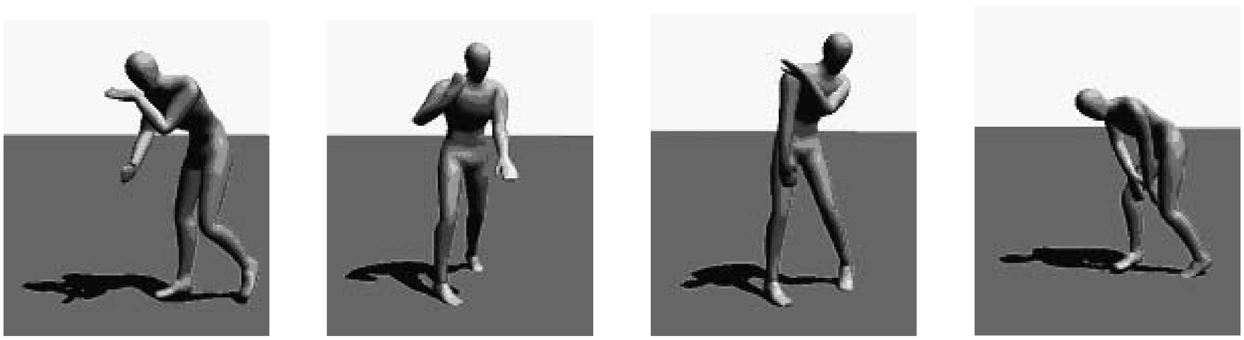
Using Nonlinear Dimensionality Reduction in 3D Figure Animation

A. Elizabeth Seward
Dept. of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235
anne.e.seward@vanderbilt.edu

Bobby Bodenheimer
Dept. of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235
bobbyb@vuse.vanderbilt.edu



Example poses from a re-sequenced hip-hop dance motion.

ABSTRACT

This paper explores a method for re-sequencing an existing set of animation, specifically motion capture data, to generate new motion. Re-using animation is helpful in designing virtual environments and creating video games for reasons of cost and efficiency. This paper demonstrates that through nonlinear dimensionality reduction and frame re-sequencing, visually compelling motion can be produced from a set of motion capture data. The technique presented uses Isomap and ST-Isomap to reduce the dimensionality of the data set. Two distance metrics for nonlinear dimensionality reduction are compared as well as the effect of global degrees of freedom on the visual appeal of the newly generated motion.

Keywords

3D figure animation, re-sequencing, motion capture, dimensionality reduction

1. INTRODUCTION

Designing a rich repertoire of behaviors for virtual humans is an important problem for virtual environments and computer games. Traditionally, animation is generated for these environments through keyframing, motion capture, or dynamic simulation. Dynamic simulation is a computationally expensive process that has so far been limited by difficulty in designing controllers for behaviors. The main drawback of animation technology such as keyframing and motion capture is lack of flexibility in re-using or editing the animation. What is keyframed or captured is difficult to use in new contexts.

This paper investigates methods for re-using animation data to generate new motion. Given an existing library of data, the problem is to rearrange the individual frames of motion into a new motion that is visually appealing. To accomplish this, we employ techniques of nonlinear dimensionality reduction to extract a parameterization of the data. Most animation data has over 40 degrees of freedom, and is thus contained spatially in a 40+-dimensional vector space. We assume that a body of coherent animation data, i.e., human motion, describes a low dimensional manifold that is embedded in this high-dimensional space. By parameterizing the manifold, we develop the ability to traverse the data in a non-sequential manner while maintaining coherence among the poses, thus generating new poses.

This paper explores two techniques for nonlinear dimensionality reduction, Isomap and ST-Isomap. These techniques are better able to capture the underlying manifold of a complicated articulated figure than traditional linear methods such as Principal Component Analysis. We also examine the use of different distance metrics to characterize distances in the higher dimensional space. Our results show ST-Isomap to be preferable to Isomap due to its incorporation of temporal information. In addition, we find the inclusion of global degrees-of-freedom in the reduction to be more visually compelling in free-form motions such as dancing, whereas in cyclic locomotion such as walking they are better omitted.

