Project Description

1 Introduction

Recognizing human action is a key component in many vision applications, such as video surveillance, 3D human pose estimation and video indexing. From the human-centered computing (HCC) point of view, an automatic action recognition system can provide an interface between artificial agents and human users accounting for perception and action in a novel interaction paradigm.

Although significant progress has been made in action recognition [1], the problem remains inherently challenging due to significant intra-class variations, viewpoint change, partial occlusion and background dynamic variations. A key limitation of many action-recognition approaches is that their models are learned from single 2D view video features on individual datasets and thus unable to handle arbitrary view change or scale and background variations. Also, since they are not generalizable across different datasets, retraining is necessary for any new dataset.

Our research is motivated by the requirement of view-invariant action recognition and the fact that the existing human motion capture data provides useful knowledge to understand the intrinsic motion structure (Fig. 2). In particular, we address the problem of modeling and analyzing human motion in the joint-trajectories space. Our view-invariant recognition system has the following functions (Fig. 1).

1. Given a labeled Mocap sequences with $M$ markers in 3D, which is a $3M$-dimensional sequential data, the low dimensional manifold structure (i.e., geodesics distance, intrinsic dimensionality, etc) is learnt by using Tensor Voting. This is an offline process, as shown in Fig. 1.

2. For other unlabeled motion sequences in 3D, after the intrinsic structure learning, we can calculate the motion similarity score with each labeled motion sequence by the proposed spatio-temporal alignment approach, to do action recognition. This is an online process.

3. More interestingly, our system can recognize actions from videos. The 2D tracking results from single view videos are often noisy and have occlusions, while the manifold structure learning algorithms remain the same and our alignment approach can naturally handle those 2D input.

Our system has the following advantages,

**Intra-person variations**, a person repeating an action twice, especially in motion dynamics, can be handled by the proposed temporal alignment method (sec. 5.2.1).

**Inter-person variations**, two people performing an action with differences in both pose style and motion dynamic, are considered by combining temporal and spatial alignment (sec. 5.2.2) together.

**3D and 2D**, our system can naturally associate the Mocap sequences with joint 2D trajectories from video tracking results. We show that the $2M$D joint-trajectories from the projected view is the linear projection of the $3M$D (dynamic) manifold. Additionally, the spatial matching is invariant to the projecting of joint 3D trajectories to arbitrary 2D view.
View Invariant, low dimensional human motion manifold models are learnt from the 3D Mocap data, and our alignment algorithms can handle 2D input (arbitrary viewpoint), thus two features enable our system to recognize actions regardless of the viewpoint, without dataset dependent training.

Occlusion Handling, if we apply the 2D visual tracker, only visible points of the body can be reasonably tracked. Instead of $M$ key points, only $K < M$ points are tracked during the whole action. The recognition procedure is the same as that of complete joint-trajectories, with only one difference that $M - K$ missing key points are not considered in the spatial alignment step.

Furthermore, after estimating actions, for partially occluded 2KD input from videos, the complete skeleton (3MD) can be recovered by using the already learnt manifold from Mocap data. When applying our approach to a video dataset, there is no training process on this dataset and the people in these videos do not necessarily appear in the labeled Mocap sequences. Thus, our approach can be considered as a transfer learning framework, i.e., the knowledge from Mocap data can be adapted to any action video.

Contributions of this proposal includes, (i) the capture of both spatial and dynamic motion properties in the spatio-temporal manifold framework, (ii) the proposal of an efficient alignment algorithm and a robust similarity metric for non-linear multivariate time series, (iii) the ability to handle viewpoint change, and to recognize actions from partially occluded noisy trajectories from videos.

We plan to validate our approach on several human motion datasets, such as the CMU motion capture dataset (Mocap), and Brown HumanEva [2], a benchmark dataset for human pose tracking and action recognition. We have already obtained promising preliminary results.
Professor Medioni is the recipient of the creative work on 3D head pose and facial deformations estimation and facial expressions recognition. Professor Medioni is the inventor of Tensor Voting algorithm [3, 4], which is a robust and efficient computational framework for learning manifold structure in high dimensional data space. Tensor Voting has been successfully applied to many core problems in computer vision and machine learning, i.e., stereo, motion analysis, intrinsic dimensionality estimation, etc. Our spatio-temporal manifold learning method is built on the existing Tensor Voting algorithm, while the temporal dimension is added to better model the sequential high dimensional motion data.

2 Prior NSF Funding

Professor Medioni was the recipient of a number of NSF grants, the latest being “Difficult Problems in Stereo and Motion Analysis”.

3 Related Work

Inspired by the success in object recognitions, low-level features like space-time interest points (STIPs) plus Histogram of Oriented Gradient (HOG) descriptors are used in many action recognition works [1, 5]. Silhouettes based features are also popular [6, 7], for which good results rely on accurate foreground extractions. Some works also use tracked key points, which are quantized as feature vectors by the pre-learned or manually designed codebook [8, 9, 10]. Action recognition is a multifaceted field, our discussion focuses on view-invariant methods, and readers can refer to a recent review [11] for more details.

