Generalized Canonical Time Warping

Feng Zhou and Fernando De la Torre

Abstract—Temporal alignment of human motion has been of recent interest due to its applications in animation, tele-rehabilitation and activity recognition. This paper presents generalized canonical time warping (GCTW), an extension of dynamic time warping (DTW) and canonical correlation analysis (CCA) for temporally aligning multi-modal sequences from multiple subjects performing similar activities. GCTW extends previous work on DTW and CCA in several ways: (1) it combines CCA with DTW to align multi-modal data (e.g., video and motion capture data); (2) it extends DTW by using a linear combination of monotonic functions to represent the warping path, providing a more flexible temporal warp. Unlike exact DTW, which has quadratic complexity, we propose a linear time algorithm to minimize GCTW. (3) GCTW allows simultaneous alignment of multiple sequences. Experimental results on aligning multi-modal data, facial expressions, motion capture data and video illustrate the benefits of GCTW. The code is available at `http://humansensing.cs.cmu.edu/ctw`.

Index Terms—Multi-modal sequence alignment, canonical correlation analysis, dynamic time warping

1 Introduction

Temporal alignment of multiple time series is an important problem with applications in many areas such as speech recognition [1], computer graphics [2], computer vision [3], and bio-informatics [4]. In particular, alignment of human motion from sensory data has recently received increasing attention in computer vision and computer graphics to solve problems such as curve matching [5], temporal clustering [6], tele-rehabilitation [7], activity recognition [8] and motion synthesis [9], [10]. While algorithms for aligning time series have been commonly used in computer vision and computer graphics, a relatively unexplored problem has been the alignment of multi-dimensional and multi-modal time series that encode human motion. Fig. 1 illustrates the main problem addressed by this paper: how can we efficiently find the temporal correspondence between (1) frames of a video, (2) samples of motion capture data and (3) samples of three-axis accelerometers of different subjects performing a similar action?

The alignment of multi-dimensional, multi-modal time series of human motion poses several challenges. First, there is typically a large variation in the subjects' physical characteristics, motion style and speed performing the activity. Second, large changes in view point also complicate the correspondence problem [11]. Third, it is unclear how existing techniques can be used to align sets of time series that have different modalities (e.g., video and motion capture data). While there is extensive literature on time series alignment (e.g., [12]), standard extensions of dynamic time warping (DTW) or Bayesian networks are not capable of aligning align multi-modal data.

To address these problems, this paper proposes generalized canonical time warping (GCTW), a technique to temporally align multi-modal time series of different subjects performing similar activities. GCTW is a spatio-temporal alignment method that temporally aligns two or more multi-dimensional and multi-modal time series by maximizing the correlation across them. GCTW can be seen as an extension of DTW or canonical correlation analysis (CCA). To accommodate for subject variability and take into account the difference in the dimensionality of the signals, GCTW uses CCA as a measure of spatial correlation. GCTW extends DTW by incorporating a feature weighting mechanism to align signals of different dimensionality and provide higher weights to the features that make both signals more correlated. It also extends DTW by parameterizing the warping path as a combination of monotonic functions, providing more accurate alignment and faster optimization strategies. Unlike exact approaches based on DTW, which have quadratic cost, GCTW uses a Gauss-Newton (GN) algorithm that has linear complexity in the length of the sequence. Preliminary versions of this paper were published in [13], [14].

2 Previous Work

2.1 Temporal Alignment

This section discusses prior work on alignment of time series in the context of computer graphics, computer vision and data mining.

In the literature of computer graphics, temporal alignment of human motion has been commonly applied to solve problems such as content modeling [15], and motion blending [2]. Hsu et al. [9] proposed iterative motion warping (IMW), a robust method that finds a spatio-temporal warping between two instances of motion captured data. Shapiro et al. [16] used independent component analysis to separate motion data

• *The authors are with the Robotics Institute, Carnegie Mellon University. E-mail: zhfe99@gmail.com, ftorre@cs.cmu.edu.*

Manuscript received 15 Dec. 2013; revised 13 Feb. 2015; accepted 10 Mar. 2015. Date of publication 17 Mar. 2015; date of current version 13 Jan. 2016. Recommended for acceptance by C.H. Lampert.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2414429

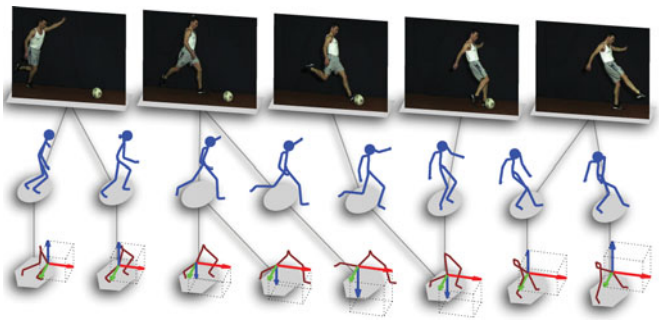

Fig. 1. Temporal alignment of sequences recorded with different sensors (from top to bottom: video, motion capture and accelerometers) of three subjects kicking a ball.

into visually meaningful components called style components. In comparison, our method solves a more general problem of aligning human motion from multi-modal time series.

In the context of computer vision, temporal alignment of video captured with different cameras and view points has been a topic of interest. Rao et al. [17] aligned trajectories of two moving points using constraints from the fundamental matrix. Junejo et al. [8] adopted DTW for synchronizing human actions with view changes based on a view-invariant description. In comparison, our method simultaneously estimates the optimal spatial transformation and temporal correspondence to align video sequences. Recently, Gong and Medioni [18] extended CTW approach to incorporate more complex spatial transformations through manifold learning. Nicolaou et al. [19] proposed a probabilistic extension of CTW for fusing multiple continuous expert annotations in tasks related to affective behavior. These works show the flexibility of our framework for aligning various types of time series.

In the field of data mining, there have been several extensions of DTW to align time series that differ in the temporal and spatial domain. Keogh and Pazzani [20], for example, used derivatives of the original signal to improve alignment with DTW. Listgarten et al. [21] proposed continuous profile models, a probabilistic method for simultaneously aligning and normalizing sets of time series in bio-informatics. Unlike these works, which were originally designed for aligning 1-D time series, our work addresses the more challenging problem of aligning multi-modal and multi-dimensional time series.

In the literature of manifold alignment, Ham et al. [22] aligned manifolds of images in a semi-supervised manner. The prior knowledge of pairwise correspondences between two sets was used to guide the graph embedding. Wang and Mahadevan [23] aligned manifolds based on an extension of the Procrustes analysis (PA). The main benefit of this approach is that PA learns a mapping generalized for out-of-sample cases. However, these models lack a mechanism to enforce temporal continuity.

2.2 Canonical Correlation Analysis (CCA)

CCA [24] is a technique to extract common features from a pair of multi-variate data. Given two sets of n variables (see footnote for the notation[1]), $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d_x \times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{d_y \times n}$, CCA finds the linear combinations of the variables in $\mathbf{X}$ that are most correlated with the linear combinations of the variables in $\mathbf{Y}$. Assuming zero-mean data ($\sum_i \mathbf{x}_i = \sum_j \mathbf{y}_j = 0$), CCA finds a combination of the original features that minimizes the sum of the distances between samples:

$$\min_{\{\mathbf{V}_x, \mathbf{V}_y\} \in \Phi} J_{cca} = \left\| \mathbf{V}_x^T \mathbf{X} - \mathbf{V}_y^T \mathbf{Y} \right\|_F^2 + \phi(\mathbf{V}_x) + \phi(\mathbf{V}_y), \quad (1)$$

where $\mathbf{V}_x \in \mathbb{R}^{d_x \times d}$ and $\mathbf{V}_y \in \mathbb{R}^{d_y \times d}$ denote the low-dimensional embeddings ($d \leq \min(d_x, d_y)$) for $\mathbf{X}$ and $\mathbf{Y}$ respectively. $\phi(\cdot)$ is a regularization function that penalizes the high-frequency components of the embedding matrices:

$$\phi(\mathbf{V}) = \frac{\lambda}{1-\lambda} \|\mathbf{V}\|_F^2, \quad (2)$$

In order to avoid the trivial solution of $\mathbf{V}_x^T \mathbf{X}$ and $\mathbf{V}_y^T \mathbf{Y}$ being zero, CCA decorrelates the canonical variates (columns of $\mathbf{V}_x^T \mathbf{X}$ and $\mathbf{V}_y^T \mathbf{Y}$) by imposing the orthogonal constraint as:

$$\Phi = \Big\{ \{\mathbf{V}_x, \mathbf{V}_y\} \;\Big|\; \mathbf{V}_x^T \Big((1-\lambda)\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I} \Big) \mathbf{V}_x = \mathbf{I}, \; \mathbf{V}_y^T \Big((1-\lambda)\mathbf{Y}\mathbf{Y}^T + \lambda \mathbf{I} \Big) \mathbf{V}_y = \mathbf{I} \Big\}. \quad (3)$$

where $\lambda \in [0, 1]$ is a weight to trade-off between the least-squared error and the regularization terms. In the case of insufficient samples where the covariance matrices ($\mathbf{X}\mathbf{X}^T$ or $\mathbf{Y}\mathbf{Y}^T$) are singular, a positive regularization term $\lambda \mathbf{I}$ is necessary to avoid over-fitting and for numerical stability. Optimizing (1) has a closed-form solution in terms of a generalized eigenvalue problem, i.e., $[\mathbf{V}_x; \mathbf{V}_y] = \mathrm{eig}_d(\mathbf{A}, \mathbf{B})$, where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & \mathbf{0} \end{bmatrix}, \qquad \mathbf{B} = (1-\lambda) \begin{bmatrix} \mathbf{X}\mathbf{X}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}\mathbf{Y}^T \end{bmatrix} + \lambda \mathbf{I}.$$

See [25] for a unification of several component analysis methods and a review of numerical techniques to efficiently solve generalized eigenvalue problems.

