Spatio-temporal LTSA and Its Application to Motion Decomposition

Hongyu Li[1,2], Junyu Niu[3,⋆], Lin Zhang[2], and Bo Hu[1]

[1] Electronic Engineering Department, Fudan University, Shanghai, China
[2] School of Software Engineering, Tongji University, Shanghai, China
[3] School of Computer Science, Fudan University, Shanghai, China

Abstract. This paper describes a STLTSA-based framework to analyze and decompose human motion for synthesis. In this work, we mainly intend to extend a manifold learning method, local tangent space alignment, to a spatio–temporal version for manifold analysis and offer an effective method of estimating the intrinsic dimensionality of motion data. Based on an assumption that a long sequence of motion is composed of a number of short motion units, we can decompose a motion into several basic motion units in a low-dimensional manifold space and extract motion cycles from the cyclic unit. The generation of new complex movement using obtained motion units is feasible and promising.

Keywords: Manifold Learning, Spatio-Temporal Neighborhood, Motion Learning, Motion Decomposition.

1 Introduction

When data lies on a low-dimensional manifold, its structure may be highly nonlinear, hence linear dimensionality reduction methods such as principal component analysis (PCA) [1] and metric multi-dimensional scaling (MDS) [2] often fail in finding the nonlinear embeddings. This has motivated extensive efforts toward developing nonlinear dimensionality reduction methods which is known as manifold learning. Manifold learning methods can be categorized into two main groups: global and local techniques. Global techniques attempt to preserve global properties of the data lying on manifolds [3]. Local techniques attempt to retain global properties of the data by preserving local properties obtained from neighborhoods around data points [4, 5].

In general, motion data is of high dimensionality and difficult to understand and analyze, its intrinsic DOFs, however, are essentially supposed to be quite few and easy to visualize. Linear methods have been widely used in [6, 7] to reduce the dimensionality of human motion for motion analysis. Recently, Wang et al [8] introduce Gaussian process dynamical models to learn nonlinear models of human motion from high-dimensional motion capture data. Li et al [9] propose a method to learn a nonlinear low-dimensional manifold for high-dimensional time series and model the dynamical process in the manifold space.

⋆ Corresponding author.

T. Huang et al. (Eds.): ICONIP 2012, Part V, LNCS 7667, pp. 498–505, 2012.

In this paper, we aim to extend locality-based manifold learning techniques to a spatio-temporal version for motion capture data. The proposed method could significantly reduce the time cost of constructing the similarity graph, which facilitates to handle the large scale data sets. According to the work [10], the local tangent space alignment (LTSA) [5] method generally performs best among the popular manifold learning techniques, it is therefore chosen to test the spatio-temporal similarity graph. In addition, an effective method is offered to estimate the intrinsic dimensionality of motion data in this study.

2 Spatio-temporal LTSA

2.1 Spatio-temporal Neighborhood Construction

To estimate local tangent spaces of a manifold, the original LTSA method requires to first construct a similarity neighborhood through selecting nearest neighbors of each point. The simplest way to construct a similarity neighborhood is to identify a fixed number k of nearest neighbors per data point according to spatial distance. Although the k-nearest neighborhood is good at describing local structure of data, such neighborhood construction has its own drawbacks to handle large scale data sets. The construction cost for k-nearest neighborhood is $\mathrm{O}(kn^2)$, which is expensive in large scale situations.

However, for time-dependent data such as motion data, the variation of data in two continuous frames is fairly low. Therefore, the temporal neighborhood implicitly contains much cue about spatial neighbors. If the temporal distances is also taken into consideration, the construction cost for k-nearest neighborhood will be dramatically reduced. Our neighborhood construction strategy is to first select $2k$ sequential frames as initial neighbors backward and forward from the current frame, and then find the k-nearest neighbors to construct a similarity graph using the spatial distance. For example, taking $k = 5$, the nearest neighbors of the i-th frame will be found from frames between $i-5$ and $i+5$ according to their spatial distance.