View Invariant Recognition. Hidden Markov Model (HMM) is built on 3D joint-trajectories (Mo-cap) to capture the dynamic information of human motion [12]. The claimed advantage of the 3D HMM model is that the dependence on view point and illumination is removed. However, HMM requires large amount of training data in relatively high dimensional space (e.g. 67) and the HMM model structure must be adaptively designed for specific application domains. These may be potential factors that make the recognition performance unsatisfactory, and AdaBoost is used to improve the accuracy [12]. View-independence is also address in [7, 13] by rendering Mocap data of various actions from multiple viewpoints, which is a time and storage consuming process. For instance, 10²
interval for camera tilt angle (range $\pi/2$) and pan angle (range $2\pi$) results in 360 render images for each pose, which increases the computational cost for recognition. Another class of methods relies on recovering 3D poses information from silhouettes. In [6], 3D models are projected onto 2D silhouettes with respect to different view point, and [14] detects 2D feature first and then back-projects them to action features based on a 3D visual hull. These methods require a computationally expensive search process over model parameters to find the best match between 2D features and 3D model. Very recently, in [15], a 3D HoG descriptor was proposed to handle view point change, and this approach requires the multiple view camera settings for training data to achieve the view-invariant recognition.

Departing from these methods, our recognition process does not require 2D pose rendering or parameters search. Our trajectory features are located at body skeleton’s key locations, with explicit semantic meaning, allowing our system to be directly applied to arbitrary scene without datasets dependent training\(^1\).

**Dynamic Manifold Model.** Non-linear manifold learning and latent variable modeling (LVM) is prominent in machine learning research in the past decade [16, 17]. In [18], Tensor Voting [4] is used to analyze the 1D manifold of landmark sequences, and the manifold structure is applied to 3D face tracking and expression inference. In particular, some probabilistic latent variable frameworks, i.e., GP-LVM, GPDM and its variants [19, 20, 21], focus on motion capture data and try to capture the intrinsic structure of human motion, which is further applied to 3D monocular people tracking [22]. One advantage of these methods is the ability to model the low-dimensional latent space with associated dynamics based on a few high-dimensional training data. Furthermore, missing values are easily handled thanks to the inherent properties of the probabilistic framework.

While our STM framework is inspired by [18, 22, 20], our goal is significantly different. GPDM methods *explicitly* model the latent human pose space, which is designed for recovering the intrinsic motion structure. By contrast, STM *implicitly* models the latent pose space and focuses on recovering the latent “completion” variable, which is more suitable for motion sequence alignment.

**Motion Sequence Matching.** Given two human motion sequences, an important question is to consider whether those two sequences represent the same motion, similar motions or distinct motions. This can be viewed as a (spatio-temporal) alignment problem, serving as a foundation for action recognition, clustering, etc. Canonical component analysis [23], proposed for learning the shared subspace between two high dimensional features, which been used as the spatial matching algorithm for activity recognition from video [24] and activity correlation from cameras [25]. Video synchronization is addressed as a temporal alignment problem in [26, 27], which uses dynamic time warping (DTW) or its variants [28]. [29] uses optimization methods to maximize a similarity measure of two human action sequences, while the temporal warping is constrained by 1D affine transformation. The same linear temporal model is also used in [30].

Very recently, as the elegant extension of canonical correlation analysis (CCA) and DTW, canonical time warping (CTW) is proposed for spatio-temporal alignment of two multivariate time series and applied to align human motion sequences between two subjects [31]. CTW is formulated as an energy minimization framework and solved by an iterative gradient descent procedure. Since spatial and temporal transformations are coupled together, the objective function becomes non-convex and the solution is not guaranteed to be global optimal. Under the STM model, we propose dynamic manifold warping (DMW), which focuses on time series with intrinsic spatial structure and guarantees global optimal solution.

**Temporal Segmentation.** Some works focus on how to correctly segment motion capture sequences. [32] proposed an on-line algorithm to decompose motion sequences into distinct action unit in the non-smoothing point by (probabilistic) Principal Component Analysis (PCA). Aligned Cluster Analysis (ACA) is developed for temporal clustering of facial behavior with a multi-subject correspondence algorithm for matching facial expressions [33, 34].

\(^1\)It requires videos to have fairly enough resolution to allow key points tracking or human pose estimation algorithms.
4 Proposed Work Overview

Our approach is depicted in Fig. 1. The joint-trajectories of $M$ human body key points are used to present a human motion sequence. These input features can be either provided by Mocap (3D) or be tracked from a single view video (2D). For modules shown in Fig. 1, those hollow ones use existing algorithm, and those solid ones (color blue) use our algorithms proposed in this proposal. We briefly introduce these modules in this section, and more technique details and proposed algorithms are given in section 5.

M.1. Structure Learning. An important assumption we make is that, although human motion data lies in a high dimensional space, the natural property suggests there is a low dimensional intrinsic space embedded in the observations. This has been previously pointed out in [20, 22] for actions such as, walking and running. In particular, we propose spatio-temporal manifold (STM) framework (sec. 5.1), incorporating both spatial and dynamic structures, to model the joint-trajectories. Given a Mocap sequence, which defines a spatio-temporal manifold (STM) in $3MD (2M$ or $2K$ for 2D input or with occlusion), where $M$ is the number of markers, Tensor Voting is used to learn the manifold structure, and voting algorithms are modified for combing the temporal information.

M.2. Spatio-Temporal Alignment. Based on the structure learning results from STM, a motion similarity function (sec. 5.2) is proposed to calculate how actions are similar in two motion sequences, after proper spatial and temporal alignment. There are four proposed algorithms in this module, temporal alignment, spatial alignment, motion similarity metric and 3D body skeleton inference. The first three are designed as unsupervised machine learning algorithms, which can be applied for non-linear multivariate time series in general. To meet the online computational requirements, our algorithms are efficient and easy to implement without explicit parameters tuning. Furthermore, our algorithms are robust to noisy input, i.e., tracking results from videos.

