# Coherent Laplacian 3-D protrusion segmentation

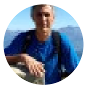
**CONFERENCE PAPER** *in* PROCEEDINGS / CVPR, IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION · JULY 2008

DOI: 10.1109/CVPR.2008.4587452 · Source: DBLP

**5 AUTHORS**, INCLUDING:

Fabio Cuzzolin
Oxford Brookes University
**91** PUBLICATIONS **518** CITATIONS

Edmond Boyer
National Institute for Research in Computer …
**119** PUBLICATIONS **2,787** CITATIONS



Radu Horaud
National Institute for Research in Computer …
**217** PUBLICATIONS **3,811** CITATIONS

# Coherent Laplacian *3-D* protrusion segmentation

Fabio Cuzzolin* Diana Mateus† David Knossow Edmond Boyer Radu Horaud
INRIA Rhone-Alpes
655 Avenue de l'Europe, Montbonnot, France
firstname.lastname@inrialpes.fr

## Abstract

*In this paper, an analysis of locally linear embedding (LLE) in the context of clustering is developed. As LLE conserves the local affine coordinates of points, shape protrusions as high-curvature regions of the surface are preserved. Also, LLE's covariance constraint acts as a force stretching those protrusions and making them wider separated and lower dimensional. A novel scheme for unsupervised body-part segmentation along time sequences is thus proposed in which* 3-D *shapes are clustered after embedding. Clusters are propagated in time, and merged or split in an unsupervised fashion to accommodate changes of the body topology. Comparisons on synthetic, and real data with ground truth, are run with direct segmentation in* 3-D *by EM clustering and ISOMAP-based clustering. Robustness and the effects of topology transitions are discussed.*

## 1. Introduction

Human motion analysis is a crucial topic in computer vision with applications in surveillance, human machine interface and animation, amongst others, and has received considerable attention from the community over the last decades [17]. *Model based* approaches, e.g. [12, 10], assume a known, often kinematic, model for the human body and recover parameters of this model in a joint space using image information: such a search is however difficult without adequate initialization. *Learning based* approaches [3, 11, 9] which directly relate visual information to learned body configurations are not affected by the initialization issue, but are limited by the use of training sets of examples. In contrast, techniques have been proposed that directly infer body poses from multiple image cues or volume sequences: *skeletonization* methods for instance recover the intrinsic articulated structure of *3-D* shapes, either directly in *3-D* [5], or in an embedded space [6, 22]. Spectral embeddings [2, 23] have indeed the ability to map *3-D* shapes onto low-dimensional manifolds, thus naturally revealing the intrinsic structure of an articulated shape. A critical issue is the presence of topological ambiguities raised by self contacts, as noticed by Sundaresan and Chellappa [22] who used an *a priori* graphical model to resolve them.

In this work, we propose a spectral approach that segments *3-D* body shapes along time without any *a priori* information nor learned examples, while seeking robustness to topological ambiguities over time. Recent attempts to extend nonlinear reduction to spatio-temporal data [14, 16] rely on enforcing temporal relationships when embedding time sequences, a hard task when handling dense shape representations. We propose instead a mechanism to enforce temporal consistency of segments obtained by collinear clustering in the embedded space, where clusters are remarkably stable under articulated motions, and propagate them over time. We favor *Local Linear Embedding* [20] (LLE) as it exhibits a number of desirable features in the specific scenario of unsupervised segmentation: it conserves shape protrusions in virtue of its local isometry, while their separation is increased and their intrinsic dimensionality reduced as an effect of the covariance constraint.

The paper is organized as follows: we first explain, in Sec. 2, how LLE's desirable features emerge from the related optimization problem. The proposal algorithm is illustrated in Sec. 3: the use of *k-wise* clustering to segment the embedded cloud, starting from detected branch terminations; how to propagate clusters over time to ensure consistency and merge/split them to fit the topology of the body. An extensive experimental Sec. 4 assesses, on both synthetic and challenging real data, the performance of our algorithm and other competing methods by comparison with ground truth labeling, tests the way they cope with topology transitions, and studies the robustness of the algorithm.

## 2. An analysis of Locally Linear Embedding

Spectral methods share a common framework in which an affinity matrix $M$, which reflects the structure of the input data-set, is eigendecomposed, and its eigenvalues and/or eigenvectors are later used for purpose-specific applica-

*F.C. is a post-doctoral, supported by a grant from the European Community under the Marie-Curie Research Training Network Visiontrain.

†D.M. is a Ph.D, supported by a grant from the European Community under the Marie-Curie Early Stage Training, Visitor.

tions. In particular, the *graph Laplacian* is an operator on functions $f$ defined on a set of points $X = \{X_i, i = 1, ..., N\}$ of the form $(Lf)_i \propto \sum_{j \in N(i)} w_{ij}(f_i - f_j)$, with $N(i)$ the set of neighbors of the point $X_i$, $w_{ij}$ a scalar (weight), and $f_i$ the value of $f$ on $X_i$. The eigenvalues of $L$ are invariant with respect to volume-preserving transformations [24]. By analogy with Fourier basis functions, Laplacian eigenfunctions are stationary functions on $X$ and form a natural "base" for functions on the grid. Their stationarity implies that their zero-level sets or *nodal sets* are very much related to protrusions and symmetries of the underlying grid of points [15] (Fig. 1). *Locally Linear Embedding* (LLE) [20] can also be thought of as a graph Laplacian based embedding[1], inheriting their good geometric properties. Some specificities, though, make it particularly suitable for the problem of unsupervised clustering of protrusions. LLE computes $d$-dimensional embeddings $Y$ of sets of input points $X = \{X_i, i = 1, .., N\}$ while preserving their local structure. For each data point $X_i$ we compute the weights $W_{ij}$ that best linearly reconstruct $X_i$ from its neighbors, by solving the following constrained least-square problem: $\arg\min_{\{W_{ij}\}} \sum_i |X_i - \sum_j W_{ij} X_j|^2$. The low-dimensional embedded vectors $Y_i$ for each $X_i$ are obtained by solving $\arg\min_{\{Y_i\}} \sum_i |Y_i - \sum_j W_{ij} Y_j|^2$; the embedded cloud is required to be centered at the origin $\sum_i Y_i = \mathbf{0}$, and constrained to have unit covariance:

