-Or, D., Ko, H.S., Terzopoulos, D., Warren, J. (eds.) PG 2004. 12th Pacific Conference on Computer Graphics and Applications, Seoul, Korea, pp. 101–110. IEEE Computer Society Press, Los Alamitos (2004)

12. Brostow, G.J., Essa, I., Steedly, D., Kwatra, V.: Novel skeletal representation for articulated creatures. In: Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, vol. 3, pp. 66–78 (2004)
13. Chu, C.W., Jenkins, O.C., Mataric, M.J.: Markerless kinematic model and motion capture from volume sequences. In: CVPR 2003. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 475–482. IEEE Computer Society Press, Los Alamitos (2003)
14. Sundaresan, A., Chellappa, R.: Segmentation and probalistic registration of articulated body models. In: Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, vol. 2, pp. 92–96 (2006)
15. Jenkins, O., Mataric, M.: A spatio-temporal extension to isomap nonlinear dimension reduction. In: Proceedings of the 31th International Conference on Machine Learning, Alberta, Canada (2004)
16. Lin, R., Liu, C.B., Yang, M.H., Ahuja, N., Levinson, S.: Learning nonlinear manifolds from time series. In: Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, vol. 2, pp. 245–256 (2006)
17. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems 14, MIT Press, Cambridge (2002)
18. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
19. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
20. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
21. Heiser, W.J., Bennani, M.: Triadic distance models: Axiomatization and least squares representation. J. Math. Psy. 41, 189–206 (1997)
22. Hayashi, C.: Two dimensional quantification based on the measure of dissimilarity among three elements. Ann. I. Stat. Math. 24, 251–257 (1972)
23. Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., Belongie, S.: Beyond pairwise clustering. In: Proceedings of CVPR, vol. 2, pp. 838–845 (2005)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. PAMI 22(8), 888–905 (2000)
25. Ng, M.J.A., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14: Proceedings of the 2001 (2001)
26. Bengio, Y., Paiement, J.F., Vincent, P.: Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. Technical report, Universite' de Montreal (2003)
27. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. 39, 1–38 (1977)
28. Mateus, D., Cuzzolin, F., Boyer, E., Horaud, R.: Articulated shape matching by locally linear embedding and orthogonal alignment. In: Proceedings of the ICCV'07-NTRL Workshop, Rio de Janeiro, Brasil (2007)