Based on those modules, several applications can be derived naturally.

A.1. Clustering and Recognition from Mocap. By calculating the similarity matrix for $N$ motion sequences and combining with graph based clustering algorithms, we can cluster a number of Mocap sequences into different action classes. Second, recognizing actions from Mocap can be done by associating the test sequences with each labeled sequence and selecting the pair with the maximum similarity. More interestingly, our system can not only recognize actions from 3D input, but also 2D input from arbitrary viewpoint, making view-invariant recognition from single view videos possible.

A.2. Recognition from Videos. To apply our approach for videos, we have a pre-processing step to extract joint 2D trajectories from image observations. The problem itself is challenging and tightly connected to human pose estimation, an important subarea in computer vision. Currently, we apply the incremental learning visual tracker (IVT) [35], an online updated appearance model is used to model the objects dynamic variation. Although tracking results are often noisy, we can still recognize actions from those tracked trajectories, even with occlusion. More state-of-the-art tracking algorithms are considered in our project.

A.3. 3D Skeleton Inference. After estimating the action label for the noisy 2D input from videos (with occlusion, $2KD$), the fully skeleton ($3MD$) can be recovered by associating with the already learnt manifold from the corresponding Mocap motion sequence.

5 Methodology

5.1 Spatio-Temporal Manifold for Human Motion Data

Suppose there is a $d$-dimensional submanifold $2R$ embedded in an ambient space of dimensionality $D \gg d$. We use latent variable model (LVM) to represent $2R$ as a mapping between the intrinsic space and the ambient space: $f : \mathbb{R}^d \rightarrow \mathbb{R}^D$ and $x = f(\tau) + \epsilon$, where $x \in \mathbb{R}^D$ is the observation variable, $\tau \in \mathbb{R}^d$ is the latent variable and $\epsilon \in \mathbb{R}^D$ is the noise. In computer vision applications, often the mapping function $f$ is highly non-linear, and the ambient space is the spatial (feature)
space, so \( M \) is also called spatial-manifold. To incorporate the temporal dimension into the standard LVM, we propose a novel framework as follows.

**Definition:** a spatio-temporal manifold (STM) is a directed traversing path \( \mathcal{M}_p \) (with boundary or compact) on a spatial-manifold \( \mathcal{M} \), and further embedded in \( \mathbb{R}^D \).

A traversing path \( \mathcal{M}_p \) can be intuitively thought as a point walking on \( \mathcal{M} \) from a starting point at time \( t_1 \) (\( \tau_{start}, x_{start} \)) to a ending point (\( \tau_{end}, x_{end} \)) at time \( t_2 \). A path is not just a subset of \( \mathcal{M} \) which ”looks like” a curve, it also includes a natural parametrization as, \( g_{\tau \rightarrow \tau} : [0 1] \rightarrow \mathcal{M} \), s.t. \( g(0) = \tau_{start} \) and \( g(1) = \tau_{end} \). So, a new latent variable \( \zeta \in [0 1] \) is associated with every point in this path. Furthermore, the relationship between \( \zeta \) and temporal index \( t \) can be modeled as a time series \( p_{t \rightarrow \zeta} : [t_1 t_2] \rightarrow [0 1] \), s.t. \( h(t_1) = 0 \) and \( h(t_2) = 1 \). Since \( \mathcal{M} \) is embedded in \( \mathbb{R}^D \) by \( f(\cdot) \), essentially the traversing path (with noise) can be described as a non-linear multivariate time series as \( x(t) = f(g((p(t)))) + \epsilon \). The left part of Fig. 3 is the graphical model of STM, and the right part is a visualization of a traversing path on a 2D sphere embedded in \( \mathbb{R}^3 \).

**STM for Human Motion Data.** Given a length \( L_x \) human action sequence (e.g. stretching), the joint-trajectories can be represented as a matrix \( X_{1:L_x} = [x_1 x_2 ... x_{L_x}] \in \mathbb{R}^{D \times L_x} \), where \( x_t \) is the joint-positions at temporal index \( t \). In 3D (Mocap), \( x_t = [p_{t11}^t p_{t12}^t p_{t13}^t ... p_{tM1}^t p_{tM2}^t p_{tM3}^t]^T \in \mathbb{R}^{3M \times 1} \) and \((p_{i1}^t p_{i2}^t p_{i3}^t)\) is the coordinate of the \( i_{th} \) marker in \( \mathbb{R}^3 \). Or in 2D (tracking trajectories), \( x_t = [u_{11}^t u_{12}^t ... u_{M1}^t u_{M2}^t]^T \in \mathbb{R}^{2M \times 1} \), \((u_{i1}^t u_{i2}^t)\) is the pixel location of the \( i_{th} \) key point. Although \( x_t \) lies in a high dimensional space, the natural property of human pose suggests \( x_t \) having lower intrinsic degree of freedom. So, \( X_{1:L_x} \) is just a sequence of sampled observations on a STM. Here, the ambient space is the joint-position space, manifold \( \mathcal{M} \) is the human pose space, and \( \mathcal{M}_p \) is a specific type of human action. The newly introduced variable \( \zeta \) is assigned to a semantic meaning which indicate the “completion” degree of a action. For a complete action sequence, we assume the starting point of the action has \( \zeta = 0 \) and the ending point has \( \zeta = 1^2 \).