Given n time-dependent data points, the time complexity is $\mathrm{O}(k^2n)$ if the k-nearest neighbors of each point are selected in terms of both the spatial and temporal distance. Since $k \ll n$ in general, using the spatio-temporal distance will greatly improve the construction efficiency of the neighborhood in comparison with solely using the spatial distance. Meanwhile, as two continuous frames vary little in motion data, our construction strategy can faithfully describe the local geometrical structure in data.

2.2 Summary of the Algorithm

Next we briefly describe in Table 1 how to extract low-dimensional coordinates Y from a set of high-dimensional motion data X with STLTSA.

It is worth that there are two free parameters, k and d, as input in the proposed method. It has been discussed in [5] that if the parameter k is too small, the

Table 1. The STLTSA algorithm

Input: the dataset $X = \{x_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^m$, the number k of nearest neighbors, and the dimensionality d of the embedded manifold.
1. Construct the spatio-temporal neighborhood represented in the form of a $m \times k$ matrix, $X_i = (x_i^j)$, for each point x_i. Column vector x_i^j is the j-th nearest neighbor of x_i. 2. Calculate the d largest eigenvectors $g_1, \ldots, g_d$ of the correlation matrix $(X_i - \bar{x}_i e^T)^T (X_i - \bar{x}_i e^T)$. e is an all-one column vector, and $\bar{x}_i$ represents the average of the neighborhood of x_i: $\bar{x}_i = \frac{1}{k} \sum_j x_i^j$. 3. Extract the local geometry G_i by setting $G_i = [e/\sqrt{k}, g_1, \ldots, g_d]$. 4. Construct the $n \times n$ alignment matrix B by locally summing as follows: $B(I_i, I_i) \leftarrow B(I_i, I_i) + I - G_i G_i^T, i = 1, \ldots, n$ with initial $B = 0$. I is a $k \times k$ identity matrix, I_i denotes the set of indices for the k-nearest neighbors of x_i. 5. Compute the $d+1$ smallest eigenvectors of B and pick up the eigenvector matrix $[u_2, \ldots, u_{d+1}]$ corresponding to the $2nd$ to $d+1st$ smallest eigenvalues.
Output: the global coordinates $Y = [y_1, \ldots, y_n] = [u_2, \ \ldots, u_{d+1}]^T$.

mapping will not reflect any global properties of data; if k is too large, the mapping will lose its nonlinear character and behave like traditional PCA as the entire data set is seen as the local neighborhood. However, the algorithm is essentially stable over a wide range of values of k. How to determine the intrinsic dimensionality d is introduced in the following section.

2.3 Intrinsic Dimensionality

It is well known that PCA [1] exploits the number of large singular values of the covariance matrix of input data to estimate intrinsic dimensionality. Furthermore, a similar estimate of LLE was also proposed by Polito and Perona [11], where $d+1$ should be less than or equal to the number of eigenvalues of a kernel matrix that are close to zero.

Likewise, one could estimate the dimensionality d with the eigengap trick in spatio-temporal LTSA. The number of eigenvalues of B that are close to zero gives us an answer that d should be no more than this number. Precisely speaking, the minimal d is considered as the intrinsic dimensionality if it satisfies that the eigengap, $|\lambda_d - \lambda_{d+1}|$, between the eigenvalues λ_d and λ_{d+1} of matrix B is more than the threshold τ,

$$|\lambda_d - \lambda_{d+1}| > \tau.$$

Due to the fact that DOFs of human motion are quite few, the intrinsic dimensionality of motion data is clearly very low. Using the walking motion data, we find that the dimensionality d is supposed to be equal to 3. This agrees with the fact that there exist exactly three DOFs that describe the rotation of joints in the motion capture data we use.

2.4 Dynamic Mapping

In this work, we provide a way of dynamic mapping, which can project new data points between the low-dimensional manifold space and the original high-dimensional space. The basic assumption of this mapping is that there exists a locally linear mapping between the original space and the manifold space, which is consistent with the derivation of LTSA [5]. Therefore, once x_j (y_j) and x_{n+1} (y_{n+1}) lie close enough to each other, the transformation matrix of x_j (y_j) is naturally applicable to x_{n+1} (y_{n+1}). Note that this assumption requires that the input data must be dense enough to sufficiently cover the whole surface of the embedded manifold, otherwise the generalization can not perform well.