$$\frac{1}{N} \sum_{i=1}^{N} Y_i \otimes Y_i = I. \qquad (1)$$

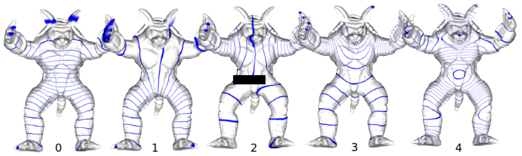


Figure 1. Laplacian eigenfunctions are associated with symmetries and protrusions of their domain. The symmetry axis of the toy corresponds to a nodal set of eigenfunction 2; cross sections of the arms are level sets of eigenfunctions 3 and 4. Courtesy [15].

The objective function to minimize can be expressed as a quadratic form $\sum_{ij} M_{ij} Y_i \cdot Y_j$ involving the matrix $M \doteq (I - W)^T (I - W)$. The optimal embedding (up to a global rotation) is found by computing the bottom $d + 1$ eigenvectors of $M$, and discarding the bottom (unitary) one.

We explore now, three of the interesting geometric properties of the LLE.

**Number of protrusions and local isometry.** Gauss' "Theorema Egregium" states that the curvature of a surface can be determined entirely by measuring angles and distances on the surface. The "Gaussian curvature" or product $\kappa = \kappa_1 \kappa_2$ of the principal curvatures $\kappa_1, \kappa_2$ (the minimum and maximum curvature of all curves passing through the point) is intrinsic, i.e. is invariant under local isometry. By definition, LLE leaves unchanged for each neighborhood *the weights* that best linearly reconstruct each data-point from its neighbors. As $\sum_j W_{ij} = 1$ [20], those weights are nothing but the affine coordinates of $X_j$ on the base formed by its neighbors $X_j$, and can be expressed in terms of their respective distances as $W_{ij} = \frac{\sum_{m \neq j} |X_i - X_m|}{\sum_m |X_i - X_m|}$. It follows that preserving the weights implies preserving the distances *up to a scale* which depends on the size of the neighborhood, measured as $\sum_m |X_i - X_m|$. In particular, such weights are preserved in the intersection of all neighborhoods with the external surface of the *3-D* shape. Distances are then preserved up to a scale *over the entire surface* of the shape: but then, local curvature is also preserved up to a scale inside all "surface" neighborhoods. Now, if the "size" $\sum_m |X_i - X_m|$ of each neighborhood in the original cloud $X$ is roughly the same (which is the case when the density of the points $X_i$ is homogeneous inside the shape) the distribution of the curvature along the surface is preserved by the embedding. As protrusions in the original *3-D* shape are obviously associated with high curvature regions of the delimiting surface (in a human body, think of feet or hands) they are also conserved after LLE embedding (also Hessian eigenmaps, [8]).

**A force related to the covariance constraint.** If the conservation of the number of protrusions is an effect of the objective function minimized by LLE (i.e. local isometry), other desirable features that help detecting and clustering shape protrusions are consequences of the covariance constraint (eq. 1). LLE is a constrained minimization problem $\min_{\vec{x} = [x_1, \ldots, x_n]} F(x_1, \ldots, x_n)$ subject to $G_1(\vec{x}) = 0$, $\ldots$ , $G_m(\vec{x}) = 0$. A typical example from physics is a point mass in a potential field $V(\vec{x})$ constrained to move along a surface $G(\vec{x}) = 0$. The problem reduces to minimize the action $S = \int L(\vec{x}, \dot{\vec{x}}) - \lambda(t) G(\vec{x}(t)) dt$ where $L$ is the Lagrangian, and $\lambda(t)$ is the Lagrangian multiplier. The term $-\lambda(t) G(\vec{x}(t))$ has the shape of a potential energy, that changes by $\delta V = \delta \vec{x} \frac{\partial(\lambda G)}{\partial \vec{x}}$ whenever the point leaves the surface by a vector $\delta \vec{x}$. This is equivalent to say that there exists *a force*

$$\vec{F} = -\lambda \frac{\partial G}{\partial \vec{x}} \qquad (2)$$

directed *orthogonally to the constraint surface*. In the case of LLE the force (eq. 2) associated with the covariance constraint pulls outwards and stretches the chain of links formed by all local neighborhoods, redistributing them on a (roughly) lower dimensional manifold (see Fig. 2-a). As this force pulls those chains in *radial* direction their separation increases in the embedding space.

**Clustering along time and pose-invariance.** As our purpose is to cluster entire sequences, though, we need the segmentation obtained at different time instants to be *con-*

[1] As the affinity matrix $M$ (as an operator) can be approximated by the square Laplacian $Mf = (I - W)^T (I - W) f \approx \frac{1}{2} \mathcal{L}^2 f$, LLE is indeed a form of Laplacian embedding.