In computer vision, CCA has been used for matching sets of images in problems such as activity recognition from video [26] and activity correlation from cameras [27]. Recently, Fisher et al. [28] proposed an extension of CCA with parameterized warping functions to align protein expressions. The learned warping function is a linear combination of hyperbolic tangent functions with non-negative

1. Bold capital letters denote a matrix $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. $\mathbf{x}_i$ and $\mathbf{x}^{(i)}$ represent the ith column and ith row of the matrix $\mathbf{X}$ respectively. x_{ij} denotes the scalar in the ith row and jth column of the matrix $\mathbf{X}$. All non-bold letters represent scalars. $\mathbf{1}_{m \times n}, \mathbf{0}_{m \times n} \in \mathbb{R}^{m \times n}$ are matrices of ones and zeros. $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix. $\|\mathbf{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}$ denotes the p-norm. $\|\mathbf{X}\|_F = \sqrt{\sum_{ij} x_{ij}^2}$ represents the Frobenious norm. $\mathrm{vec}(\mathbf{X})$ denotes the vectorization of matrix $\mathbf{X}$. $\mathbf{X} \circ \mathbf{Y}$ is the Hadamard product. $[\mathbf{X}_1; \cdots; \mathbf{X}_n]$ denotes the vertical concatenation of sub-matrices $\mathbf{X}_i$. $\mathrm{eig}_d(\mathbf{A}, \mathbf{B})$ denotes the top d eigenvectors $\mathbf{V}$ that solve the generalized eigenvalue problem $\mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{V}\Lambda$.

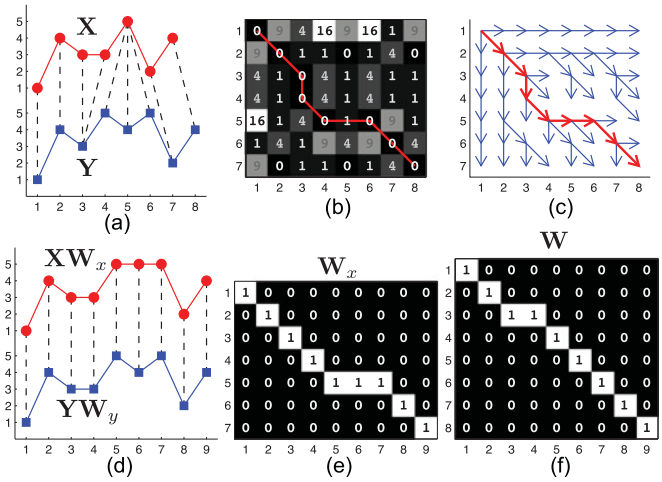

Fig. 2. An example of DTW for aligning time series. (a) Two 1-D time series ($n_x = 7$ and $n_y = 8$) and the optimal alignment between samples computed by DTW. (b) Euclidean distances between samples, where the red curve denotes the optimal warping path ($l = 9$). (c) DP policy at each pair of samples, where the three arrow directions, $\downarrow, \searrow, \rightarrow$, denote the policy, $\pi(\cdot,\cdot) \in \{[1,0],[1,1],[0,1]\}$, respectively. (d) A matrix-form interpretation of DTW as stretching the two time series in matrix products. (e) Warping matrix $\mathbf{W}_x$. (f) Warping matrix $\mathbf{W}_y$.

coefficients, ensuring monotonicity. Similarly, Shariat and Pavlovic [29] imposed monotonic constraints on CCA using non-negative least squares for activity recognition tasks. However, these methods were unable to deal with feature weighting.

2.3 Dynamic Time Warping (DTW)

Given two time series, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}] \in \mathbb{R}^{d \times n_x}$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{n_y}] \in \mathbb{R}^{d \times n_y}$, DTW [1] aligns $\mathbf{X}$ and $\mathbf{Y}$ such that the sum of the distances between the aligned samples is minimized:

$$\min_{\{\mathbf{p}_x, \mathbf{p}_y\} \in \Psi} \; J_{dtw} = \sum_{t=1}^{l} \|\mathbf{x}_{p_t^x} - \mathbf{y}_{p_t^y}\|_2^2, \tag{4}$$

where $l \geq \max(n_x, n_y)$ is the number of indices used to align the samples. The optimal l is automatically selected by the DTW algorithm. The warping paths, $\mathbf{p}_x \in \{1 : n_x\}^l$ and $\mathbf{p}_y \in \{1 : n_y\}^l$, denote the composition of alignment in frames. The ith frame in $\mathbf{X}$ and the jth frame in $\mathbf{Y}$ are aligned if there exist $p_t^x = i$ and $p_t^y = j$ at some step t.

In order to find a polynomial time solution, the warping paths must satisfy three constraints:

$$\begin{aligned} \Psi = \Big\{ \{\mathbf{p}_x, \mathbf{p}_y\} \;\Big|\; & \mathbf{p}_x \in \{1 : n_x\}^l \quad \text{and} \quad \mathbf{p}_y \in \{1 : n_y\}^l, \\ \textbf{Boundary}: \;& [p_1^x, p_1^y] = [1,1] \quad \text{and} \quad [p_l^x, p_l^y] = [n_x, n_y], \\ \textbf{Monotonicity}: \;& t_1 \geq t_2 \Rightarrow p_{t_1}^x \geq p_{t_2}^x \quad \text{and} \quad p_{t_1}^y \geq p_{t_2}^y, \\ \textbf{Continuity}: \;& [p_t^x, p_t^y] - [p_{t-1}^x, p_{t-1}^y] \in \{[0,1],[1,0],[1,1]\} \Big\}. \end{aligned} \tag{5}$$

The choice of step size in the continuity constraint is not unique. For instance, replacing the step size by $\{[2,1],[1,2],[1,1]\}$ can avoid the degenerate case in which a single frame of one sequence is assigned to many consecutive frames in the other sequence. See [1] for an extensive review on several DTW's modifications to control the warping paths.

Although the number of possible ways to align $\mathbf{X}$ and $\mathbf{Y}$ is exponential in n_x and n_y, dynamic programming (DP) [30] offers an efficient approach with complexity of $O(n_x n_y)$ to minimize J_{dtw} using Bellman's equation:

$$J^*(p_t^x, p_t^y) = \min_{\pi(p_t^x, p_t^y)} \|\mathbf{x}_{p_t^x} - \mathbf{y}_{p_t^y}\|_2^2 + J^*(p_{t+1}^x, p_{t+1}^y),$$

where the cost-to-go value function, $J^*(p_t^x, p_t^y)$, represents the cost remaining starting at the tth step using the optimal policy π^*. The policy function, $\pi(\cdot,\cdot) : \{1 : n_x\} \times \{1 : n_y\} \rightarrow \{[1,0],[0,1],[1,1]\}$, defines the deterministic transition between consecutive steps, $[p_{t+1}^x, p_{t+1}^y] = [p_t^x, p_t^y] + \pi(p_t^x, p_t^y)$. Once the policy queue is known, the alignment steps can be recursively selected by backtracking, $p_l^x = n_x$ and $p_l^y = n_y$.

Fig. 2a shows an example of DTW for aligning two 1-D time series. Fig. 2b illustrates the Euclidean distance of each pair of samples. To compute the optimal warping path, DP efficiently enumerates all possible steps as in Fig. 2c from the upper-left corner to the bottom-right one. At the end, the optimal alignment (red curve) can be computed by iteratively tracing back along the arrows.

Given two sequences of length n_x and n_y, exact DTW has a computational cost in space and time of $O(n_x n_y)$. In practice, various modifications [1] on the step size, local weights and global constraints (e.g., the Sakoe-Chiba and Itakura Parallelogram bands [1]) have been proposed to speed up the DTW computation as well as to better control the possible routes of the warping paths. In recent work [31], [32], [33], a multi-scale searching scheme has been shown to effectively generate a speedup from one to three orders of magnitude, compared to the classic DTW algorithm. More recently, Rakthanmanon et al. [34] have shown that DTW for mining 1-D sub-sequences can be scaled up to very large datasets using early-abandoning and cascading lower bounds. However, most of these works are originally designed for 1-D time series. In comparison, our method can be applied to deal with more general multi-dimensional sequences and align signals of different dimensionality.

3 Canonical Time Warping (CTW)

DTW lacks a feature weighting mechanism and thus it cannot be directly used for aligning multi-modal sequences (e.g., video and motion capture) with different features. To address this issue, this section presents CTW, a unified framework that combines DTW with CCA.