The paper is organized as follows. In Section 2 we provide background information on this area of animation, nonlinear dimensionality reduction, and distance metrics. In Section 3 we describe the techniques used to evaluate a body of animation data. Finally, in Section 4 we discuss these results and provide directions for future work.

2. BACKGROUND

In this section we place our work in context.

2.1 3D Animation Re-Sequencing

Motion capture research has concentrated on studying ways of editing and modifying existing motions. See Gleicher[4] for a survey of work in this area. As mentioned previously, work in minimizing the amount of manual editing needed for using motion capture has been approached in two primary ways: through the use of probabilistic methods that synthesize new motion and through methods that re-use the original data for synthesis.

In contrast, other researchers have drawn inspiration from the work of Schödl et al. [15] on video textures to retain the original motion sequences but play them back in non-repetitive streams. Video textures is similar to our goal of re-sequencing animation data, specifically the "video-based animation" section of their work, although it is an image-based technique and does not directly generalize to 3D animation data. They use the L_2 distance to compute the differences between frames for building the video structure. We want to compare the differences between frames in a similar fashion to analyze the data for re-sequencing. Schödl et al. [15] assume a large data set with incremental changes between frames. In their follow-up work [14], they use user-directed video sprites for creating new character animations.

For 3D motion data, Sidenbladh et al. [17] employ a probabilistic search method to find the next pose in a motion stream and obtain it from a motion database. Arikan and Forsyth [1] construct a hierarchy of graphs connecting a motion database and use randomized search to extract motion satisfying specified constraints. Kovar and Gleicher [7, 8] use dynamic programming to locally parameterize large data sets of motion and to construct a directed graph of motion that can be traversed to generate different styles of motion. Lee et al. [11] model motion as a first-order Markov process and also construct a graph of motion. They demonstrate three interfaces for controlling the traversal of their graph.

One of the features of this body of research is that they employ different underlying cost metrics for evaluating transition points in the graph. Lee et al. use a cost function based on joint orientations and velocities. Kovar et al. use a cost function based on the distance between point samples of the mesh representation of the character. Arikan and Forsyth use a hybrid method similar to that of Lee but involving joint accelerations as well. Sidenbladh et al. use a probabilistic search method. In our work, we will use the L_2 distance between orientations of the articulated figure and compare it to the metric of Lee et al. However, we modify the weights used in the Lee metric according to the results of [19].

2.2 Dimensionality Reduction

Dimensionality reduction for large data sets seeks to describe a high-dimensional vector space in a low-dimensional embedding. A commonly used dimensionality reduction method is Principle Component Analysis (PCA) [6], a linear embedding technique that generates mean data and eigenvectors that span the principle shape variations in the data space. However, this technique does not retain the spatio-temporal structure in the data that we are seeking. We assume our data have some underlying spatial surface (manifold) for which we wish to discover an embedding into a lower-dimensional space. Multidimensional scaling (MDS)[10] is another approach that finds an embedding that preserves the pairwise distances, equivalent to PCA when those distances are Euclidean. However, many data sets contain essential nonlinear structures that are invisible to PCA and MDS.

Two techniques for manifold-based nonlinear dimensionality reduction are Isomap [18] and Locally Linear Embeddings (LLE) [13]. Both methods use local neighborhoods of nearby data to find a low-dimensional manifold embedded in a high-dimensional space. However, neither of these methods account for temporal structure in animation. A modified version of Isomap, called Spatio-Temporal Isomap (ST-Isomap) [5], can account for the temporal dependencies between sequentially adjacent frames. We borrow the idea of extending Isomap using temporal neighborhoods from [5], and use ST-Isomap for dimensionality reduction of animation data to maintain the temporal structure in the embedding. Jenkins and Matarić [5] focus on synthesizing humanoid motions from a motion database by automatically learning motion vocabularies. Starting with manually segmented motion capture data, ST-Isomap is applied to the motion segments in two passes, along with clustering techniques for each of the resulting sets of embeddings. Motion primitives and behaviors are then extracted and used for motion synthesis.