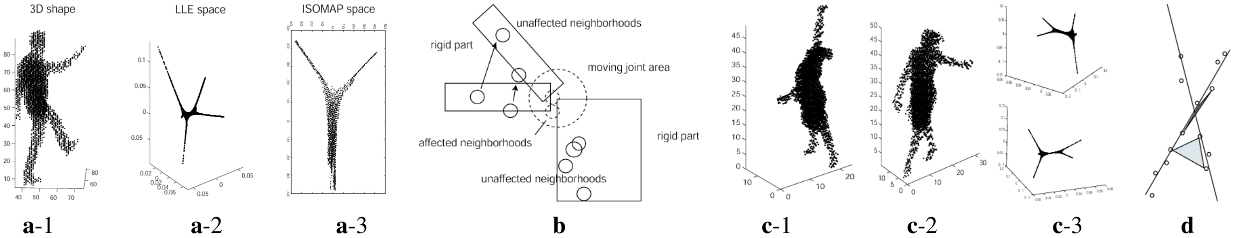


Figure 2. **a**) The same *3-D* point cloud (a-1) is maped with LLE (a-2) and ISOMAP [23] (a-3). LLE increases the separation of protrusions like legs and arms, unlike methods based on geodesic distances [6]. **b**) The number of neighborhoods affected by articulated motion is relatively small. **c**) Stability of LLE under articulated motion. Different poses (c-1, c-2) of the same articulated body are mapped to the same embedding (c-3). **d**) k-wise clustering (with $k = 3$). Areas of triangles defined by triads of points measure their collinearity.

*sistent*. Indeed 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 and other graph Laplacian embeddings [2]. However, we assume an articulated body is formed by rigid parts linked by a small number of joints to a core. Thus, all local neighborhoods incident on a rigid part are preserved along the motion, while only the few neighborhoods containing evolving joint(s) are affected (Fig. 2-b). Fig. 2-c pictures two different poses of a dancer performing a ballet, and shows how the related embedded clouds obtained through LLE for $d = 3$ and $K = 10$ are indeed stable.

## 3. Unsupervised spectral segmentation

Shape protrusions are then conserved in the LLE spectral domain while their separation is increased and their intrinsic dimensionality reduced. The effect is dramatic in situations in which body-parts are physically close to each other in *3-D*, and makes clustering much easier after embedding.

We propose an *unsupervised*, time-consistent, method for protrusion segmentation of *3-D* shapes where *no kind of model* (not even a weak topological, or graphical one, as in [22]) is assumed, and the number of clusters themselves is inherently determined by the lower dimensional structure of the embedded cloud. As a consequence, it makes no sense to use generic k-means to segment protrusions (in opposition to generic spectral clustering [18]) and we adopt *k-wise clustering* [1]. Given a sequence of *3-D* shapes, this segmentation is *consistently propagated* along time, and the number of clusters estimated in an automatic way to fit the topology. In the remaining of the section, we will introduce the different stages of the proposed algorithm.

**k-wise clustering in the embedded shape.** We seek clusters formed by sets of roughly collinear embedded points. This can be done by measuring the affinity of *triplets* of points (Fig. 2-d), for instance the area of the triangle they form (or the volume of the $(d-1)$-dimensional hyper-edge in the general case [1]). Given a set of data-points $V = \{v_i, i = 1, ..., N\}$:

**1.** In the first step an *affinity hyper-graph* is built. A weighted undirected hyper-graph $H$ is a pair $(V, h)$, where $V$ is the set of vertices of $H$, and subsets $z$ of $V$ of size $k$ are called hyper-edges. The function $h$ associates non-negative weights $h(z)$ with each hyper-edge ($k$-tuple) $z = \{v_{j_1}, ..., v_{j_k}\}$, and measures the affinity of each hyper-edge.

**2.** Then, a weighted graph $G = (V, g)$ that approximates the hyper-graph $H$ is constructed by constrained least squares optimization, assuming that the weight of each hyper-edge is the arithmetic mean of the weights of the edges of $G$ incident on it (*clique averaging*).

**3.** The approximating graph $G$ is partitioned by n-cut [21].

Here the set of vertices of the hyper-graph is the embedded cloud $V = Y$, where hyper-edges are formed by $d$ elements in case of a $d$-dimensional embedding space. The k-wise algorithm yields a segmentation in the embedding space that can be trivially re-mapped back to the original *3-D* space (Fig. 4), using the ordering of the data-points.

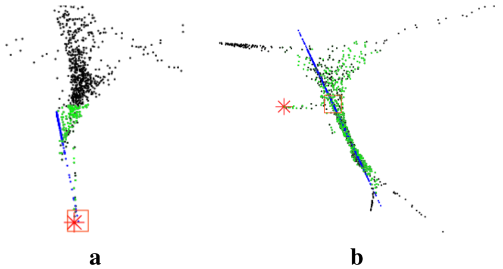


Figure 3. **a**) Branch termination detected. **b**) Branch termination not detected (internal point).

**Boundary detection and number of clusters.** The geometric qualities of LLE embeddings (Sec. 2) allow to estimate, at each time, instant the "correct" number of clusters (Fig. 3) by detecting protrusion boundaries [19]. Each point of the embedded cloud is tested to decide whether or not it is a branch termination by finding its nearest neighbors (within a threshold distance, plotted in green on Fig. 3), and fitting a line to them (in blue). An embedded point is a branch termination iff all its neighbors, projected on this line, lay on one side of the point's own projection (red square, Fig. 3-a). A branch termination is not detected when the projection has neighbors on both sides (Fig. 3-b).

**Temporal consistency and seed propagation.** For sequences of *3-D* clouds, we want to ensure the *temporal* *consistency* of the segmentation, factoring out topology changes the cloud has to be decomposed into the "same" segments at all times.