3.1 Least-Squares Formulation for DTW

In order to have a compact and compressible energy function for CTW, it is important to note that the original objective of DTW (4) can be reformulated in matrix form as:

$$\min_{\{\mathbf{p}_x, \mathbf{p}_y\} \in \Psi} \; J_{dtw} = \|\mathbf{X}\mathbf{W}_x - \mathbf{Y}\mathbf{W}_y\|_F^2, \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{d \times n_x}$ and $\mathbf{Y} \in \mathbb{R}^{d \times n_y}$ denote the two time series to be aligned. $\mathbf{W}_x = \mathbf{W}(\mathbf{p}_x) \in \{0,1\}^{n_x \times l}$ and $\mathbf{W}_y = \mathbf{W}(\mathbf{p}_y) \in \{0,1\}^{n_y \times l}$ are two binary warping matrices (Figs. 2e and 2f) associated with the warping paths by a non-linear mapping,

$$\mathbf{W}(\mathbf{p}) : \{1:n\}^l \rightarrow \{0,1\}^{n \times l}, \tag{7}$$

The value of the elements of $\mathbf{W}$ is always 0 or 1. $w_{p_t,t}$ is 1 for any step $t \in \{1:l\}$, and zero otherwise. The matrix $\mathbf{W}_x\mathbf{X}$ has the effect to replicate (possibly multiple times) samples of the $\mathbf{X}$. Similarly for $\mathbf{W}_y$. Fig. 2d shows that the DTW alignment in Fig. 2a can be equivalently interpreted as stretching the two time series $\mathbf{X}$ and $\mathbf{Y}$ by multiplying them with the warping matrices $\mathbf{W}_x$ and $\mathbf{W}_y$, respectively as shown in Figs. 2e and 2f. Note that (6) is very similar to CCA's objective (1). CCA applies a linear transformation to combine the rows (features), while DTW applies binary transformations to replicate the columns (time). It is important to notice that reformulating DTW as a least-squared optimization function allows easy generalizations, and this is one of the important contributions of this paper.

3.2 Objective Function of CTW

In order to accommodate for differences in style and subject variability, add a feature selection mechanism, and reduce the dimensionality of the signals, CTW adds a linear transformation ($\mathbf{V}_x \in \mathbb{R}^{d_x \times d}$ and $\mathbf{V}_y \in \mathbb{R}^{d_y \times d}$) to the least-squared form of DTW (6), as in the case of CCA. Moreover, this transformation allows alignment of temporal signals with different dimensionality (e.g., video and motion capture). In a nutshell, CTW combines DTW and CCA by minimizing:

$$\begin{aligned}\min_{\{\mathbf{V}_x,\mathbf{V}_y\}\in\Phi,\{\mathbf{p}_x,\mathbf{p}_y\}\in\Psi} J_{ctw} = &\left\|\mathbf{V}_x^T\mathbf{X}\mathbf{W}_x - \mathbf{V}_y^T\mathbf{Y}\mathbf{W}_y\right\|_F^2 \\ &+ \phi(\mathbf{V}_x) + \phi(\mathbf{V}_y),\end{aligned} \tag{8}$$

where $\mathbf{V}_x \in \mathbb{R}^{d_x \times d}$ and $\mathbf{V}_y \in \mathbb{R}^{d_y \times d}$ parameterize the spatial transformation and project the sequences into the same low-dimensional coordinate system. Constrained by (5), $\mathbf{W}_x$ and $\mathbf{W}_y$ warp the signal in time to maximize the temporal correlation. Similar to CCA, $\phi(\cdot)$ is a regularization term (2) for $\mathbf{V}_x$ and $\mathbf{V}_y$. In addition, the projections satisfy the constraints,

$$\begin{aligned}\Phi = \Big\{\{\mathbf{V}_x,\mathbf{V}_y\} \;\Big|\; &\mathbf{V}_x^T\Big((1-\lambda)\mathbf{X}\mathbf{W}_x\mathbf{W}_x^T\mathbf{X}^T + \lambda\mathbf{I}\Big)\mathbf{V}_x = \mathbf{I}, \\ &\mathbf{V}_y^T\Big((1-\lambda)\mathbf{Y}\mathbf{W}_y\mathbf{W}_y^T\mathbf{Y}^T + \lambda\mathbf{I}\Big)\mathbf{V}_y = \mathbf{I}\Big\},\end{aligned}$$

where $\lambda \in [0,1]$ is to trade-off between the least-squared error and the regularization term.

Equation (8) is the main contribution of this paper. CTW is a clean extension of CCA and DTW to align two signals in space and time. It extends previous work on CCA by adding temporal alignment and on DTW by allowing a feature selection and dimensionality reduction mechanism for aligning signals of different dimensions.

3.3 Optimization of CTW

Optimizing J_{ctw} is a non-convex optimization problem with respect to the warping matrices and projection matrices. We take a coordinate-descent approach that alternates between solving the temporal alignment using DTW, and computing the spatial projections using CCA.

Given the warping matrices, the optimal projection matrices are the leading d generalized eigenvectors, i.e., $[\mathbf{V}_x;\mathbf{V}_y] = \mathrm{eig}_d(\mathbf{A},\mathbf{B})$, where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{X}\mathbf{W}_x\mathbf{W}_y^T\mathbf{Y}^T \\ \mathbf{Y}\mathbf{W}_y\mathbf{W}_x^T\mathbf{X}^T & \mathbf{0} \end{bmatrix},$$

$$\mathbf{B} = (1-\lambda)\begin{bmatrix} \mathbf{X}\mathbf{W}_x\mathbf{W}_x^T\mathbf{X}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}\mathbf{W}_y\mathbf{W}_y^T\mathbf{Y}^T \end{bmatrix} + \lambda\mathbf{I}.$$

In most experiments, we initialized CTW by setting $\mathbf{W}_x$ and $\mathbf{W}_y$ a uniform warping that aligns the sequences. In this case, the warping paths are computed as

$$\mathbf{p}_x = round(linspace(1, n_x, l))', \tag{9}$$

where $round(\cdot)$ and $linspace(\cdot)$ are MATLAB functions. But CTW can be initialized by any other time warping method such as DTW or iterative motion warping [9]. The dimension d can be selected to preserve a certain amount (e.g., 90 percent) of the total correlation. Once the spatial transformation is computed, the temporal alignment is computed using standard approaches for DTW. Alternating between these two steps (spatial and temporal alignment) monotonically decreases J_{ctw}. J_{ctw} is bounded below, so the proposed algorithm will converge to a critical point.

4 Generalized Canonical Time Warping (GCTW)

In the previous section, we described CTW, a technique that is able to align two multi-modal sequences with different features. However, CTW has three main limitations inherited from DTW: (1) The exact computational complexity of DTW for multi-dimensional sequences is quadratic both in space and time; (2) CTW and its extensions address the problem of aligning two sequences, but it is unclear how to extend it to the alignment of multiple sequences; (3) The temporal alignment is computed using DTW, which relies on DP to find the optimal path. In some problems (e.g., sub-sequence alignment) the warping path provided by DP is too rigid (e.g., the first and the last samples have to match).

To address these issues, this section proposes GCTW, an efficient technique for spatio-temporal alignment of multiple time series. To accommodate for subject variability and to take into account the difference in the dimensionality of the signals, GCTW uses multi-set CCA (mCCA). To compensate for temporal changes, GCTW extends DTW by incorporating a more efficient and flexible temporal warping parameterized by a set of monotonic basis functions. Unlike existing approaches based on DP with quadratic complexity, GCTW efficiently optimizes the time warping function using a Gauss-Newton algorithm, which has linear complexity in the length of the sequence.

4.1 Objective Function of GCTW

Given a collection of m time series, $\{\mathbf{X}_i\}_{i=1}^m$, GCTW aims to seek for each $\mathbf{X}_i = [\mathbf{x}_1^i,\ldots,\mathbf{x}_{n_i}^i] \in \mathbb{R}^{d_i \times n_i}$, a low-dimensional spatial embedding $\mathbf{V}_i \in \mathbb{R}^{d_i \times d}$ and a non-linear temporal transformation $\mathbf{W}_i = \mathbf{W}(\mathbf{p}_i) \in \{0,1\}^{n_i \times l}$ parameterized by $\mathbf{p}_i \in \{1:n_i\}^l$, such that the resulting sequence $\mathbf{V}_i^T\mathbf{X}_i\mathbf{W}_i \in \mathbb{R}^{d \times l}$ is well aligned with the others in the least-squared

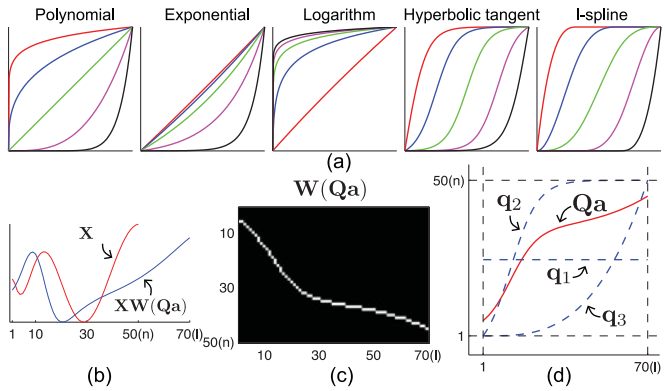

Fig. 3. Approximating temporal warping using monotonic bases. (a) Five common choices for monotonic bases. (b) An example of time warping $\mathbf{XW}(\mathbf{Qa}) \in \mathbb{R}^{1\times 70}$ of 1-D time series $\mathbf{X} \in \mathbb{R}^{1\times 50}$. (c) The warping matrix. (d) The warping function $\mathbf{Qa}$ is a linear combination of three basis functions including a constant function ($\mathbf{q}_1$) and two monotonic functions ($\mathbf{q}_2$ and $\mathbf{q}_3$).