3. METHODS

Isomap and ST-Isomap work by computing a distance matrix between all pairs of input data. ST-Isomap adjusts this distance matrix to account for temporal neighbors. Both algorithms next estimate the geodesic distances (distances along the manifold, as opposed to in the high-dimensional space) by computing the all-pairs shortest paths from the original distance matrix. MDS is then applied to construct a lower dimensional embedding of the data. So for our work, motion capture data is preprocessed and the distance matrix is computed. Then, nonlinear dimensionality reduction is used to understand the structural aspects of the data. The shortest cost path is computed between the start frame and the end frame, and the data is re-sequenced using this shortest cost path.

3.1 Pre-Processing Motion Capture Data

Our data set consists of three sets of motion capture data taken from the motion capture data at Carnegie Mellon University [3], and the Georgia Institute of Technology. The first set of motion depicts a *walking* human and is 516 frames long. The second set of motion represents a *running* human and is 405 frames in length. The third set of motion is that of a human dancing in a *hip-hop* style and is 800 frames long. Motion capture data is temporal data depicting the animation of an articulated figure consisting of rigid bodies connected by joints [2]. The orientation of the joints is represented by a quaternion for each joint [16]. Each joint typically expresses three degrees of freedom for the articulated figure. In addition, there are six additional degrees of freedom corresponding to a global position (three) and global orientation (three).

As mentioned above, quaternions are used to represent orientation. However, nonlinear dimensionality reduction tools assume metrics in Euclidean space. Therefore, to use such tools, the quaternions must be linearized into vectors. This step is accomplished using a technique described by Lee and Shin [12]. In this technique, quaternions are projected onto the tangent space of the quaternion sphere determined by a pre-selected quaternion, as depicted in Figure 1. For each joint k, given a sequence of quaternions $\{q_{i,k}\}_{i=0}^{n}$, the projection of each quaternion into Euclidean space (the tangent

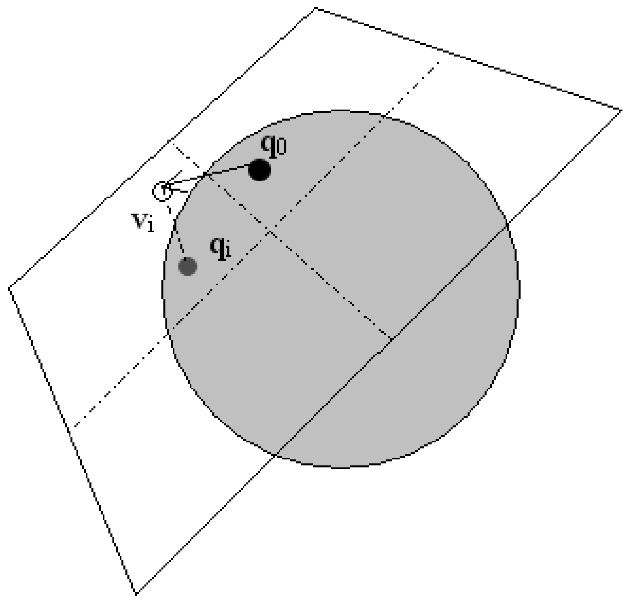


Figure 1: Depiction of the method for linearizing quaternions. A candidate quaternion, denoted by q_0, is selected and the tangent space to the quaternion sphere is formed. Other quaternions (q_i) are projected into the tangent space yielding a vector v_i.

space of the quaternion sphere at $q_{0,k}$) is given by

$$\nu_{i,k} = \sum_{j=0}^{i-1} \log\left(q_{j,k}^{-1} q_{j+1,k}\right)$$

where $\nu_{i,k}$ is the vector, or the linearized quaternion, for joint k at frame i. All joint angles for every frame are linearized in this fashion. Let m represent the number of joints for a particular set of data, and let n be the number of frames for that data set. The vectors resulting from this operation are formed into a $3m \times n$ matrix. Each column of the matrix contains n vectors, each representing a joint, stacked consecutively. Thus, each column of the matrix represents a frame of motion. We will denote the ith column or frame as V_i in the following.