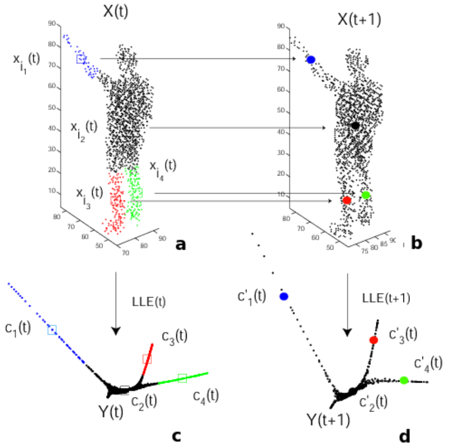


Figure 4. Seed propagation for time-consistent clustering in the embedding space. The anti-images (**a**) of centroids at time $t$ (**c**) are added to the *3-D* cloud at time $t+1$ (**b**). Their embeddings $c'_j(t+1)$ are the seeds from which to start clustering the new embedded cloud (**d**).

Centroid clusters at time $t$ can indeed be used to generate initial seeds for clustering at time $t+1$ (Fig. 4). Let $n$ be the number of clusters.

**1.** The embedded cloud $Y(t)$ at time $t$ is clustered using k-wise clustering, with $k = d$ (Fig. 4-c);

**2.** For each centroid $c_j(t)$ $j = 1, ..., n$, of these clusters the original data-point $x_{i_j}(t)$ (3-D *cluster centroid*) whose embedding $y_{i_j}(t)$ is the closest neighbor of $c_j(t)$ is found (Fig. 4-a);

**3.** The data-set $X(t+1) = \{x_i(t+1), i = 1, ..., N(t+1)\}$ at time $t+1$ is augmented with the positions of the old *3-D* centroids at time $t$, yielding a new data-set: $X'(t+1) = X(t+1) \cup \{x_{i_j}(t), j = 1, ..., n\}$ (Fig. 4-b);

**4.** LLE is applied to $X'(t+1)$, obtaining $Y(t+1) \cup \{c'_j(t+1), j = 1, ..., n\}$ (Fig. 4-d);

**5.** The images $c'_j(t+1)$ of the *old 3-D* centroids $x_{i_j}(t)$ in the *new* embedded space will then be used as seeds to start clustering the embedded cloud $Y(t+1)$ (Fig. 4-d).

**Topology changes and merging/splitting.** Laplacian methods like LLE are less sensitive to changes of the topology of the moving body than geodesic-based embeddings like ISOMAP (Fig. 2-a-3), as they compute only sets of weights inside local neighborhoods. However, those transitions still have important effects on the embedded cloud. In fact, in an unsupervised context where no prior knowledge on the body structure is available, there is no reason to tell apart adjacent body-parts: it is more sensible to fit the number of clusters to the mutated topology. The branch detection algorithm provides a tool to both initialize clustering and implement the necessary change in the number and the location of clusters when such a change occurs.

**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 (when previously separated body-parts get too close to be distinguished);

**4.** For each partition of $Y(t)$ not containing any old seed a new seed is defined as the related branch termination (when previously indistinguishable body-part becomes well separated).

**Algorithm.** Let us summarize our algorithm. At each instant $t$:

**1.** The current data-set $X(t) = \{x_i(t), i = 1, ..., N(t)\}$ for $t = 0$, $X'(t) = X(t) \cup \{x_{i_j}(t-1)\}$ for $t > 0$, is mapped to an embedding space of dimension $d$ yielding $Y(t) = \{y_i(t), i = 1, ..., N(t)\} = LLE(X(t))$ for $t = 0$, and $Y'(t) = \{y_i(t), i = 1, ..., N(t)\} \cup \{c'(t)\} = LLE(X'(t))$ for $t > 0$.

**2.** All branch terminations of $Y(t)$ are detected: the natural number of clusters $n(t)$ for time $t$ is then set to the number of branches (plus one for the core of the shape);

**3.** The embedded cloud $Y(t)$ is clustered into $n(t)$ groups by $d$-wise clustering starting from $n(t)$ seeds:

- if $t = 0$, we use all branch terminations as seeds;
- if $t > 0$, seeds are derived from old centroids by splitting/merging, $\{c'_j(t), j = 1, \ldots, n(t-1)\}$;

**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 *3-D* shape;

**6.** All cluster centroids $c_j(t)$ are re-mapped to *3-D*, the corresponding *3-D* centroids $x_{i_j}(t)$ are added to the new data-set $X(t+1)$ at time $t+1$ (Fig. 4).

## 4. Experiments

We tested our proposed methodology, on both synthetic and real data, in order to have both qualitative and quantitative assessments of its performance. In the case of synthetic sequences, we used as ground truth the labels automatically generated in a virtual setup (Fig. 6-b). We did the same for a number of real sequences for which motion capture data were available. Using this ground truth, we worked

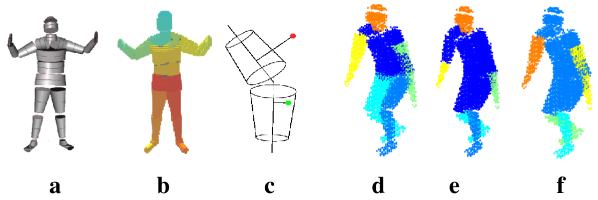


Figure 6. **a**) Synthetic model to generate labels. **b**) Generated labeled data. **c**) To label a voxel, we assign the label of the closest model body-part it lies in. **d**,**e**,**f**) Three different segmentations of the model to compare our segmentation result with.