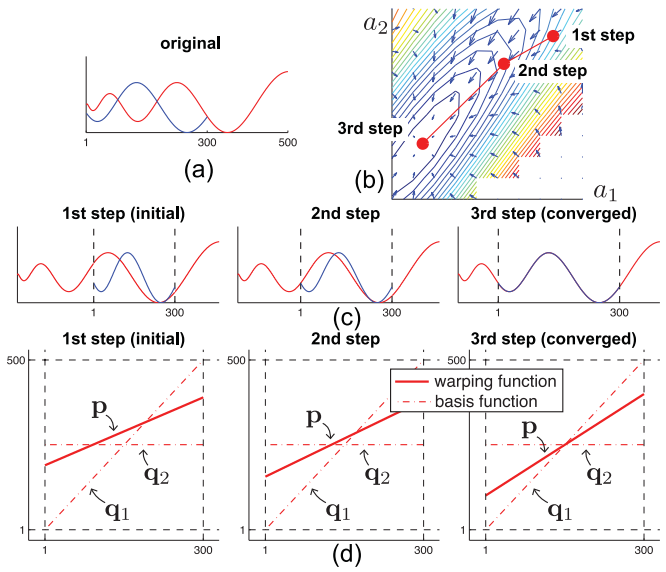

Fig. 4. An example of using Gauss-Newton for solving the sub-sequence alignment problem. (a) Two 1-D sequences. (b) The contour of the objective function (J_a as defined in (13)) with respect to the weights of two bases. (c) The Gauss-Newton optimization procedure, the longer red sequence is warped to match the shorter blue sequence. (d) Warping function ($\mathbf{p}$) as a combination of a linear function ($\mathbf{q}_1$) and a constant function ($\mathbf{q}_2$) used for scaling and translation respectively.

sense. In a nutshell, GCTW minimizes the sum of pairwise distances:

$$\min_{\{\mathbf{V}_i\}_i\in\Phi,\{\mathbf{p}_i\}_i\in\Psi} J_{gctw} = \sum_{i=1}^{m}\sum_{j=1}^{m}\frac{1}{2}\left\|\mathbf{V}_i^T\mathbf{X}_i\mathbf{W}_i - \mathbf{V}_j^T\mathbf{X}_j\mathbf{W}_j\right\|_F^2 + \sum_{i=1}^{m}(\phi(\mathbf{V}_i)+\psi(\mathbf{p}_i)), \tag{10}$$

where $\phi(\mathbf{V}_i) = m\lambda/(1-\lambda)\|\mathbf{V}_i\|_F^2$ is the regularization function penalizing the irregularity of the spatial transformation $\mathbf{V}_i$. $\lambda \in [0,1]$ is a trade-off parameter between the least-squared error and the regularization term. Following the multi-set CCA (mCCA) [35], GCTW constrains the spatial embeddings as:

$$\Phi = \left\{ \{\mathbf{V}_i\}_i \;\middle|\; \sum_{i=1}^{m}\mathbf{V}_i^T\big((1-\lambda)\mathbf{X}_i\mathbf{W}_i\mathbf{W}_i^T\mathbf{X}_i^T + \lambda\mathbf{I}\big)\mathbf{V}_i = \mathbf{I} \right\}.$$

$\psi(\cdot)$ and $\Psi(\cdot)$, defined in the following sections, are used to respectively regularize and constrain the temporal transformation $\mathbf{p}_i$.

Let us consider a single sequence $\mathbf{X} \in \mathbb{R}^{d\times n}$ and its temporal warping, $\mathbf{p} \in \{1:n\}^l$. While the possible composition of the temporal warping path, $\mathbf{p}$, is locally enforced by the original DTW constraints (5), the global shape of any valid $\mathbf{p}$ must correspond to a monotonic and continuous trajectory in matrix $\mathbf{W} \in \{0,1\}^{n\times l}$ starting from the upper-left corner and ending at the bottom-right one. Recall that any nonnegative combination of monotonic trajectories is guaranteed to be monotonic. GCTW parameterizes the warping path $\mathbf{p}$ as a linear combination of monotonic functions:

$$\mathbf{p} \approx \sum_{c=1}^{k} a_c\mathbf{q}_c = \mathbf{Qa}, \tag{11}$$

where $\mathbf{a} \in \mathbb{R}^k, \mathbf{a} \geq \mathbf{0}$ is the non-negative weight vector and $\mathbf{Q} = [\mathbf{q}_1,\ldots,\mathbf{q}_k] \in [1,n]^{l\times k}$ is the basis set composed of k pre-defined monotonically increasing functions. Fig. 3a illustrates five common choices for $\mathbf{q}_c$, including (1) polynomial (ax^b), (2) exponential ($\exp(ax+b)$), (3) logarithm ($\log(ax+b)$), (4) hyperbolic tangent ($\tanh(ax+b)$) and (5) I-spline [36]. Similar work by Fisher et al. [28] also used hyperbolic tangent functions as temporal bases, and the weights were optimized using a non-negative least squares algorithm. However, GCTW differs in three aspects: (1) GCTW allows aligning multi-dimensional time series that have different features, while [28] can only align one-dimensional time-series; (2) GCTW uses a more efficient eigen decomposition to solve mCCA and quadratic programming for optimizing the weights; and (3) GCTW uses a family of monotonic functions that allow for a more general warping (e.g., sub-sequence alignment), and constraints to regularize the solution.

To approximate the DTW constraints (5) on the warping path $\mathbf{p}$, we alternatively impose the following constraints on the weights $\mathbf{a}$.

Boundary conditions. We enforce the position of the first frame, $p_1 = \mathbf{q}^{(1)}\mathbf{a} \geq 1$, and the last frame, $p_l = \mathbf{q}^{(l)}\mathbf{a} \leq n$, where $\mathbf{q}^{(1)} \in \mathbb{R}^{1\times k}$ and $\mathbf{q}^{(l)} \in \mathbb{R}^{1\times k}$ are the first and last rows of the basis matrix $\mathbf{Q}$ respectively. In contrast to DTW, which imposes a tight boundary (i.e., $p_1 = 1$ and $p_l = n$), GCTW allows $\mathbf{p}$ to index a sub-part of $\mathbf{X}$. This relaxation is useful in solving the more general problem of sub-sequence alignment. For instance, Fig. 4 illustrates an example of matching a shorter 1-D sequence (blue) to a sub-sequence of the longer one (red). In this sub-sequence alignment problem, GCTW models the time warping $\mathbf{p}$ as a combination of a linear basis $\mathbf{q}_1$ and a constant basis $\mathbf{q}_2$.

Monotonicity. We enforce $t_1 \leq t_2 \Rightarrow p_{t_1} \leq p_{t_2}$ by constraining the sign of the weight: $\mathbf{a} \geq \mathbf{0}$. Note that constraining the weights is a sufficient condition to ensure monotonicity but it is not necessary. See [37], [38], [39] for in-depth discussions on monotonic functions.

Continuity. To approximate the hard constraint on the step size, GCTW penalizes the curvature of the warping

path using a temporal regularization term, $\sum_{t=1}^{l} \|\nabla \mathbf{q}^{(t)}\mathbf{a}\|_2^2 \approx \|\mathbf{F}_l\mathbf{Q}\mathbf{a}\|_2^2$ where $\mathbf{F}_l \in \mathbb{R}^{l\times l}$ is the 1st order differential operator.

In summary, we constrain the warping path in (10) by adding the following constraints on $\mathbf{a}$,

$$\begin{aligned} &\psi(\mathbf{a}) = \eta\|\mathbf{F}_l\mathbf{Q}\mathbf{a}\|_2^2, \quad \Psi = \{\mathbf{a} \mid \mathbf{L}\mathbf{a} \le \mathbf{b}\}, \\ &\text{where } \mathbf{L} = [-\mathbf{I}_k; -\mathbf{q}^{(1)}; \mathbf{q}^{(l)}] \text{ and } \mathbf{b} = [\mathbf{0}_k; -1; n]. \end{aligned} \tag{12}$$

Therefore, given a basis set of k monotone functions, all feasible weights belong to a polyhedron in $\mathbb{R}^k$ parameterized by $\mathbf{L} \in \mathbb{R}^{(k+2)\times k}$ and $\mathbf{b} \in \mathbb{R}^{k+2}$. For instance, Fig. 3d illustrates an example of a warping function (solid line) as a combination of three monotone functions (dotted lines).

4.2 Optimization of GCTW w.r.t. Spatial Basis

Minimizing J_{gctw} (10) is a non-convex optimization problem with respect to the temporal transformation and the spatial projection. We optimize GCTW by alternating between solving for the time warping using an efficient Gauss-Newton algorithm (see Section 4.3), and computing the spatial transformation using mCCA.

Assuming the time warping is fixed, mCCA computes the optimal $\{\mathbf{V}_i\}_i$ using a generalized eigen decomposition as $[\mathbf{V}_1; \ldots; \mathbf{V}_m] = \text{eig}_d(\mathbf{A}, \mathbf{B})$, where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{Y}_1\mathbf{Y}_1^T & \cdots & \mathbf{Y}_1\mathbf{Y}_m^T \\ \vdots & \ddots & \vdots \\ \mathbf{Y}_m\mathbf{Y}_1^T & \cdots & \mathbf{Y}_m\mathbf{Y}_m^T \end{bmatrix} - \begin{bmatrix} \mathbf{Y}_1\mathbf{Y}_1^T & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Y}_m\mathbf{Y}_m^T \end{bmatrix},$$

$$\mathbf{B} = (1-\lambda)\begin{bmatrix} \mathbf{Y}_1\mathbf{Y}_1^T & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Y}_m\mathbf{Y}_m^T \end{bmatrix} + \lambda\mathbf{I}, \ \mathbf{Y}_i \triangleq \mathbf{X}_i\mathbf{W}_i.$$

These steps monotonically decrease J_{gctw}, and because the function is bounded below, the alternating scheme will converge to a critical point.