5.1.1 Structure Learning

With the temporal index \( t \), STM is a multivariate time series. Interestingly, without \( t \), the set of all points in a STM is essentially a 1D manifold.

**Proposition 1.** The set of all points on a self-disjoint traversing path \( \mathcal{M}_p \) satisfies a set of regularity conditions on \( d \)-dimensional submanifold \( \mathcal{M} \) (embedded in \( \mathbb{R}^D \)), denoted as \( S(\mathcal{M}_p) \), is a 1-dimensional submanifold in \( \mathbb{R}^D \).  

\(^2\)For periodic motion, i.e, walking, it defines a motion cycle.
The proof of proposition 1 is straightforward. Also empirically, without \( t \), a human motion sequences is found to be a smooth 1D manifold with possible intersection points\(^3\). Given \( \{x_s\}_{s=1}^{L} \) be \( L \) ordered data points sampled from a STM, the goal of learning is to estimate the tangent space and recover the latent “completion” variable \( \zeta_t \) from those samples. Note that our goal is different from most latent variable models, which aim to identify \( \tau \) \([16, 17]\) and sometimes \( f(\cdot) \) \([19, 20, 21]\).

**Learning** \( d_{Geo}(\cdot) \): we use Tensor Voting to calculate the minimum traversing distance between \( x_t \) and \( x_{t+1} \) to approximate the geodesic distance. Tensor Voting is a non-parametric framework propose to estimate the geometric information of manifolds, as well as the intrinsic dimensionality \([4]\). Let \( x^0 = x_s \) and \( x^K = x_{s+1} \), we have

\[
d_{Geo}(x_s, x_{s+1}; \mathcal{M}_p) \approx \sum_{k=0}^{K} \| x^k - x^{k+1} \|_{L_2}
\]

where \( x^{k+1} \) is updated from the current point \( x^k \) (Fig. 5),

\[
x^{k+1} = x^k + \alpha J^*(x^k) J^T(x^k) (x_{s+1} - x^k)
\]

until \( x^{k+1} \) converges to \( x_{s+1} \). \( \alpha \) is a step length, and \( J^*(x^k) \) is the tangent space estimation on \( x^k \) by Tensor Voting. \([18]\) uses Tensor Voting to estimate the manifold structure for 3D face tracking in 126D space, while the temporal index is not explicitly considered. Our algorithm is a revised version of \([18]\) under the STM framework.

**Learning** \( \zeta_t \): a two stage approach is possible, first estimate \( \tau \) (or \( f(\cdot) \)) on a collection of time series, and then optimize \( \{\zeta_{1:L}\} \). While we propose a solution which performs direct estimation for individual sequence based on the learnt geodesic distance.

\[
\zeta_t^* = \frac{\sum_{s=1}^{L-1} d_{Geo}(\tau_s, \tau_{s+1}; \mathcal{M}_p)}{\sum_{s=1}^{L-1} d_{Geo}(\tau_s, \tau_{s+1}; \mathcal{M}_p)}
\]

Since the traversing path is continuous and smooth, the global geodesic distance is approximately decomposed to the sum of the local distance, inspired by ISOMAP \([16]\).

In summary, STM is a extension of the manifold framework by adding the extra temporal dimension to model the sequential data. STM also extends the multivariate time series framework by incorporating the latent variable model in the spatial space.

**Preliminary Results.** CMU Mocap dataset is chosen in our experiments, \( M = 15 \) key points are used to represent the human body, resulting in joint 3D trajectories in 45D space. These 15 key points are selected manually from total 30 markers, which is extracted from the “ame” and “asf” files by our joint-angle to joint-position transfer algorithm. Fig. 4 illustrates 15 markers representation and the latent completion variable \( \zeta \) learning results from a “stretching” sequence in an action unit, i.e., from the action’s start to the end. We uniformly divide the sequence into 5 stages in the time index. The dynamic variations in stage 2 and 4 are larger than the others, and these two stages correspond to “open-up” and “take-back” hands. Stage 3 has the smallest variations, because it corresponds to the “peak” state, i.e., there is almost no hands movement. Stage 1 and 5 correspond to the starting and ending stage.

**5.2 Motion Sequence Matching by Spatio-Temporal Alignment**

The goal of this section is, given two human action sequences \( X_{1:L_x} \in \mathbb{R}^{D_x \times L_x} \) (\( \mathcal{M}_p \)) and \( Y_{1:L_y} \in \mathbb{R}^{D_y \times L_y} \) (\( \mathcal{M}_p \)) , to calculate the motion distance score \( S(X_{1:L_x}, Y_{1:L_y}) \) after proper spatial and temporal alignment. The problem is inherently challenging because of the large spatial/temporal scale difference between human actions, ambiguity between human poses, as well as the inter/intra subject variability \([31]\).

---

\(^3\)Intrinsic dimensionality on those points are large than 1.

\(^4\)Both can be 3D joint-trajectories, or one is 3D and the other is 2D joint-trajectories from the video tracking results.
Figure 4: An illustration of the non-linearity of $\zeta(t)$. Top: action “stretching” (Mocap), 6 samples are uniformly distributed in 368 frames; bottom: estimated latent “completion” variable. The whole action is decomposed into 5 stages.

5.2.1 Temporal Alignment

We model the motion sequence alignment as a spatial-temporal alignment problem under the STM framework, and incorporate manifold learning and temporal alignment together, resulting in dynamic manifold warping (DMW).