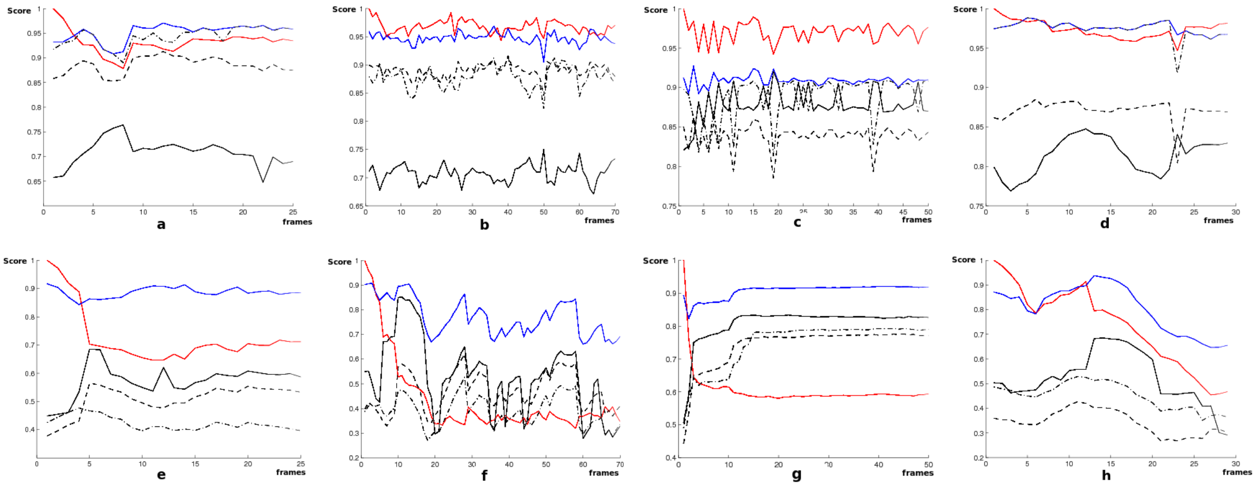


Figure 5. Top row: Segmentation scores obtained by comparing the labeling generated by our segmentation algorithm with ground truth provided for a number of synthetic sequences. Scores obtained over four different sequences of different length, from 25 to 70 frames, are shown. From left to right: “surf” (**a**,**e**), “Mars” (**b**,**f**), “laughter” (**c**,**g**), and “walk” (**d**, **h**). Red curves plot the consistency score, blue ones, the coarsening score. Solid, dashdot and dashed lines represent the scores respectively associated with the three *a priori* segmentations depicted on Fig. 6-d,e,f. Bottom row: segmentation scores obtained with dynamic EM clustering on the same synthetic sequences.

out three performance indicators:

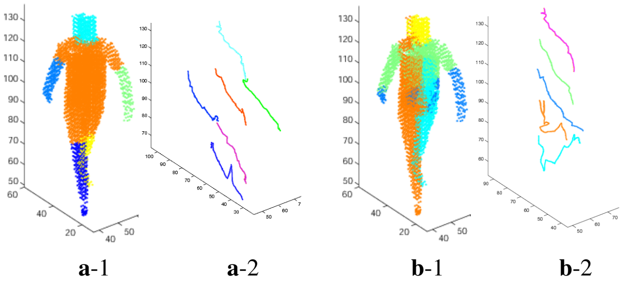
*Coarsening:* a segmentation is considered to be valid when it does not cut in half any of the rigid parts that compose the body. To measure this, associate to each segment of the ground-truth model (Fig. 6-b) the (unsupervised) cluster that best represents this segment (containing the largest set of points in common). Then, calculate the percentage of points out of this cluster, but still associated to the segment. If all clusters correspond to the *a priori* segments, this percentage is 0, and we give a score of 1.

*Segmentation:* compare the obtained segmentation with three different “natural” *a priori* subdivisions of the body. The score is similar to the coarsening one, but taking the labels from the segments in (Fig. 6-d,e,f).

*Time consistency:* take as reference the segmentation at time $t = 0$ for each body part and evaluate the drift along time. For each $t > 0$, and for each cluster, measure the similarity between the current label distribution and the initial one. The consistency score tends to one when the similarity measure is constant in time.

We compared our results with those of similar schemes in which seeds are also passed to the next frame to ensure time consistency, but clustering is performed in *3-D* on the original data-set using EM[2] or in the ISOMAP space [23] using k-means.

**Synthetic data.** We first run a set of experiments on several synthetic sequences generated by simulating the evolution of a human body model formed by a kinematic model plus the volumes representing its rigid links. Fig. 5 shows, as a flavor of the performances, the obtained segmentation scores for four different synthetic sequences using k-wise clustering (top row) and dynamic EM-clustering (bottom row). We can appreciate that the unsupervised segmentation turns out to be very consistent along time, as witnessed by the consistency score (red curves) between 95 and 100$\%$ at all times, even for fairly long sequences. In all cases, the boundary between clusters normally lies in correspondence to articulations (represented by the coarsening score, blue curves). The *a priori* segmentation scores (different black curves) seems to favor Fig. 6-e partition, i.e. it tends to detect the outermost rigid links instead of entire protrusions (e.g. forearms and tibias instead of arms and legs). This somehow unexpected result can be explained by pointing out that, in the synthetic model, joints have lower density and cause the embedded cloud to bend. Concerning the dynamic EM clustering (Fig. 5 bottom row), not only the absolute segmentation performance (black curves) is consistently and dramatically worse than that of the proposed algorithm, but also that the obtained segmentation is not at all consistent along time (red curves). Clusters drift inside the shape, along with the motion, and usually span different



**a**-1 **a**-2 **b**-1 **b**-2

Figure 7. Segmentation results and centroid trajectories for our algorithm (**a**-1,2) and EM clustering (**b**-1,2) (“walk”).

[2] The probability density of the data is modeled as a convex combination of 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 [7].