4.3 Optimization of GCTW w.r.t. Temporal Weights

This section describes how GCTW relaxes the warping path to be a linear combination of monotonic paths, 11 provides a new model for temporal alignment and new methods for optimizing it. Given k bases, $\mathbf{Q} \in \mathbb{R}^{l\times k}$, optimizing (10) with respect to the warping paths $\{\mathbf{p}_i\}_i$ can be written as:

$$\begin{aligned} \min_{\{\mathbf{a}_i\}_i} J_a = &\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{1}{2}\big\| \underbrace{\mathbf{V}_i^T\mathbf{X}_i\mathbf{W}(\mathbf{Q}\mathbf{a}_i)}_{\mathbf{Z}(\mathbf{a}_i)} - \underbrace{\mathbf{V}_j^T\mathbf{X}_j\mathbf{W}(\mathbf{Q}\mathbf{a}_j)}_{\mathbf{Z}(\mathbf{a}_j)} \big\|_F^2 \\ &+ \sum_{i=1}^{m}\eta\|\mathbf{F}_l\mathbf{Q}\mathbf{a}_i\|_2^2, \quad \text{s.t. } \mathbf{L}_i\mathbf{a}_i \le \mathbf{b}_i, \ \forall i, \end{aligned} \tag{13}$$

where $\mathbf{W}(\cdot)$ is a non-linear mapping function defined in (7). $\mathbf{L}_i$ and $\mathbf{b}_i$ are used to constrain the monotonicity and boundary of the time warping for each sequence. To shorten in notation, we denote each term $\mathbf{V}_i^T\mathbf{X}_i\mathbf{W}(\mathbf{Q}\mathbf{a}_i) \in \mathbb{R}^{d\times l}$ as $\mathbf{Z}(\mathbf{a}_i)$.

A direct optimization of J_a is difficult due to the non-linear function $\mathbf{W}(\cdot)$. Inspired by the Lucas-Kanade framework for image alignment [40], we approximate the temporal alignment problem using a Gauss-Newton method. More specifically, GN iteratively updates the weights $\hat{\mathbf{a}}_i \leftarrow \mathbf{a}_i + \delta_i$ by minimizing a series of first-order Taylor approximations of J_a centered at each term $\mathbf{Z}(\mathbf{a}_i)$ given the initial $\mathbf{a}_i$, where $\delta_i \in \mathbb{R}^k$ denotes the increment of the weight $\mathbf{a}_i$.

To better understand the approximation, let us first focus on, $\mathbf{z}_t(\mathbf{a}_i) \in \mathbb{R}^d$, the tth column of $\mathbf{Z}(\mathbf{a}_i) = [\mathbf{z}_1(\mathbf{a}_i), \ldots, \mathbf{z}_l(\mathbf{a}_i)]$, which can be rewritten as,

$$\mathbf{z}_t(\mathbf{a}_i) = \big[\mathbf{V}_i^T\mathbf{X}_i\mathbf{W}(\mathbf{Q}\mathbf{a}_i)\big]_t = \mathbf{V}_i^T\mathbf{X}_i[\mathbf{W}(\mathbf{Q}\mathbf{a}_i)]_t, \tag{14}$$

where $[\cdot]_t$ denotes the tth column of a matrix. According to the definition of $\mathbf{W}(\cdot)$ in (7), $[\mathbf{W}(\mathbf{Q}\mathbf{a}_i)]_t \in \{0,1\}^n$ is a binary vector with only one non-zero element located at $\mathbf{q}^{(t)}\mathbf{a}_i$, where $\mathbf{q}^{(t)} \in \mathbb{R}^{1\times k}$ is the tth row of $\mathbf{Q}$. In other words, $\mathbf{z}_t(\mathbf{a}_i)$ is a replication of $\mathbf{q}^{(t)}\mathbf{a}_i^{th}$ column of the signal $\mathbf{V}_i^T\mathbf{X}_i$, i.e., $\mathbf{z}_t(\mathbf{a}_i) = [\mathbf{V}_i^T\mathbf{X}_i]_{\mathbf{q}^{(t)}\mathbf{a}_i}$. Following [40], we approximate $\mathbf{z}_t(\mathbf{a}_i + \delta_i)$ as,

$$\mathbf{z}_t(\mathbf{a}_i + \delta_i) \approx \mathbf{z}_t(\mathbf{a}_i) + \nabla\big(\mathbf{V}_i^T\mathbf{X}_i\big)\big|_{\mathbf{q}^{(t)}\mathbf{a}_i}\frac{\partial \mathbf{q}^{(t)}\mathbf{a}_i}{\partial \mathbf{a}_i}\delta_i, \tag{15}$$

where $\nabla(\mathbf{V}_i^T\mathbf{X}_i)|_{\mathbf{q}^{(t)}\mathbf{a}_i} \in \mathbb{R}^d$ denotes the row-wise gradient[2] of the signal $\mathbf{V}_i^T\mathbf{X}_i$ around column $\mathbf{q}^{(t)}\mathbf{a}_i$. The term, $\partial\mathbf{q}^{(t)}\mathbf{a}_i/\partial\mathbf{a}_i = \mathbf{q}^{(t)} \in \mathbb{R}^{1\times k}$ is the Jacobian of the time warping. Putting together the approximations of each column of $\mathbf{Z}(\mathbf{a}_i + \delta_i)$ using (15) yields:

$$\begin{aligned} &\text{vec}(\mathbf{Z}(\mathbf{a}_i + \delta_i)) \approx \mathbf{v}_i + \mathbf{G}_i\delta_i, \\ &\text{where } \mathbf{v}_i = \text{vec}(\mathbf{Z}(\mathbf{a}_i)), \mathbf{G}_i = \begin{bmatrix} \nabla\big(\mathbf{V}_i^T\mathbf{X}_i\big)|_{\mathbf{q}^{(1)}\mathbf{a}_i}\mathbf{q}^{(1)} \\ \vdots \\ \nabla\big(\mathbf{V}_i^T\mathbf{X}_i\big)|_{\mathbf{q}^{(l)}\mathbf{a}_i}\mathbf{q}^{(l)} \end{bmatrix}. \end{aligned} \tag{16}$$

Plugging (16) in (13) yields the approximation:

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{1}{2}\|\mathbf{v}_i + \mathbf{G}_i\delta_i - \mathbf{v}_j - \mathbf{G}_j\delta_j\|_2^2 + \sum_{i=1}^{m}\eta\|\mathbf{F}_i\mathbf{Q}(\mathbf{a}_i + \delta_i)\|_2^2.$$

Minimizing it with respect to all the weight increments $\delta = [\delta_1; \ldots; \delta_m] \in \mathbb{R}^{mk}$ yields a quadratic programming problem:

$$\min_{\delta} \quad \frac{1}{2}\delta^T\mathbf{H}\delta + \mathbf{f}^T\delta, \quad \text{s.t.} \quad \mathbf{L}\delta \le \mathbf{b} - \mathbf{L}\mathbf{a}, \tag{17}$$

2. The gradient is independently computed for each row of $\mathbf{V}_i^T\mathbf{X}_i$.

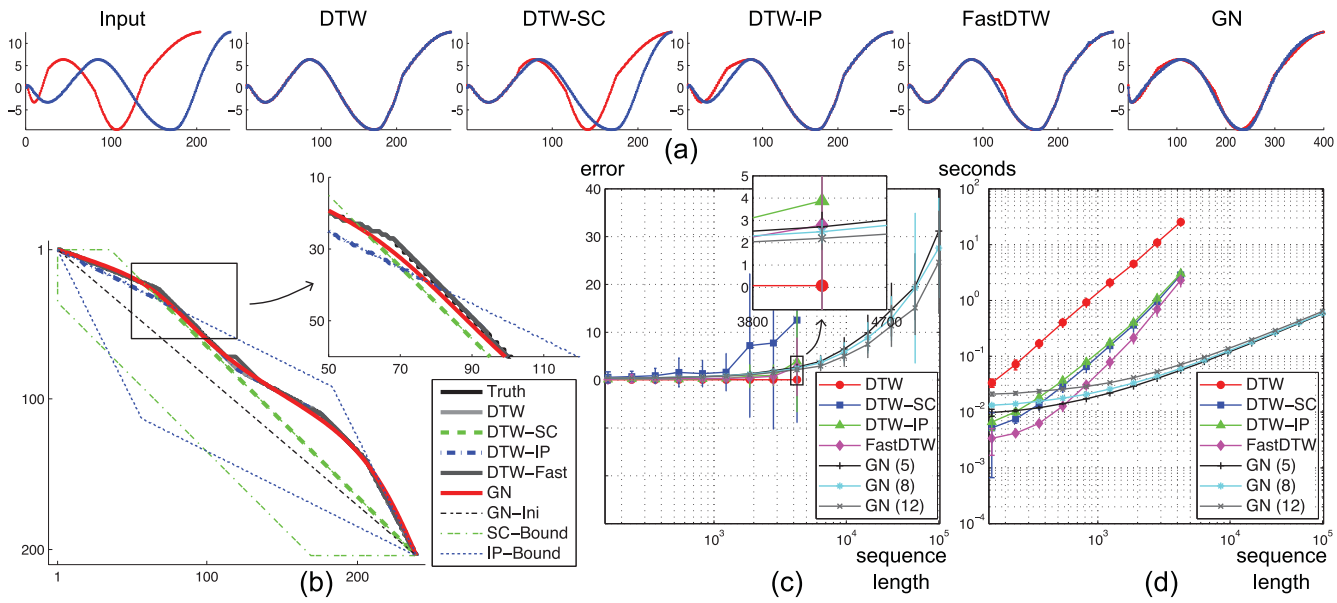

Fig. 5. Comparison between Gauss-Newton and variants of DTW for temporal alignment. (a) An example of two 1-D time series and the alignment results calculated using DTW, DTW constrained in the Sakoe-Chiba band (DTW-SC) and the Itakura Parallelogram band (DTW-IP), DTW optimized in a multi-level scheme (FastDTW) and Gauss-Newton (GN). (b) Comparison of different warping paths. GN-Init denotes the initial warping used for GN. SC-Bound and IP-Bound denote the boundaries of SC band and IP band respectively. (c) Alignment errors. (d) Computational costs.