3.2 Distance Metrics

The input distance matrix for the nonlinear dimensionality reduction tools is now computed. Each entry of the distance matrix is given by $D_{ij} = ||V_j - V_i||$, where the norm is described below. An important aspect of the dimensionality reduction process is the creation of this matrix. We use two different distance metrics (norms), the L_2 distance and the Lee distance [11], to compute the local neighborhoods. The L_2 distance is simply the Euclidean distance between two frames considered as vectors. The Lee metric is more complicated. Given two frames i and j, the Lee distance is computed

$$D_{ij} = d(p_i, p_j) + d(v_i, v_j) \quad (1)$$

where $d(v_i, v_j)$ is the weighted distance of joint velocities, ν weights the velocity difference with respect to $d(p_i, p_j)$, and $d(p_i, p_j)$ is the weighted difference of joint orientations. This term is given by

$$d(p_i, p_j) = \|p_{i,0} - p_{j,0}\|^2 + \sum_{k=1}^{m} w_k \left\|\log\left(q_{j,k}^{-1} q_{i,k}\right)\right\|^2 . \quad (2)$$

Right and Left Hip	1.0000
Right and Left Knee	0.0901
Right and Left Shoulder	0.7884
Right and Left Elbow	0.0247

Table 1: Joints with non-zero weights for the Lee distance metric. The weights on other joints are set to zero.

Motion	Global Orientation	Reduction Method	Distance Metric	Number of k-nearest neighbors
Hip-Hop	Included	Isomap	L_2	22
		ST-Isomap	L_2	1
		Isomap	Lee	1
		ST-Isomap	Lee	1
Walk	Excluded	Isomap	L_2	79
		ST-Isomap	L_2	79
		Isomap	Lee	3
		ST-Isomap	Lee	1
Run	Excluded	Isomap	L_2	107
		ST-Isomap	L_2	107
		Isomap	Lee	7
		ST-Isomap	Lee	1

Table 2: Minimum number of nearest neighbors to achieve a connected component for the motion datasets and dimensionality reduction method. Also noted is whether the global degrees of freedom are included in the dimensionality processing.

In Equation 2, $p_{i,0}, p_{j,0} \in \mathbf{R}^3$ are the global translational positions of the figure at frames i and j, respectively; m is the number of joints in the figure; $q_{i,k}, q_{j,k}$ are the orientations of joint k and frames i and j, respectively, expressed as quaternions. The log-norm term represents the geodesic norm in quaternion space, and each term is weighted by w_k. We use the optimal weights determined by Wang and Bodenheimer [19] and shown in Table 1. The velocity term, $d(v_i, v_j)$, is computed similarly.

3.3 Dimensionality Reduction

An important question in re-sequencing motion capture data is how to handle global degrees of freedom. Thus, for input into Isomap and ST-Isomap, we examine the effects of including and excluding the global degrees of freedom. To find a path in the embedding from a start frame to an end frame, it is necessary to have one connected component in the embedding. What this means is that the nearest neighbor distances in the higher-dimensional space are not "too large." As a result, the number of k-nearest neighbors is set to the minimum size needed to achieve this specification. This number is discovered through trial-and-error. The number of nearest neighbors is shown in Table 2.

The resulting embedding structure for the hip-hop dancing motion analyzed using ST-Isomap is shown in Figure 2. In this figure, each node of the graph represents a frame of data, with its connection structure as determined by the distance metric and pairwise distance matrix. It is clear that there is considerable structure in the data. Figure 3 shows the residual variances (error) if the high dimensional data is represented as a manifold of dimension d. In this case the optimal dimension is four, although Figure 2 necessarily shows a two-dimensional representation.

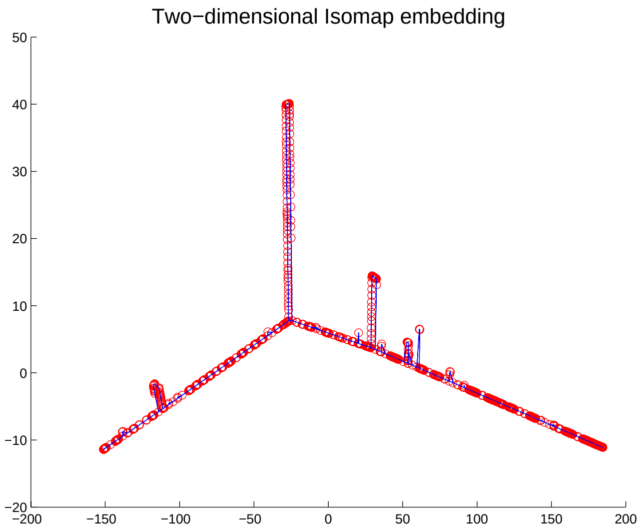


Figure 2: The ST-Isomap structure for the hip-hop dancing motion with global degrees of freedom included. The red circles indicate a frame of data and the blue lines show connectivity information as determined by the distance matrix.