Formulation. Given two time series $X_{1:L_x} \in \mathbb{R}^{D_x \times L_x}$ and $Y_{1:L_y} \in \mathbb{R}^{D_y \times L_y}$, find the optimal aligning path $P = [p_1, p_2, \ldots, p_L] \in \mathbb{R}^{2 \times L}$ by minimizing the following loss function,

$$L_{DMW}(F_x(\cdot), F_y(\cdot), W_x, W_y) = \| F_x(X_{1:L_x}) W_x^T - F_y(Y_{1:L_y}) W_y^T \|_P^2$$

where $W_x \in \{0, 1\}^{L_x \times L_x}$, $W_y \in \{0, 1\}^{L_y \times L_y}$ are binary selection matrices encoding the aligning path $P$. $w_{t,t'}^x = w_{t,t'}^y = 1$ is equivalent to $p_t = [t_x, t_y]^T$, which means $x_{t_x}$ corresponds to $y_{t_y}$ at step $t$ in the aligning path. $F(\cdot)$ is spatial transformation function such that $X_{1:L_x}$ and $Y_{1:L_y}$ are mapped to a shared subspace with the same dimensionality. If $F(\cdot)$ is an identity function, then $L_{dmw}$ reduces to $\| X_{1:L_x} W_x^T - Y_{1:L_y} W_y^T \|_P^2$, which is equivalent to performing DTW directly on $X_{1:L_x}$ and $Y_{1:L_y}$.

Unlike the alternative iterative algorithm to optimize $L_{DMW}$, i.e., optimize $W$ with fixed $F$ and then optimize $F$ with fixed $W$, we propose a two-step approach without the iterative computing. Instead of optimizing $F_x, F_y$ in eq. 4, we directly estimate them under the STM framework.

Step 1. Under the STM model in section 5.1, we choose $F(\cdot)$ to be $h^{-1}(\cdot)$, essentially $F_x(X_{1:L_x})$ is $\zeta^T_{1:L_x} \in \mathbb{R}^{1 \times L_x}$, and $F_y(Y_{1:L_y})$ is $\zeta^T_{1:L_y} \in \mathbb{R}^{1 \times L_y}$. $\zeta$ represent the universal structure for all STMs, making aligning two sequences with different actions possible. If the sequence is training data (i.e. Mocap), then eq. 3 can be used, otherwise, the following approach can be used to speed up the process.

For two neighborhood points $x_s, x_{s+1} \in \mathbb{R}^{D_x \times 1}$, Tensor Voting provides the tangent space estimation results on these two points as $J^*(x_s) \in \mathbb{R}^{D_x \times 1}$ and $J^*(x_{s+1}) \in \mathbb{R}^{D_x \times 1}$. The geodesic distance $d_{geo}(x_s, x_{s+1}; \mathcal{M}_{ps})$ can be approximated by choosing an optimal traveling transaction point $x^*_s = x_s + \beta J^*(x_s)$ as follows,

$$\min_{\beta \in \mathbb{R}} \| x_s + \beta J^*(x_s) - x_{s+1} \|_2^2$$

\footnote{For self-disjoint points on the path, with intrinsic dimensionality 1.}
Figure 5: **Learning geodesic distance.** Left: $K$-path method in sec. 5.1; right: a closed-form solution.

Figure 6: **Dynamic manifold warping.** Left: action “stretching” (368 frames, Mocap); middle: action “jogging” (47 frames, HumanEva); right: a $368 \times 47$ aligning matrix. The non-linearity of the aligning path is visualized by the dark blue region (blue indicates the small error, and red indicates the large error).

This is a second order approximation of the minimum traversing path in eq. 1.2. The **variable length** $K$ piecewise directed path $x^0 \rightarrow x^1 \ldots \rightarrow x^K$ ($x^0 = x_s$ and $x^K = x_{s+1}$) is approximated as a **fixed length** 2 directed path $x_s \rightarrow x^*_s \rightarrow x_{s+1}$, as visualized in Fig. 5. The optimal $\beta^*$ has a closed-form solution as $\frac{1}{2}J^*(x_s) = x^*_s$.

The closed-form solution is extremely fast and produce reliable results in our experiments.

**Step 2.** After learning the geodesic distance, combining with eq. 3, we can obtain the estimated results for $\zeta^x_{1:L_x}$ and $\zeta^y_{1:L_y}$, denoted as $\zeta^x \in \mathbb{R}^{1 \times L_x}$ and $\zeta^y \in \mathbb{R}^{1 \times L_y}$. Replace $\mathcal{F}(\cdot)$ with $\zeta^x$ and $\zeta^y$ in eq. 4. $\mathcal{L}_{DMW}$ reduces to the following formulation,

$$\mathcal{L}_{DMW}(W_x, W_y) = \| \zeta^x W_x^T - \zeta^y W_y^T \|^2_2$$

This is equivalent to performing DTW in the transform domain, i.e., $\zeta^x$ and $\zeta^y$. Optimizing eq. 6 results in **variable length** aligning path (vary from $\max(L_x, L_y)$ to $L_x + L_y - 1$), which is not proper for similarity metric. Thus, referenced DTW is proposed to fix the path length by setting one align matrix to be identity,

$$\| \zeta^x I_{L_x} - \zeta^y W_y^T \|^2_F$$

where $I_{L_x}$ is an identity matrix. $X_{1:L_x}$ is chosen as the reference sequence, and $Y_{1:L_y}$ is aligned to $X_{1:L_x}$ by the warping matrix $W_y \in \mathbb{R}^{L_y \times L_x}$. The align path in eq. 7 has the fixed length $L_x$. Since
Figure 7: Temporal Alignment Results. DMW is compared with DTW and CTW. Reference sequence is shown in the first row, followed by aligned results. 3 red arrows indicate 3 key states in the reference sequence, i.e., the peak of the first boxing, back to the natural state and the peak of the second boxing. The aligning sequence also has 3 red arrows, indicating the peak of the first jumping, back to natural state and the peak of the second jumping.