whose components are computed as follows:

$$\mathbf{H} = \begin{bmatrix} m\mathbf{G}_1^T\mathbf{G}_1 + \eta\mathbf{Q}^T\mathbf{F}_1^T\mathbf{F}_1\mathbf{Q} & \cdots & -\mathbf{G}_1^T\mathbf{G}_m \\ \vdots & \ddots & \vdots \\ -\mathbf{G}_m^T\mathbf{G}_1 & \cdots & m\mathbf{G}_m^T\mathbf{G}_m + \eta\mathbf{Q}^T\mathbf{F}_m^T\mathbf{F}_m\mathbf{Q} \end{bmatrix},$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{G}_1^T(m\mathbf{v}_1 - \sum_i \mathbf{v}_i) + \eta\mathbf{Q}^T\mathbf{F}_1^T\mathbf{F}_1\mathbf{Q}\mathbf{a}_1 \\ \vdots \\ \mathbf{G}_m^T(m\mathbf{v}_m - \sum_i \mathbf{v}_i) + \eta\mathbf{Q}^T\mathbf{F}_m^T\mathbf{F}_m\mathbf{Q}\mathbf{a}_m \end{bmatrix},$$

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_m \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}, \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{L}_m \end{bmatrix}.$$

Note that the objective function of (17) is convex. Fig. 4 illustrates an example of aligning two 1-D time series (Fig. 4a) using this approach. To achieve sub-sequence alignment, we model the time warping path $\mathbf{p}$ as a combination of a linear basis $\mathbf{q}_1$ and a constant one $\mathbf{q}_2$ (Fig. 4d). As shown in Fig. 4c, Gauss-Newton takes three steps to find the optimal warping parameter in a 2-D space (Fig. 4b).

In most experiments, we initialized $\mathbf{a}_i$ by uniformly aligning the sequences (see GN-Init curve in Fig. 5b). However, better results can be achieved by using a more sophisticated initialization method. The length of the warping path l is usually set to be $l = 1.1\max_{i=1}^{m} n_i$. The computational complexity of the algorithm is $O(dlmk + m^3k^3)$.

4.4 Comparison with Other DTW Techniques

As discussed in [1], [33], there are various techniques that have been proposed to accelerate DTW. For instance, the Sakoe-Chiba band (DTW-SC) and the Itakura Parallelogram band (DTW-IP) reduce the complexity of the original DTW algorithm to $O(\beta n_x n_y)$ by constraining the warping path to be in a band of a certain shape, where $\beta < 1$ is the size ratio between the band and the original search space of DTW. However, using a narrow band (a small β) cuts off potential warping space, and may lead to a sub-optimal solution. For instance, Fig. 5a shows an example of two 1-D time series and the alignment results calculated by different algorithms. The results computed by DTW-SC and DTW-IP are less accurate than the ones computed using our proposed Gauss-Newton. This is because both the SC and IP bands are over-constrained (Fig. 5b). Alternatively, instead of constraining the warping path, exhaustive DTW search can be approximated in a multi-level scheme. For instance, Salvador and Chan [33] introduced FastDTW by recursively projecting a solution from a coarser resolution and refining the projected solution in a higher resolution. Although the coarse-to-fine framework could largely reduce the search space, the solution is not exact. See Fig. 6 for a detailed comparison.

To provide a quantitative evaluation, we synthetically generated 1-D sequences at 15 scales. For DTW-SC, we set the bandwidth to be $\beta = 0.1$, which is common in practical applications. For FastDTW, we recursively shrink the sequence length in half from the finest level to the coarsest one. We then propagated the DTW solution from coarse to fine with radius $r = 5$. For GN, we varied k to be $5, 8, 12$ to investigate the effect of the number of bases. For each scale, we randomly generated 100 pairs of sequences. The error was computed using (19) and shown in Figs. 5c and 5d. DTW obtains the lowest error but takes the most time to compute. This is because DTW exhaustively searches the entire parameter space to find the global optima. DTW-SC, DTW-IP and FastDTW all need less time than DTW because they search a smaller space. Empirically, DTW-SC is the least accurate compared to DTW-IP and FastDTW in our synthetic dataset. GN is most computationally efficient because it has linear complexity in the sequence length. Moreover, increasing the number of bases monotonically reduces the error.

	Method	Degrees-of-Freedom		Complexity $O(\cdot)$	
		Embedding	Warping	Embedding	Warping
Two-sequence alignment $\mathbf{X} \in \mathbb{R}^{d_x \times n_x}$ $\mathbf{Y} \in \mathbb{R}^{d_y \times n_y}$	DTW	–	$2l$	–	$n_x n_y$
	DTW-SC	–	$2l$	–	$\beta n_x n_y$
	DTW-IP	–	$2l$	–	$\beta n_x n_y$
	FastDTW	–	$2l$	–	$r(n_x + n_y)/2$
	DDTW	–	$2l$	–	$n_x n_y$
	IMW	$2dn_x$	$2l$	$dLS(2n_x)$	$n_x n_y$
	CTW	$d(d_x + d_y)$	$2l$	$eig(d_x + d_y)$	$n_x n_y$
	GCTW	$d(d_x + d_y)$	$2k$	$eig(d_x + d_y)$	$2dlk + 8k^3$
Multi-sequence alignment $\{\mathbf{X}_i \in \mathbb{R}^{d_i \times n_i}\}_i$	DTW	–	ml	–	$\Pi_i n_i$
	mDTW	–	ml	–	$l \sum_i n_i$
	mDDTW	–	ml	–	$l \sum_i n_i$
	mIMW	$2l \sum_i d_i$	ml	$\sum_i d_i LS(2l)$	$l \sum_i n_i$
	mCTW	$d \sum_i d_i$	ml	$eig(\sum_i d_i)$	$l \sum_i n_i$
	GCTW	$d \sum_i d_i$	mk	$eig(\sum_i d_i)$	$dlmk + m^3 k^3$

Fig. 6. Comparison of temporal alignment algorithms as a function of degrees-of-freedom and complexity. l is the length of warping path. $LS(n)$ and $eig(n)$ denote the complexity of solving a least-squares of n variables and a generalized eigenvalue problem with two n-by-n matrices, respectively.

5 Experiments

This section compares CTW and GCTW against state-of-the-art methods for temporal alignment of time series in seven experiments. In the first experiment, we compared the performance of CTW and GCTW against DTW, DDTW [20], and IMW [9] in the problem of aligning synthetic time series of varying complexity. In the second experiment, we aligned videos of different subjects performing a similar activity; each video is represented using different types of visual features. In the third experiment, we showed how GCTW and CTW can be applied to provide a metric useful for activity recognition. In the fourth experiment, we aligned facial expressions across subjects on videos with naturally occurring facial behavior. In the fifth experiment, we showed how GCTW can be applied to large-scale alignment. We aligned approximately $50{,}000$ frames of motion capture data of two subjects cooking the same recipe. In the sixth experiment, we showed how GCTW can be used to localize common sub-sequences between two time series. The last experiment shows how GCTW is able to align three sequences of different subjects performing a similar action recorded with different sensors (motion capture data, accelerometers and video). In the first four experiments, the ground truth was known and we provided quantitative evaluation of the performance. In the others, we evaluated the quality of the alignment visually.

5.1 Evaluation Methods

In the experiments, we compared CTW and GCTW with several state-of-the-art methods for temporal alignment of time series. Below, we provide a brief description of the techniques that we used for comparison.

DTW and mDTW. DTW is solved using a standard dynamic programming algorithm that minimizes (4). To evaluate the performance of temporal alignment of multiple sequences, we extended the concept of Procrustes analysis [41] to time series. That is, given $m\ (> 2)$ time series, multi-sequence DTW (mDTW) seeks for a set of warping paths $\{\mathbf{p}_i\}_i$ that minimizes:

$$\begin{aligned} \min_{\{\mathbf{p}_i \in \mathbf{\Psi}\}_i} J_{mdtw} &= \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{1}{2} \|\mathbf{X}_i \mathbf{W}_i - \mathbf{X}_j \mathbf{W}_j\|_F^2 \\ &= m \sum_{i=1}^{m} \left\| \mathbf{X}_i \mathbf{W}_i - \frac{1}{m} \sum_{j=1}^{m} \mathbf{X}_j \mathbf{W}_j \right\|_F^2. \end{aligned} \tag{18}$$

mDTW alternates between independently solving for each $\mathbf{p}_i$ using an asymmetrical DTW and updating the mean sequence $\frac{1}{m}\sum_{j=1}^{m} \mathbf{X}_j \mathbf{W}_j$ by averaging $\{\mathbf{X}_i \mathbf{W}_i\}_i$.

DDTW and mDDTW. In order to make DTW invariant to translation, derivative DTW (DDTW) [20] uses the derivatives of the original features and minimizes:

$$\min_{\{\mathbf{p}_x, \mathbf{p}_y\} \in \mathbf{\Psi}} J_{ddtw} = \|\mathbf{X}\mathbf{F}_{n_x}^T \mathbf{W}_x - \mathbf{Y}\mathbf{F}_{n_y}^T \mathbf{W}_y\|_F^2,$$

where $\mathbf{F}_{n_x}$ and $\mathbf{F}_{n_y}$ are the 1^{st} order differential operators. To align multiple sequences, multi-sequence DDTW (mDDTW) extends DDTW in the Procrustes framework similar to (18).