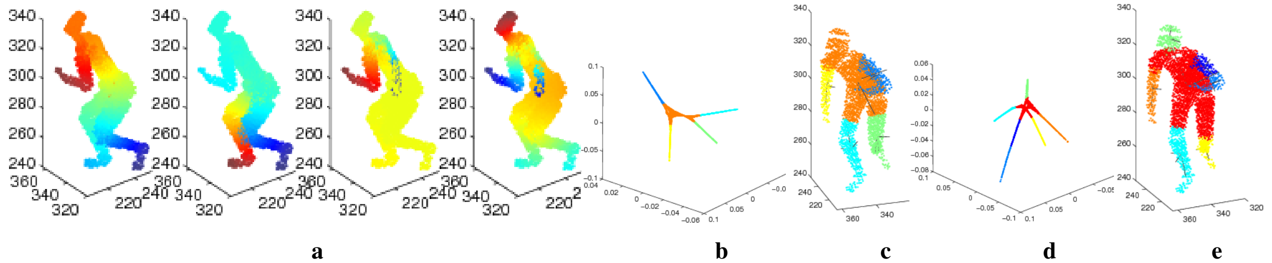


Figure 8. Different eigenfunctions capture different aspects of the shape geometry. **a**) The first four eigenfunctions for a pose. Each eigenfunction corresponds to a 1 by $N$ eigenvector of the affinity matrix: its value of the $i$-th point of the grid is the $i$-th entry of the eigenvector and is rendered here in color. **b**,**c**) Segmented embedding cloud and corresponding segmentation in *3-D* obtained by selecting eigenvectors 1, 2, and 3. **d**,**e**) Corresponding results for eigenvectors 2, 3, and 4.

distinct body-parts. This phenomenon is clear in Fig. 7-b-2, where the irregular trajectories of the obtained EM clusters for a sub-sequence of "walk" are visually rendered.

**Real data.** We also measured performances on a large number of real-world, high-resolution voxel-set sequences generated from images captured with our acquisition system (8 synchronized cameras). For sequences for which we do have motion capture data, it is possible to quantitatively measure the performance of all competing methods.

Voxel-set sequences actually captured through an acquisition system suffer from a number of unpleasant phenomena, like presence of large gaps or holes in the grid of voxels, noise, disconnected components, not to mention entire missing body-parts (see Fig. 9-c,f). Fig. 9 illustrates how, unlike EM and for adequate values of the parameters (we used $d = 4, K = 25$ for this experiment), scores are still very high, and the method shows remarkable resilience to unreliable data capture. One can notice a couple of drops in the scores (mirrored by a brief sudden glitch in clusters trajectories, Fig. 9-b,e). They correspond to frames in which gaps are so wide they affect the quality of the segmentation, even though the shape of the embedding cloud remains stable. Fig. 10 shows the scores obtained by all competing methods over other two challenging sequences of real voxel-sets. Our method exhibits strong resilience to data of very poor quality, easily outperforming at the same time both clustering in *3-D* or in a geodesic-based embedding space. At times there are glitches due to extremely corrupted data ($t = 8, t = 17$, right) but the topology adaptation algorithm brings swiftly back the segmentation on track.

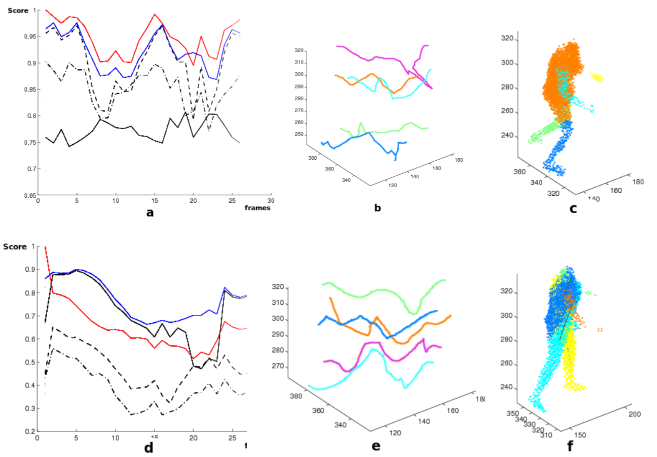


Figure 9. Segmentation scores (**a**,**d**) and centroid trajectories (**b**,**e**) obtained for the real sequence "Mars", from $t = 389$ to $t = 415$. Top row show our results. Bottom row shows the dynamic EM clustering results. The segmentation corresponding to the critical frame $t = 408$ of the sequence is shown (**c**,**f**).

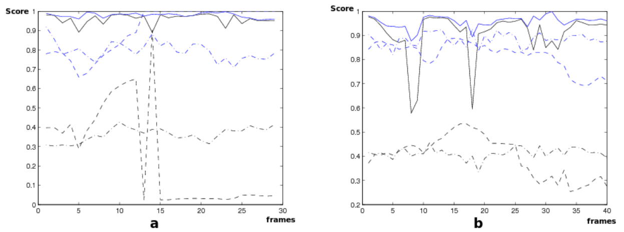
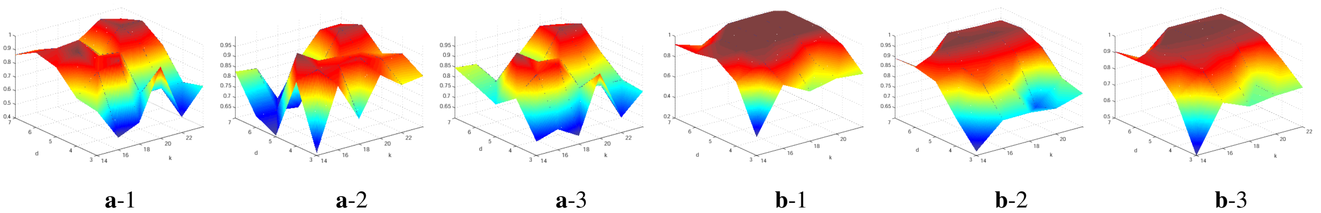


Figure 10. Segmentation scores for two other real sequences, and for all competing methods. Only the coarsening (blue) and a-priori (segmentation depicted on Fig. 6-e) scores are plotted. Solid lines: our method; dashed lines: dynamic EM. Dashdot lines: cluster propagation in ISOMAP space.