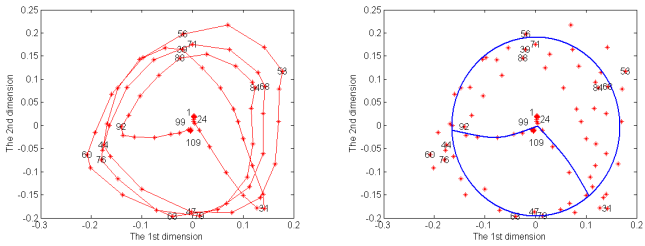


Fig. 1. The 2D manifold description of the walking motion with STLTSA. Red stars denote the mapping results of the input motion data into the 2D manifold space, the number is the frame index. (a): the normal walking path connected along the frame order, (b): the simplification of the walking path as a circle plus two curves.

2.5 Analysis

Using the walking motion containing 109 frames of 54-dimensional motion capture data with three walking cycles as an example, we applied the STLTSA method to such data and projected them into a 2D manifold space for visual analysis. Fig. 1 presents the 2D description of the walking motion in this manifold space, where a red star corresponds to an action in the walking motion sequence and the number is the frame index. The manifold curve in Fig. 1(a) depicts a transition path of actions connected along the time order. The transition path can be considered as the concatenation of three sub-paths: first transferring from the beginning (No. 1) to the initial action of walking (No. 39); then walking three cycles(from No. 40 through No. 88) and finally returning to the end (No. 109) close to the beginning. Although the walking trajectory is not completely identical in each cycle due to the liberty of human motion, the cyclic characteristic of such motion is still clearly expressed in the manifold space. Thus this walking path can be simplified and approximately sketched as a circle plus two curves, as shown in Fig. 1(b).

3 Motion Decomposition

According to the geometrical analysis of the reduced motion data, it is easy to deconstruct human motion in the manifold space. Assuming that a complex human behavior is always composed of several small motion units each of which represents a simpler behavior, one can decompose a long motion sequence into some small motion units. Sometimes the low-dimensional manifold curve may not be continuous and smooth due to the liberty and instability of human motion. So slight modification of the manifold coordinates is often required for denoising. In this section, we will put a special emphasis on how effectively decomposing cyclic motion and how removing noise from the manifold coordinates.

3.1 Decomposition of Cyclic Motion

In particular, the decomposition of a motion sequence M can be formulated as the sum of different motion units M_{u_i}, $M = \sum_i M_{u_i}$. Motion units in acyclic motion usually have the uncertain type and amount, therefore we will use cyclic motion as examples to introduce motion decomposition. For a whole cyclic motion sequence, it generally contains five basic motion units: the preparing unit (M_p), the initial unit (M_i), the cyclic unit (M_c), the final unit (M_f), the wind-up unit (M_w). M_c is still divisible as the sum of N cycles: $M_c = \sum C_u = N * C_u$, where C_u denotes a cycle that is a primitive action.

If the manifold curve computed with STLTSA is relatively smooth and continuous, motion decomposition can be easily completed with the derivative of this curve. In particular, the preparing unit starts from the beginning through the position where the first derivative of this curve changes sharply; the initial unit ends at the first position where the second derivative is zero; the cyclic unit continues until the last position with the second derivative being zero; the final unit is over if the manifold coordinates do not change obviously; the remainder is the wind-up unit. In the real applications, we use the first and second order difference of manifold coordinates with respect to time instead of the derivative of the manifold curve,