IMW and mIMW. Similar to CTW, iterative motion warping [9] alternates between time warping and spatial transformation to align two sequences. Assuming the same number of spatial features between $\mathbf{X} \in \mathbb{R}^{d \times n_x}$ and $\mathbf{Y} \in \mathbb{R}^{d \times n_y}$, IMW translates and re-scales each feature in $\mathbf{X}$ independently to match with $\mathbf{Y}$. Written in a simple matrix form, IMW minimizes:

$$\begin{aligned} \min_{\mathbf{p}_x \in \mathbf{\Psi}, \mathbf{A}_x, \mathbf{B}_x} \quad J_{imw} &= \|(\mathbf{X} \circ \mathbf{A}_x + \mathbf{B}_x)\mathbf{W}_x - \mathbf{Y}\|_F^2 \\ &\quad + \lambda_a \left\|\mathbf{A}_x \mathbf{F}_{n_x}^T\right\|_F^2 + \lambda_b \left\|\mathbf{B}_x \mathbf{F}_{n_x}^T\right\|_F^2, \end{aligned}$$

where $\mathbf{A}_x, \mathbf{B}_x \in \mathbb{R}^{d \times n_x}$ are the scaling and translation parameters respectively. λ_a and λ_b are the weights for the least-squared error and the regularization terms. The regularization terms are used to enforce a smooth change in the columns of $\mathbf{A}_x$ and $\mathbf{B}_x$. In the experiments, we set them to be $\lambda_a = \lambda_b = 1$. IMW takes a coordinate-descent approach to optimize the time warping, scaling and translation. Given the warping matrix $\mathbf{W}_x$, the optimal spatial transformation can be computed in closed-form. To align multiple sequences, we extended IMW to multi-sequence IMW (mIMW) in the Procrustes framework similar to mDTW (18).

mCTW. CTW was originally proposed to align two multimodal sequences. We extended CTW to multi-sequence CTW (mCTW) for aligning multiple time series using the Procrustes analysis framework. mCTW optimizes the same objective (10) as GCTW does, however, the temporal alignment step is different. mCTW alternates between warping each time series using asymmetric DTW and updating the mean sequence, while GCTW uses Gauss-Newton for jointly optimizing over all weights of the bases.

Fig. 6 compares different temporal alignment methods as a function of the number of variables to optimize and their computational complexity. The comparison reports two cases: (1) alignment of two sequences (top of the table) and more than two sequences. Given two time series, $\mathbf{X} \in \mathbb{R}^{d \times n_x}$ and $\mathbf{Y} \in \mathbb{R}^{d \times n_y}$, DTW and DDTW have the same complexity $O(n_x n_y)$ for finding the optimal l-length warping path. IMW additionally solves d least-squared problems for each row of $\mathbf{A}_x$ and $\mathbf{B}_x$ independently. Similarly, CTW relies on DTW to optimize the

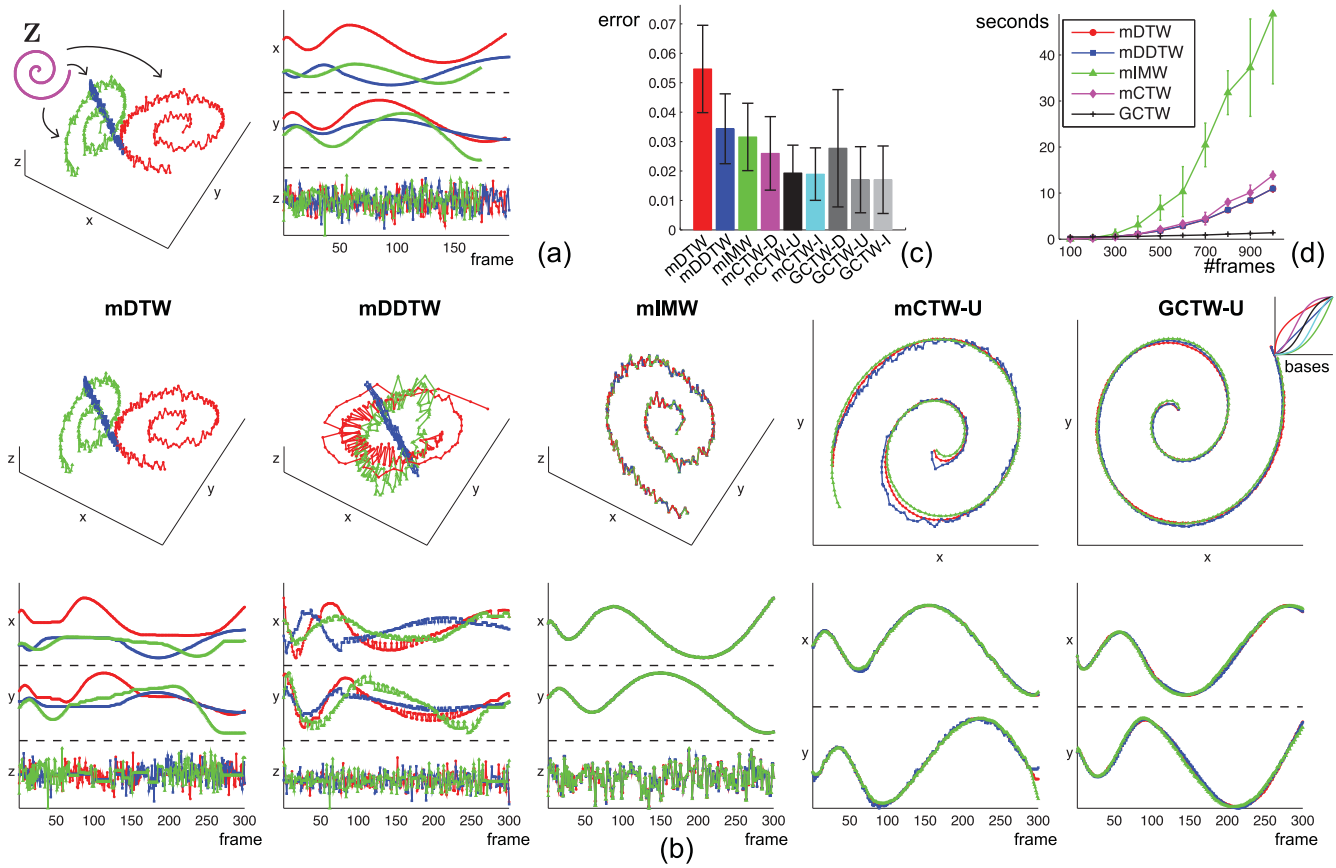

Fig. 7. Comparison of temporal alignment algorithms on the synthetic dataset. (a) An example of three synthetic time series generated by performing a random spatio-temporal transformation of a 2-D latent sequence $\mathbf{Z}$ and adding Gaussian noise in the third dimension. (b) The alignment results. (c) Mean and variance of the alignment errors. (d) Mean and variance of the computational cost (time in seconds).

time warping, resulting in a complexity of $O(n_x n_y)$ in both space and time. However, CTW uses CCA to accommodate the difference in the number of features by solving a generalized eigen-decomposition of two $(d_x + d_y)$-by-$(d_x + d_y)$ matrices. CTW has fewer variables than IMW and thus is less likely to overfit the data. Compared to CTW, GCTW has the same complexity for computing the spatial embedding. The main computational advantage of GCTW is using Gauss-Newton, which optimizes a small-scale QP with $2k$ variables for the time warping. Given m sequences, $\{\mathbf{X}_i \in \mathbb{R}^{d_i \times n_i}\}_{i=1}^{m}$, a direct generalization of the DTW is computationally infeasible due to the combinatorial explosion of possible warpings, incurring a complexity of $O(\prod_{i=1}^{m} n_i)$. In the experiment, mDTW is used as an approximation of the exact DTW optimization. However, mDTW and other DTW-based methods (mDDTW, mIMW and mCTW) still have quadratic complexity. Instead, GCTW approximates the combinatorial problem of time warping as a continuous optimization that can be more efficiently optimized by solving a small-scale QP with mk variables.

5.2 Evaluation Metric

For all experiments, we used the normalized distance from the ground-truth as an error metric. More specifically, let us denote the alignment result of m sequences by a set of time warping paths, $\mathbf{P}_{alg} = [\mathbf{p}_1^{alg}, \ldots, \mathbf{p}_m^{alg}] \in \mathbb{R}^{l_{alg} \times m}$, where $\mathbf{p}_i^{alg} \in \mathbb{R}^{l_{alg}}$ is the time warping path for the ith sequence. To evaluate the error of the time warping paths given by different methods, we computed their difference from the ground-truth path, $\mathbf{P}_{tru} = [\mathbf{p}_1^{tru}, \ldots, \mathbf{p}_m^{tru}] \in \mathbb{R}^{l_{tru} \times m}$, where the number of warping steps (l_{alg} and l_{tru}) could be different. To better understand the error, let us consider each warping path $\mathbf{P} \in \mathbb{R}^{l \times m}$ as a curve in $\mathbb{R}^m$ with l points (rows of $\mathbf{P}$). For instance, Figs. 8c and 10c compare the warping paths as 3-D and 2-D curves respectively. The error can be hence defined as the normalized distance between the curves $\mathbf{P}_{alg}$ and $\mathbf{P}_{tru}$,

$$\begin{aligned} error &= \frac{1}{2}\Big(dist(\mathbf{P}_{alg}, \mathbf{P}_{tru}) + dist(\mathbf{P}_{tru}, \mathbf{P}_{alg})\Big), \\ \text{where } dist(\mathbf{P}_1, \mathbf{P}_2) &= \frac{1}{l_1 l_2} \sum_{i=1 \cdots l_1} \min_{j=1 \cdots l_2} \left\| \mathbf{p}_1^{(i)} - \mathbf{p}_2^{(j)} \right\|_2. \end{aligned} \tag{19}$$

The term, $\min_{j=1 \cdots l_2} \|\mathbf{p}_1^{(i)} - \mathbf{p}_2^{(j)}\|_2$, measures the shortest distance between the point $\mathbf{p}_1^{(i)}$ and any point on the curve $\mathbf{P}_2$, where $\mathbf{p}_1^{(i)} \in \mathbb{R}^{1 \times m}$ and $\mathbf{p}_2^{(j)} \in \mathbb{R}^{1 \times m}$ are the ith row of $\mathbf{P}_1$ and jth row of $\mathbf{P}_2$ respectively.