ζx and ζy are monotonically increasing sequences, dynamic programming provides an extremely efficient solution \(O(L_x + L_y)\) as follows,

\[
p(2, t + 1) - p(2, t) = \arg \min_{\delta t \geq 0} \| \xi_{p(1, t)} - \zeta^y_{p(2, t) + \delta t} \|
\]

where \(P \in \mathbb{R}^{2 \times L_x}\) always has \(p(1, t) = t\), and satisfies boundary conditions, that \(p_1 = [1 \ 1]^T\) and \(p_{L_x} = [L_x \ L_y]^T\). \(P\) can be by recursively optimized by eq. 8 from the starting point \([1 \ 1]^T\).

**Preliminary Results.** The proposed dynamic manifold warping (DMW) algorithm (eq. 4) is compared with other state-of-the-art algorithms. In particular, dynamic time warping (DTW) [26] is chosen as the baseline algorithm and canonical time warping (CTW) [31] is chosen as the alternative method. Several Mocap sequences are selected for the experiments, and the alignment ground-truth is manually provided. To make the comparison more clearer, the sequences may include more than one action unit. Fig. 7 shows the visual comparison for two motion sequences, one is boxing (twice) and the other is side jumping (twice). DTW does not consider the spatial transformation between two time series, making it difficult to align two motion sequences of by two people, as shown in the second row. CTW significantly outperforms DTW, while still has improvement space. Our DMW algorithm gets the best results among three methods.

For future work, we plan to include more motion sequences and manually label the alignment ground-truth, to provide quantitative performance comparison with other state-of-the-art algorithms.

---

*For recognition, all sequences only include one action unit.*
5.2.2 Temporally Local Spatial Alignment

The following framework is proposed,

\[ \| V_x \mathcal{U}(X_{t_1:t_2}) - V_y \mathcal{U}(Y_{t_1:t_2}) \|_F^2 \]  \hspace{1cm} (9)

where \( X_{t_1:t_2} \in \mathbb{R}^{D_x \times (t_2-t_1+1)} \) are consecutive frame features \( x_{t_1} \) to \( x_{t_2} \) in the reference sequence, and \( Y_{t_1:t_2} \in \mathbb{R}^{D_y \times (t_2-t_1+1)} \) are temporally corresponding samples in the aligned sequence \( Y_{1:L_y} \).

\( \mathcal{U} \) is the feature extraction function and \( V_x \) is the spatial transformation function (same for \( V_y \)). The spatial matching is restricted on local segments, since the global matching on the entire sequences is often not accurate due to the non-linear variation.

Denoting the extracted features by \( \mathcal{U}(\cdot) \) as two zero mean feature sets, \( U_x \in \mathbb{R}^{d_1 \times n} \) and \( U_y \in \mathbb{R}^{d_2 \times n} \), we consider an unsupervised learning approach; i.e., canonical correlation analysis (CCA), in which a pair of linear transformations is optimized in the sense of maximizing the correlation in transformed features as follows,

\[
E(V_x, V_y) = Tr(V_x^T U_x (V_y^T U_y)^T)
\]

s.t., \( V_x^T U_x U_x^T V_x = V_y^T U_y U_y^T V_y = I_d \)

Where \( V_x \in \mathbb{R}^{d_1 \times d} \) and \( V_y \in \mathbb{R}^{d_2 \times d} \) are two linear projection matrix for \( U_x \) and \( U_y \), and \( I_d \) is the identity matrix of size \( d \times d \). Minimizing this objective function is equivalent to solving a generalized eigenvalue problem. The metric can be induced in the transform domain as,

\[
D_{CCA}(U_x, U_y) = \| V_x^* U_x - V_y^* U_y \|_F^2
\]  \hspace{1cm} (11)

\( V_x^* \in \mathbb{R}^{d_1 \times d} \) and \( V_y^* \in \mathbb{R}^{d_2 \times d} \) are the solutions of eq. 10.

5.2.3 Motion Distance Score

Based on the proposed DMW for temporal matching and CCA for spatial matching, we further propose two types of motion similarity score function by choosing two feature extraction functions \( \mathcal{U}(\cdot) \). In particular, instead of treating \( x_t \in \mathbb{R}^{D_x \times 1} \) (or \( y_t \)) as a high dimensional vector, the implicit structure in the joint-position space is considered. In sec. 5.1, \( x_t = [p_{t11} \ p_{t12} \ p_{t13} \ ... \ p_{tM1} \ p_{tM2} \ p_{tM3}]^T \in \mathbb{R}^{3M \times 1} \), the 3D Euclidean space is implicitly embedded in the joint-position space \( \mathbb{R}^{3M} \). Thus, we reformulate \( x_t \) as,

\[
\mathbf{x}_t = \left( \begin{array}{cccc}
p_{t11} & \ldots & p_{tM1} \\
p_{t12} & \ldots & p_{tM2} \\
p_{t13} & \ldots & p_{tM3} \\
\end{array} \right) \in \mathbb{R}^{3 \times M}
\]  \hspace{1cm} (12)

which turns to be \( M \) samples in \( \mathbb{R}^3 \) (similar operation for \( y_t \in \mathbb{R}^{2M} \) to \( \in \mathbb{R}^{2 \times M} \)). This operation is defined as \( T^3 : \mathbb{R}^{3M} \rightarrow \mathbb{R}^{1 \times M} \), or \( T^2 : \mathbb{R}^{2M} \rightarrow \mathbb{R}^{2 \times M} \).