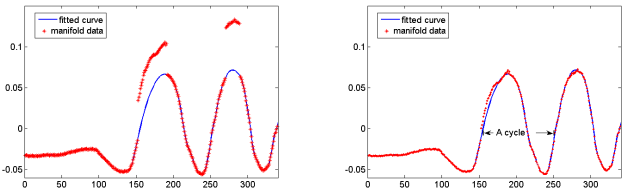
$$f'_y(t) \approx f_y(t+1) - f_y(t),$$
$$f''_y(t) \approx f_y(t+1) - 2f_y(t) + f_y(t-1),$$

where t is the frame index ranging in $[1, n]$, and $f_y(t)$ denotes manifold coordinates $Y = \{y_1, \cdots, y_n\}$ computed with STLTSA. In this means, the extraction of a cycle contained in the cyclic unit would be fairly simple. With the first order difference $f'_y(t)$ being close 0, we can find the valley and peak in the manifold curve. Between each pair of peak and valley alternately appearing in the time order, a point where the second order difference $f''_y(t)$ is closest to 0 will be picked out as a candidate of the margin of a cycle. Generally, the first and third candidates construct a motion cycle in the cyclic unit. As shown in Fig. 2(b), a pirouette cycle is automatically extracted in this means.

3.2 Denoising

Each motion cycle in real applications is just approximately consistent rather than completely identical due to the randomness and liberty of human action. Therefore, the cyclic unit may not take on such perfect periodicity in the manifold space. To recover the intrinsic shape of the manifold curve, one has to remove noise from manifold coordinates through smooth curve fitting. The task of curve fitting can be completed in terms of the least square method. The noise in manifold coordinates can be removed through translating noisy points toward the fitted curve. Noisy manifold coordinates $N_p(t)$ are defined as those points that deviate the fitted curve $f_c(t)$: $N_p(t) = \{f_y(t) \mid \|f_c(t) - f_y(t)\| > \gamma)\}$, where γ is a predefined threshold close to 0. The translation distance d_t of noisy manifold coordinates can be simply computed through the average deviation distance of the noise points from the fitted curve, After translation, the noisy manifold coordinates are changed into

$$\tilde{f}_y(t_i) = \begin{cases} f_y(t_i) - d_t, \, if \;\; f_y(t_i) \geq f_c(t_i); \\ f_y(t_i) + d_t, \, if \;\; f_y(t_i) < f_c(t_i). \end{cases} \tag{1}$$



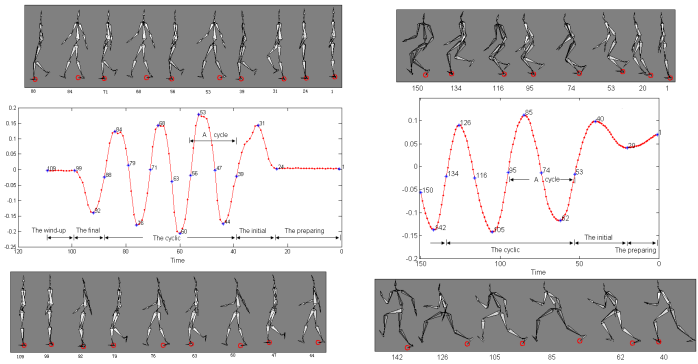
(a) Before translation (b) After translation

Fig. 2. An example of translation about the pirouette data. The reduced manifold coordinates are marked with red stars and blue curve represents the fitted curve. (a): before translation, (b): after translation.

Fig. 2 gives an example of translation for denoising. The pirouette data is tested with STLTSA and the 1D manifold coordinates are displayed with red stars in this figure. The blue curve represents the fitted curve and the x-axis the time order. As in Fig. 2(a), some points in these two pirouette cycles obviously deviate from the fitted manifold curve, arising from the surrounding noise. These disjointed points are subsequently translated to the proximity of the fitted curve according to the formula (1). The results after translation are shown in Fig. 2(b), which reveals better periodic regularity.