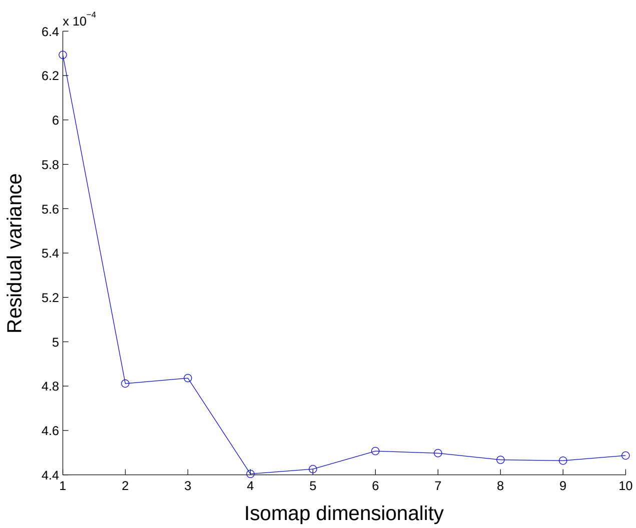


Figure 3: The residual variances for the ST-Isomap dimensionality reduction. The horizontal axis represents a candidate embedding dimensionality.

3.4 Shortest Cost Path and Re-Sequencing

To generate the re-sequenced motion, the Isomap or ST-Isomap embedding is traversed from a given start frame to a given end frame. If needed, keyframes from the data are added to make the length of the total motion sequence appropriate. Dijkstra's algorithm is used to compute the shortest cost path from one frame to another. This path is an array of frame numbers corresponding to frames in the original motion. These resulting frames are then extracted from the initial motion and played back.

4. RESULTS AND DISCUSSION

This paper has explored the use of nonlinear dimensionality reduction techniques to re-sequence animation. The main issues that we encountered are the inclusion or exclusion of global degrees of freedom, the use of Isomap or ST-Isomap as a nonlinear dimensionality reduction tool, and the selection of a distance metric, either the L2 metric or the more sophisticated Lee metric. We examined the results of these issues with three sets of motion capture data: a hip-hop dancing motion, a walking motion, and a running motion. Including global degrees of freedom appears to work well for only free-form motions. Free-form motions are defined as motions for which people do not hold an a priori expectation of the next move. Examples of such motions include martial arts, boxing, and dancing. In contrast, cyclic linear motions are characterized by the animated figure moving in only one direction with a single movement repeated several times. Examples of such motions include walking and running.

Including global degrees of freedom for cyclic linear motions hurts the smoothness and visual appeal of the re-sequenced motion. Frames in the shortest cost path are generally not in the same order as they are in the original motion, thereby causing the figure to slide from the background to the foreground multiple times. However, including the global degrees of freedom for free-form motions appears to improve their visual coherence. Because the figure is not expected to move in any particular direction at any time, the varying order of the frames does not destroy the visual appeal of the motion. In fact, including the global degrees of freedom keeps the figure from being pinned to one location and enhances the look of the motion.

For animated data, we believe ST-Isomap is a better choice than Isomap. Sometimes, Isomap and ST-Isomap produce identical arrays of shortest cost paths. However, when these arrays differ, ST-Isomap seems to yield a more visually compelling motion with fewer jerks than the motion extracted from Isomap's output. The reason for this difference can most likely be attributed to ST-Isomap's preservation of temporal aspects of the data, which is important for animation.

Another interesting aspect of our work involves the number of k-nearest neighbors selected for Isomap and ST-Isomap to achieve one connected component. This number may reflect the intrinsic sparseness of the data. For animation, dense data sets would be helpful in re-sequencing. However, not all data sets exhibit this characteristic. Perhaps the number of k-nearest neighbors provides a metric for how well a data set can be re-sequenced.