The first feature extraction function is chosen as \( \mathcal{U}_1(x_t) = T(x_t) \), which is the static pose feature (joint-position in the matrix formulation). The second one is \( \mathcal{U}_2(x_t, x_{t+1}) = T(x_t) - T(x_{t+1}) \).
<table>
<thead>
<tr>
<th>Methods / Data</th>
<th>Mocap(S)</th>
<th>Mocap(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D(combine)</td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td>3D(Static)</td>
<td>85%</td>
<td>83%</td>
</tr>
<tr>
<td>3D(Motion)</td>
<td>60%</td>
<td>67%</td>
</tr>
<tr>
<td>2D(Combine)</td>
<td>90%</td>
<td>87%</td>
</tr>
<tr>
<td>Lv&amp;Nevatia [12]</td>
<td>NA</td>
<td>92%</td>
</tr>
</tbody>
</table>

Table 1: Recognition Performance Rates. (S) means the rate is measured by # of sequences and (F) means the rate is measured by # of frames. All recognitions are done at the sequence level.

which is the motion pose feature between two consecutive frames. Thus, the final similarity score $S_1(X_{1:L_x}, Y_{1:L_y})$ given by the static features is as follows,

$$S_1 = \sum_{t=1}^{L_x} D_{CCA}(T(x_t), T(\tilde{y}_t))$$

where $\tilde{y}_t$ is the temporally corresponding frame estimated by eq. 4. The similarity score $S_2(X_{1:L_x}, Y_{1:L_y})$ given by the motion features is as follows,

$$\sum_{t=1}^{L_x} D_{CCA}(T(x_t) - T(x_{t+1}), T(\tilde{y}_t) - T(\tilde{y}_{t+1}))$$

These two scores can be linearly combined,

$$S(X_{1:L_x}, Y_{1:L_y}) = \lambda S_1 + (1 - \lambda) S_2$$

where $\lambda \in [0, 1]$ can be either optimized by cross-validation in the supervised setting (i.e., recognition), or chosen manually in the unsupervised setting (i.e., clustering). Eq. 15 is a summary result of eq. 3 4 10 and two feature extraction functions. The similarity metric is not symmetric, so we fix the testing sequence to be the reference sequence.

5.3 Results of Matching

Based on the proposed matching algorithm, action recognition is derived naturally. Assume there are number of labeled motion sequences $\{X_{mocap}^i\}_{i=1}^N$ associated with action label $I_i \in I$, where $I = 1, 2, \ldots, C$ indicates $C$ action classes. Given a joint trajectories $Y$ from another Mocap sequence or a video clip by the tracker, the estimated action label $I_y$ is given by

$$I_y = \arg \min_{i \in \{1, 2, \ldots, N\}} S(Y, \{X_{mocap}^i, I_i\})$$

5.3.1 Recognition and Clustering from Mocap

We collected 3978 frames from CMU Mocap performed by fifteen people, performing 10 natural actions (details in Fig. 9). For action clustering, motion distance scores between any two sequences is calculated, resulting in a $10 \times 10$ average motion distance matrix $S$ for these 10 actions (Fig. 9). It is clear that the diagonal area has the smallest variations, which shows the effectiveness our similarity function eq. 15. By applying graph based clustering algorithm on the similarity matrix of all sequences, we can get clustering results. For action recognition, we use the leave-one-out procedure for each sequence, i.e., each sequence is treated as unlabeled and associated with all other sequences. Since each person only performs a specific action once, the recognition process can not benefit from the fact that the same person repeating the same action results in quite large similarity. $\lambda$ in eq. 15 is set to be 0.5 and results (Table 1) show that our approach only misclassifies 5% sequences, or 1.2% by weighing with the number of frames. To demonstrate the superiority of combining both static (eq. 13) and motion features (eq. 14), recognition rates for individual methods are reported in Table 1. The recognition rate drops down to 85% for static features only and 60% for motion features only.
Furthermore, to demonstrate the ability of recognizing actions from arbitrary 2D view, Mocap sequences are projected to joint 2D trajectories in 30D space by a synthetic affine camera. We achieve 90% accuracy in this 2D view recognition. [12] also reports recognition rate for Mocap data, but direct comparison is difficult, because [12] collects a large number of sequences (not only from CMU MOcap) and uses Adaboost for training. Our approach only requires a few sequences for each action. Moreover, 2D view recognition is not considered in [12].

For future work, we plan to include more sequences from CMU Mocap or other motion capture dataset. And the number of action classes is going to be increased to 15 to 20, including almost all important primitive human actions. In terms of 2D action recognition, we intend to project 3D motion data to all kinds of 2D view (from top to bottom, left to right, front to back), and systematically measure the recognition performance in all cases. Furthermore, the markers corresponding to the visually occlusion part, i.e., the right part of human markers under the left-side view, will be considered as missing during the recognition process. Such experiments can not only demonstrate the capability of view-invariant recognition of our approach, but can also serve as a foundation for action recognition from video tracking results (noisy 2D input with occlusion).