4 Experimental Results

Here two examples regarding walking and running are offered to illustrate motion decomposition in Fig. 3. The 1D manifold coordinates (y-axis) are extracted



(a) Walking (b) Running

Fig. 3. The periodicity of human walking and running. The middle shows the variation of 1D manifold coordinates obtained by STLTSA with time (from right to left for visual consistency). Some postures corresponding to numbered points in the middle are shown in the top and bottom.

using with STLTSA and are shown with the variation of time (x-axis). The walking data describes a series of human actions, beginning with the "attention" posture, then lifting up his left foot and going forward for 7 steps, and finally returning the "attention" posture. These steps actually contain three walking cycles each of which is composed of two steps. For simplification, only 109 data points are used in our experiments, where the periodicity of the "walk" motion is still quite clear. The whole walking process is decomposed into five basic motion units as in Fig. 3(a): the preparing unit (frames 1-24), the initial unit (frames 25-31), the cyclic unit (frames 32-88), the final unit (frames 89-99) and the wind-up unit (frames 100-109). In this case, the cyclic unit includes three walking cycles each of which needs two steps respectively with the left and right legs. A walking cycle C_u starts from frames 39 to 56.

The used running data, composed of 150 frames, are cut off from a long sequence. Although the input running motion is not a complete process, i.e., it does not include all five motion units, the periodic regularity of its cyclic unit is still clear, as shown in Fig. 3(b). The running sequence can be divided into four basic motion units: the preparing, initial, cyclic, and final units.

If some motion is fully understandable and can be decomposed into several different motion units, conversely, directly connecting these motion units can also restore the motion, or even create a new complex motion. For self-connection, a cycle is repeated directly and arbitrarily during the synthesis. For the connection of different motion units, it requires a common posture between two units. The results of motion synthesis are recorded in our supporting video.

5 Conclusion

In this work, we propose the spatio-temporal LTSA (STLTSA) method to analyze and decompose human motion. This method extends the original LTSA

to handling temporal sequences, where the nearest neighborhood is constructed through the temporal consistency. In essence, the STLTSA method is also applicable in behavior classification [12] or motion analysis [13] with video data. Combining video data with motion capture data could effectively improve the performance of human motion tracking and recognition.

Acknowledgement. This work was partially supported by Natural Science Foundation of China Grant 60903120, 863 Project 2009AA01Z429, Shanghai Natural Science Foundation Grant 09ZR1434400, and Innovation Program of Shanghai Municipal Education Commission.

References

1. Jolliffe, I.T.: Principal Components Analysis. Springer (1986)
2. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling, 2nd edn. Chapman and Hall/CRC, Boca Raton (2001)
3. Tenenbaum, J., Silva, V.D., Langford, J.: A global geometric framework for nonlinear dimension reduction. Science 290, 2319–2323 (2000)
4. Roweis, S., Saul, L.: Nonlinear dimension reduction by locally linear embedding. Science 290, 2323–2326 (2000)
5. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimension reduction via local tangent space alignment. SIAM Journal of Scientific Computing 26, 313–338 (2004)
6. Park, M.J., Shin, S.Y.: Example-based motion cloning: Research articles. Comput. Animat. Virtual Worlds 15, 245–257 (2004)
7. Shin, H.J., Lee, J.: Motion synthesis and editing in low-dimensional spaces. Computer Animation and Virtual Worlds 17, 219–227 (2006)
8. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Trans. Pattern Anal. Mach. Intell. 30, 283–298 (2008)
9. Li, R., Tian, T.P., Sclaroff, S.: Divide, conquer and coordinate: Globally coordinated switching linear dynamical system. IEEE Trans. Pattern Anal. Mach. Intell. 34, 654–669 (2012)
10. van der Maaten, L., Postma, E.O., van den Herik, H.J.: Dimensionality reduction: A comparative review (2008)
11. Polito, M., Perona, P.: Grouping and dimensionality reduction by locally linear embedding. In: Neural Information Processing Systems, pp. 1255–1262 (2001)
12. Jenkins, O.C., Mataric, M.J.: Deriving action and behavior primitives from human motion data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2551–2556 (2002)
13. Laptev, I., Belongie, S.J., Perez, P., Wills, J.: Periodic motion detection and segmentation via approximate sequence alignment. In: Proceedings of ICCV 2005 (2005)