We cannot make any definitive conclusions about the distance metric based on this work. For the hip-hop and walk motions, the L_2 distance seems to yield the more visually coherent motion. However, for the run motion, the Lee distance appears to give a better motion with greater smoothness. In future work, we plan to test our methods on more data sets and draw a conclusion about whether the L_2 or Lee distance generally produces more appealing motions.

We believe that this work has demonstrated the necessity of data post-processing following nonlinear dimensionality reduction to ensure visual coherence for cyclic linear motions. Thus, future investigation will involve the discovery of such techniques for post-processing motions. Perhaps dynamically simulating the center of mass of the data by providing re-calculated global degrees of freedom coupled with footskate cleanup mechanisms is required to ensure visual appeal [9]. This issue will be our main avenue of future exploration.

5. REFERENCES

[1] ARIKAN, O., AND FORSYTH, D. A. Interactive motion generation from examples. *ACM Transactions on Graphics 21*, 3 (July 2002), 483–490. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[2] BODENHEIMER, B., ROSE, C., ROSENTHAL, S., AND PELLA, J. The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97* (Sept. 1997), D. Thalmann and M. van de Panne, Eds., Springer NY, pp. 3–18. Eurographics Animation Workshop.

[3] http://mocap.cs.cmu.edu.

[4] GLEICHER, M. Comparing constraint-based motion editing methods. *Graphical Models 63*, 2 (March 2001), 107–134. ISSN 1524-0703.

[5] JENKINS, O. C., AND MATARIC, M. J. Automated derivation of behavior vocabularies for autonomous humanoid motion. In *Proc. of the 2nd Intl. joint conf. on Autonomous agents and multiagent systems* (2003), pp. 225–232.

[6] JOLLIFFE, I. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[7] KOVAR, L., AND GLEICHER, M. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 559–568. Proceedings of ACM SIGGRAPH 2004.

[8] KOVAR, L., GLEICHER, M., AND PIGHIN, F. Motion graphs. *ACM Transactions on Graphics 21*, 3 (July 2002), 473–482. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[9] KOVAR, L., SCHREINER, J., AND GLEICHER, M. Footskate cleanup for motion capture editing. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 97–104.

[10] KRUSKAL, J. B., AND WISH, M. *Multidimensional Scaling*. Sage Publications, Beverly Hills, 1978.

[11] LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics 21*, 3 (July 2002), 491–500. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[12] LEE, J., AND SHIN, S. Y. General construction of time-domain filters for orientation data. *IEEE Transactions on Visualization and Computer Graphics 8*, 2 (April - June 2002), 119–128. ISSN 1077-2626.

[13] ROWEIS, S., AND SAUL, L. Nonlinear dimensionality reduction by locally linear embedding. *Science 290* (2000), 2323–2326.

[14] SCHÖDL, A., AND ESSA, I. Controlled animation of video sprites. In *Symposium on Computer Animation* (2001), ACM Press / ACM SIGGRAPH, pp. 121–127.

[15] SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. Video textures. In *Proceedings of ACM SIGGRAPH 2000* (July 2000), Computer Graphics Proceedings, Annual Conference Series, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 489–498. ISBN 1-58113-208-5.

[16] SHOEMAKE, K. Animating rotation with quaternion curves. *ACM Transactions on Graphics 19*, 3 (July 1985), 245–254.

[17] SIDENBLADH, H., BLACK, M. J., AND SIGAL, L. Implicit probabilistic models of human motion for synthesis and tracking. In *Computer Vistion — ECCV 2002 (1)* (Copenhagen, Denmark, May 2002), A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., Lecture Notes in Computer Science, Springer-Verlag, pp. 784–800. 7th European Conference on Computer Vision.

[18] TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290* (2000), 2319–2323.

[19] WANG, J., AND BODENHEIMER, B. An evaluation of a cost metric for selecting transitions between motion segments. In *Symposium on Computer Animation 2003* (San Diego, CA, July 2003), D. Breen and M. Lin, Eds., ACM SIGGRAPH/Eurographics, pp. 232–238.