5.3.2 Action Recognition from Videos

For videos, often $K = 7$ or 8 key points are estimated from the side view, resulting in joint 2D trajectories in 14D or 16D space. The preliminary results are reported on Brown HumanEva dataset.
To validate our approach on video-based action recognition, we choose a number of video sequences from Brown HumanEva dataset (1 and 2), which is a benchmark proposed for human motion analysis [2] (Fig. 10). The reason that standard action benchmark datasets, like KTH [37] or Weizmann [38] are not used, is because the low resolution of videos makes key points tracking results unreliable. For the IVT tracker [35], we manually label the key points at the first frame and we do not tune any parameter in the tracking process. [39] reports the highest action recognition performance on HumanEva-1, and a latent pose estimator is proposed to improve the recognition process. Contrast to [39], our STM framework focuses on learning the motion “completion” degree and implicitly model the latent human pose. Departing from [39], our approach does not need any training process in HumanEva, all sequences selected from HumanEva-1 or 2 are treated as unlabeled data, and the recognition is automatically done when tracking results are given.

For future work, we will select more sequences from the HumanEva dataset across different views (4 views in total), quantitatively measure the recognition rate, and compare with other state-of-the-art methods on HumanEva. Besides, other interesting action recognition datasets suitable for view-invariant action recognition, like IXMAS or UCF, will also be considered for validating our approach. Furthermore, the natural human motion sequences are continuous and possibly include multiple “action unit”, thus temporal segmentation can be viewed as a pre-processing step for video action recognition. We intend to implement the propose TTV algorithm (Sec. 5.4) to automatically segment input video sequences into pre-defined action units, and quantitatively measure the performance as well as the impact to action recognition.

5.4 Temporal Segmentation

Temporal segmentation of human motion behavior is a crucial step for building the computational framework to analyze human motion. In the supervised training stage, automatically segmenting the raw motion sequences into the pre-defined “action unit” can significantly reduce the efforts of manual labeling. In the testing stage, although the action recognition can be done at the single frame level, most of the existing systems work at the sequence level, relying on the accurate motion sequence segmentation.
In this proposal, we propose an on-line fashion based motion sequences segmentation algorithm, the key part of which is temporal-Tensor Voting (TTV), a new version of Tensor Voting (TV) designed for processing continuous sequential data, i.e. multi-dimensional time series. We use TTV to measure the discontinuity score at every frame, and perform sequence clustering based on this score function. Temporal-Tensor Voting (TTV) follows the same framework of Tensor Voting, and the main difference is, the voting neighborhood is defined in the temporal dimension, not in the spatial dimension. It is notable that, previous works have used TV to process sequential data, in which the time index is added as another dimension [36], while the special properties of the temporal dimension is not reflected. Formally, given any high dimensional sequential sequence \( X_{1:L} \in \mathbb{R}^{D \times L} \), TTV tries to estimate a generic tensor for each point \( x_{t_1} \), which encodes the local spatial-temporal information at the current time \( t_1 \). Formally, given initial tensors \( T^0_{t_2} \in \mathbb{R}^{D \times D} \) at point \( x_{t_2} \), TTV calculate as follows,

\[
T_{t_1} = \sum_{t_2 \in \{t_1 - \delta t, t_1 + \delta t\}} Vote_{\sigma}(x_{t_2}, T^0_{t_2}; x_{t_1}) \tag{17}
\]

where \( \sigma \) is the voting bandwidth, votes only happens in the temporal neighborhood region \( \{t_1 - \delta t, t_1 + \delta t\} \), which is differ from the spatial neighborhood region in TV. After calculating eq. 17, a smooth index \( Q(x_t) \) is calculated based on \( T_t \), to measure the intrinsic motion variation at time \( t \), the large value of \( Q(x_t) \) indicates the motion transaction happens. Temporal segmentation can be done by dividing \( X_{1:L} \) into sub-sequences at time slot with the local largest \( Q(x_t) \).

6 Broader Impact

The potential impact of the proposed research covers multi-aspects, academic research, university education and the society.

6.1 Impact on the Computer Vision and Machine Learning Communities

This proposal addresses view-invariant action recognition by learning the manifold model from Mocap data, which can recognize action from arbitrary viewpoint, even from the top, without dataset dependent training. We intend to propose a number of unsupervised machine learning algorithms for spatio-temporal manifold learning, non-linear time series matching and 3D motion data recovering, which can improve the state-of-the-art performance in the specific application domain.

Driven by this proposal, many important human related computer vision topics are combined together, including action recognition, human pose estimation/tracking, realistic motion synthesis, etc. The center of all these topics is the low-dimensional manifold structure learnt from human motion capture data. By connecting those fields, many potential algorithms can be naturally derived, i.e., action recognition assisted human tracking, human pose estimation based action recognition, manifold learning based body skeleton recovering, etc.

We intend to submit our methodology and results to conferences and journals for review and publication, as our track record of extensive participation and involvement in the organization of conferences indicates.

6.2 Impact at USC and the community at large

The proposed research deals with difficult computer vision problems. As soon as it becomes mature, we intend to introduce some topics into the curriculum of the “Computer Vision”(CSCI-574) and “Advanced Topics in Computer Vision”(CSCI-674 a and b) courses which are already offered in the Computer Science and Electrical Engineering Department of the University of Southern California.

7 Summary

Ambitious research agenda.
Within the scope of the view-invariant action recognition
Propose a number of algorithm which improve the state-of-the-art
Significant academic and social impact
References