**Estimating the optimal number of neighbors.** The proposed segmentation methodology relies critically on the "good" properties of LLE discussed in Section 2, which in turn depend on its two basic parameters: the size $K$ of the neighborhoods and the dimension $d$ of the embedding space. In particular, $K$ affects both the stability of the embedded shape along time and its lower-dimensionality, on which the estimation of the number of clusters itself depends. For excessive values of $K$, some neighborhoods comprise points in different body-parts (Fig. 12-b). In those cases, the farthest element (as it belongs to another, distinct link) is relatively distant from all others. If we plot the distance between this point and all its fellows, we notice a large jump (Fig. 12-d). This is not the case for neighborhoods which span a single rigid part (Fig. 12-a,c). We can set as correct $K$ any of those values which yield only "regular" neighborhoods.

**Selection of eigenfunctions and body-parts.** As they are associated with specific symmetries of the underlying cloud of points, the eigenvectors of the affinity matrix $M$



**a**-1 **a**-2 **a**-3 **b**-1 **b**-2 **b**-3

Figure 11. Consistency (a-1,b-1), segmentation (a-2,b-2) and average ((consistency + segmentation)/2) (a-3,b-3) scores obtained over two example synthetic sequences (Erwan (a) and Space-Surf (b)) for different values of the parameters $K = 14 : 2 : 22$ (on the abscissa) and $d = 3 : 7$ (on the ordinate) of the algorithm. The best performances are achieved for a wide range of the parameters.

that we select after SVD determine the structure of the embedded cloud. It is then more accurate to replace the problem of tuning the dimension $d$ of the embedding space with the question of finding an appropriate *selection of eigenfunctions* of the Laplacian, or pseudo-Laplacian, operator in order to make the structure of the *3-D* shape in terms of protrusions emerge. It is striking that, looking at Fig. 8-a, the positive (in red) or negative (dark blue) peaks of each eigenfunction are actually located on the protrusions of the underlying shape, i.e. its high-curvature regions. Selecting some particular functions is equivalent to selecting the associated peaks, which delivers embedded clouds with a different number of branches. In Fig. 8-b,c, eigenfunctions 1,2,3 determine an embedding where the algorithm is unable to resolve the head, which appears instead when selecting eigenfunctions 2,3,4 (Fig. 8-d,e).

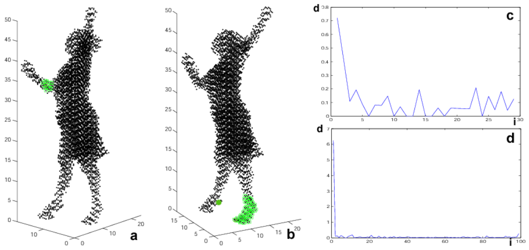
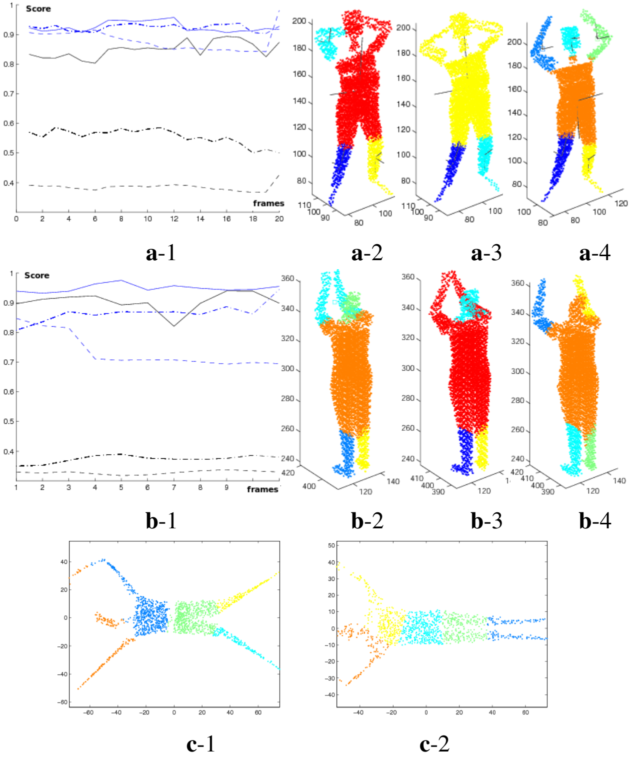


Figure 12. Non admissible values of $K$ are characterized by "anomalous" neighborhoods spanning distinct body-parts (**b**), in opposition to "regular" neighborhoods (**a**). (**c-d**). The corresponding distance (from the center point to all the others).

**Robustness with respect to the parameters.** It is important to assess the sensitivity of the algorithm to the main parameters $K$ and $d$ (or better, the list $[e_1, ..., e_d]$ of the indices of the selected eigenvectors). Fig. 11 quantitatively illustrates how the segmentation scores vary when different values of the parameters $K, d$ are used to compute the embedding (assuming for sake of simplicity that we select the first $d$ eigenfunctions). The stability of both consistency in time and quality of the segmentation in a large region of the parameter space speaks of the robustness of the approach.

**Robustness to topology changes.** No matter how robust Laplacian methods may be, instants in which different parts of the articulated body come to contact still have important effects on the shape of $\{Y_i\}$. These events have instead dramatic consequences on embeddings based on measuring geodesic distances along the body, since new paths appear affecting the distance between all pairs of points in the original cloud. Fig. 13 compares the segmentation scores of methods based on local and global distances in situations in which the topology of the body changes. Propagating clusters in the LLE space exhibits superior results and robustness, as the algorithm smoothly adapts to topology changes owing to the properties of the embedded cloud.



**a**-1 **a**-2 **a**-3 **a**-4

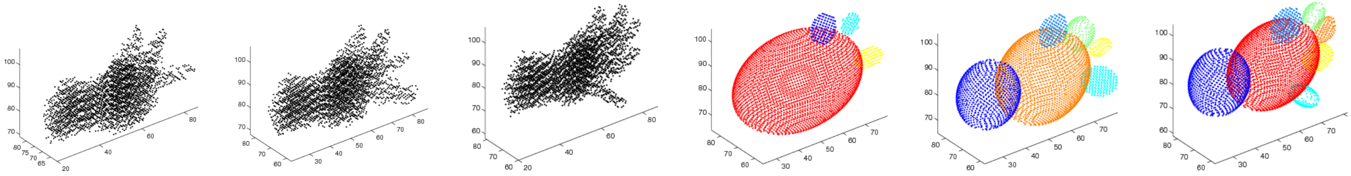
**b**-1 **b**-2 **b**-3 **b**-4

**c**-1 **c**-2

Figure 13. Measuring the relative performances of local methods based on graph Laplacians (represented by LLE and Laplacian Eigenmaps), EM clustering, and global embeddings based on geodesic distance (represented by ISOMAP) for sequences affected by topology changes. **a**-1, **b**-1) Solid - our method, dashed is dynamic EM and dash-dot is k-means in ISOMAP space. Some examples of how the segmentation copes with topology transitions are given (**a**-2,3,4-"wake up", **b**-2,3,4-"clap"). **c**-1,2 are the ISOMAP clustering for the corresponding final frames.

## 5. Conclusions

In this paper we presented a novel dynamic segmentation scheme in which moving articulated bodies are clustered in



**a**-1 **a**-2 **a**-3 **b**-1 **b**-2 **b**-3

Figure 14. **a**-1,2,3) A sequence of voxel-sets capturing a counting hand in an augmented reality environment. **b**-1,2,3) Corresponding rough articulated model fitting based on clusters.

an embedding space, and clusters propagated in time to ensure temporal consistency. By exploiting some desirable characteristic of LLE, we can estimate the optimal number of clusters in order to merge/split clusters in correspondence of topology transitions. We compared the performance of the algorithm versus direct EM clustering in *3-D*, k-means clustering in ISOMAP space, and ground truth labeling provided through motion capture. In our view, the proposed unsupervised segmentation algorithm can be seen as a building block of a wider motion analysis framework, as it provides a coherent body-part segmentation along a sequence. For example, we can fit ellipsoids to the segmented protrusions by aligning the moments or principal axes: Fig. 14 shows the resulting rough model fit to a sequence of voxel-sets representing a counting hand. An extension of the proposed approach to widely separated, non contiguous poses is quite straightforward if we use for cluster propagation methods that match different poses of the same articulated object by aligning their embedded images [13, 4]. The framework can also be naturally applied to certain classes of deformable objects, for as long as deformations preserve the volume of the body, the eigenvalues of the graph Laplacian remain stable.

## References

[1] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *CVPR'05*, 2005.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS'01*, 2001.

[3] M. Brand. Shadow puppetry. In *ICCV'99*, 1999.

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.

[5] G.-J. Brostow, I. Essa, D. Steedly, and V. Kwatra. Novel Skeletal Representation For Articulated Creatures. In *ECCV'04*, 2004.

[6] C.-W. Chu, O. C. Jenkins, and M. J. Mataric. Markerless kinematic model and motion capture from volume sequences. In *CVPR'03*, 2003.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977.

[8] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[9] A. Elgammal and C. Lee. Inferring 3D body Pose from Silhouettes using Activity Manifold Learning. In *CVPR'04*, 2004.

[10] D. Gavrila and L. Davis. 3D model-based tracking of humans in action: a multi-view approach . In *CVPR'96*, 1996.

[11] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV'03*, 2003.

[12] D. Hogg. Model Based vision: a program to see a walking person. In *Image and Vision Computing*, 1983.

[13] V. Jain and H. Zhang. Robust 3d shape correspondence in the spectral domain. In *SMI'06*, 2006.

[14] O. Jenkins and M. Mataric. A spatio-temporal extension to isomap nonlinear dimension reduction. In *ICML'04*, 2004.

[15] B. Lévy. Laplace-Beltrami eigenfunctions: Towards and algorithm that understands geometry. In *SMI'06*, 2006.

[16] R. Lin, C.-B. Liu, M.-H. Yang, N. Ahuja, and S. Levinson. Learning Nonlinear Manifolds from Time Series. In *ECCV'06*, 2006.

[17] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3), 2006.

[18] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS'01*, 2001.

[19] G. Rosman, A. Bronstein, M. Bronstein, and R. Kimmel. Manifold analysis by topologically constrained isometric embedding. *International Journal of Applied Mathematics and Computer Sciences*, 1(3):117–123, 2004.

[20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[22] A. Sundaresan and R. Chellappa. Segmentation and Probalistic Registration of Articulated Body Models. In *ICPR'06*, 2006.

[23] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[24] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph.*, 24(3):496–503, 2005.