5.3 Aligning Synthetic Sequences

In the first experiment, we synthetically generated spatio-temporal signals (3-D in space and 1-D in time) to evaluate the performance of mCTW and GCTW. As shown in Fig. 7a, the signals were randomly generated by spatially

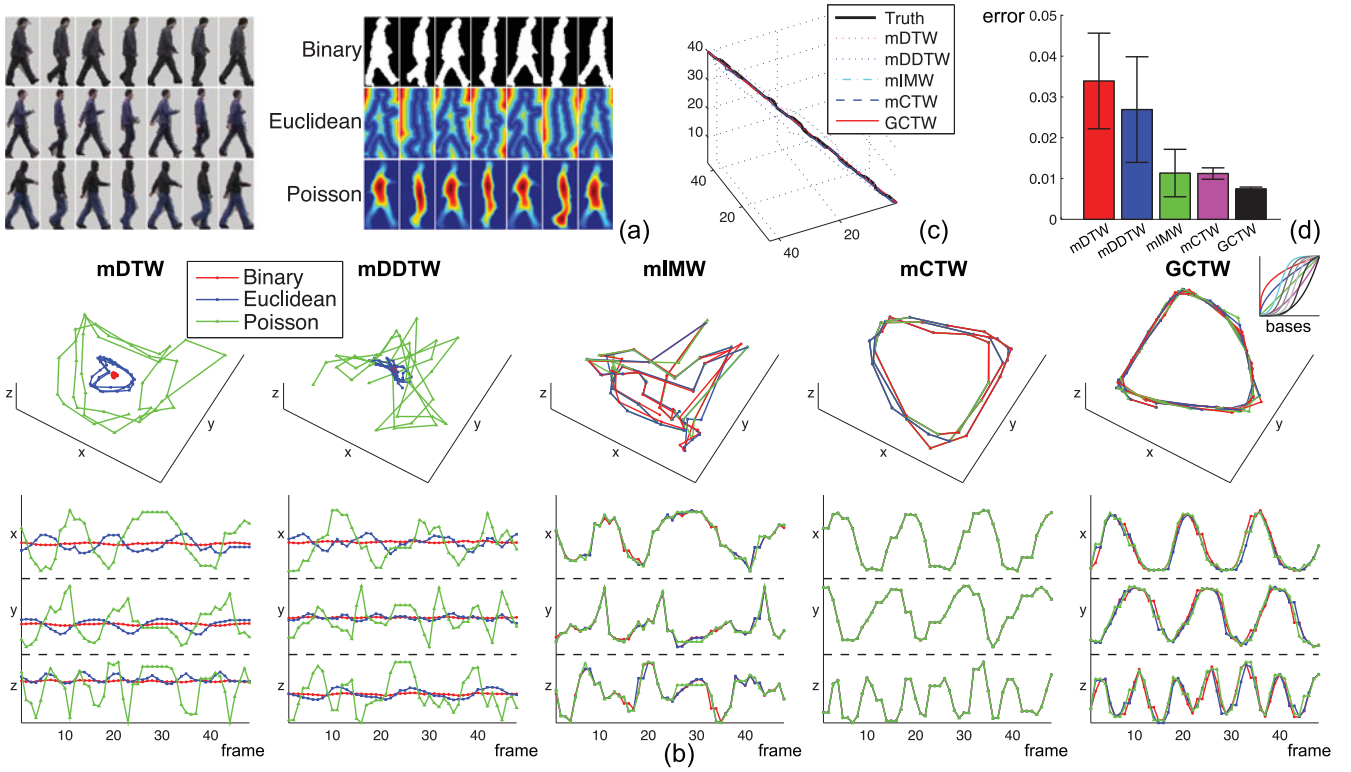

Fig. 8. Comparison of temporal alignment algorithms for aligning multi-feature video data. (a) An example of three aligned videos by GCTW. The left three sequences are the original frames after background subtraction, while the right three are the binary images, the Euclidean distance transforms and the solutions of the Poisson equation. (b) The alignment results. (c) Comparison of time warping paths. (d) Mean and variance of the alignment errors.

and temporally transforming a latent 2-D spiral, $\mathbf{Z} \in \mathbb{R}^{2\times l}, l = 300$ as $\mathbf{X} = [\mathbf{U}^T(\mathbf{Z} + \mathbf{b}\mathbf{1}^T)\mathbf{M}; \mathbf{e}^T] \in \mathbb{R}^{3\times n}$, where $\mathbf{U} \in \mathbb{R}^{2\times 2}$ and $\mathbf{b} \in \mathbb{R}^2$ were a randomly generated projection matrix and translation vector respectively. To synthesize the temporal distortion, a binary selection matrix $\mathbf{M} \in \{0, 1\}^{l\times n}$ was generated by randomly choosing $n \leq l$ columns from the identity matrix $\mathbf{I}_l$. The third spatial dimension $\mathbf{e} \in \mathbb{R}^n$ was added with a zero-mean Gaussian noise. In this experiment, mCTW and GCTW were compared with mDTW, mDDTW and mIMW for aligning three time series. The ground-truth alignment was known and the performance of each method was evaluated in terms of the alignment errors defined in (19). We repeated the above process 100 times with random numbers. In each trial, we studied three different initialization methods for mCTW and GCTW: uniform alignment for mCTW-U and GCTW-U as in (9), mDTW for mCTW-D and GCTW-D, and mIMW for mCTW-I and GCTW-I. The subspace dimensionality for CCA d was selected to preserve 90 percent of the total correlation. In this case, we have sufficient samples ($l = 300$) in 3-D space, and the regularization weight λ was set to zero. For GCTW, we selected three hyperbolic tangent and three polynomial functions as monotonic bases (upper-right corner in Fig. 7b).

Figs. 7 illustrates a comparison of the previously described methods for aligning multiple time series. Fig. 7b shows the spatio-temporal warping estimated by mDTW, mDDTW, mIMW, mCTW-U and GCTW-U. Fig. 7c shows the alignment errors (19) for 100 randomly generated time series. Both mDTW and mDDTW performed poorly in this case since they do not have a feature weight mechanism to adapt the spatial transformation of the sequences. mIMW warps sequences towards others by translating and re-scaling each frame in each dimension. Moreover, mIMW has more parameters ($2l\sum_i d_i$) than mCTW and GCTW ($d\sum_i d_i$), and hence mIMW is more prone to over-fitting. Furthermore, mIMW tries to fit the noisy dimension (third spatial component), biasing alignment in time, whereas both mCTW and GCTW had a feature selection mechanism that effectively canceled out the third dimension. Among all initializations, the uniform alignment (mCTW-U and GCTW-U) and mIMW (mCTW-I and GCTW-I) achieved the best results. It is important to notice that mCTW and GTW always improved the initial error, that is, mCTW-D and mCTW-I obtained lower errors than their initializations given by mDTW and mIMW respectively. Compared to mCTW, GCTW achieved better performance when aligning more than two sequences because GCTW jointly optimizes over all the possible time warpings for each time series, while mCTW takes a greedy approach by warping each sequence towards the mean sequence independently.

Fig. 7d evaluates the computational cost of each method with respect to the average sequence length. mIMW was the most computationally intensive method because it solves a least-squared problem for each feature dimension. mCTW was more expensive than DTW-based methods because of the additional computation to solve CCA. As expected, GCTW was the most efficient.

5.4 Aligning Videos with Different Features

In the second experiment, we used mCTW and GCTW to align video sequences of different people performing a similar action. Each video was encoded using different visual features. The video sequences were taken from the Weizmann database [42], which contains nine people performing 10 actions. To represent dynamic videos, we subtracted the background (the left three rows in Fig. 8a) and computed three popular shape features (the right three rows of Fig. 8a) for each 70-by-35 re-scaled mask image, including (1) binary image, (2) Euclidean distance transform [43], and (3) solution of Poisson equation [44]. In order to reduce the dimension of the feature space (2,450), we picked the top 123 principal components that preserved 99 percent of the total energy. We randomly selected three sequences and manually labeled their temporal correspondence as ground-truth. We repeated the process 10 times. All methods were initialized with uniform alignment. For GCTW, we used five hyperbolic tangent and five polynomial functions as the monotonic bases.

Fig. 8d shows the error for 10 randomly generated sets of videos. Neither mDTW nor mDDTW was able to align the videos because they were not able to handle alignment of signals of different dimensions. mIMW registered the top three components well in space; however, it overfitted the data and also computed a biased time warping path. In contrast, mCTW and GCTW warped the sequences accurately in both space and time.

5.5 Activity Classification

This section explores the use of CTW and GCTW as a (dis) similarity measure between videos with different features, and between videos and motion capture data. This (dis)similarity is used in combination with a nearest neighbor classifier.

Given a training set of videos of a subject performing an activity recorded with different visual features $\{\mathbf{X}_i\}_i$, our goal is to find the testing video $\{\mathbf{Y}_j\}_j$ that contains similar activity. We took 24 sequences from the Weizmann dataset [42]: eight people performed three actions (walking, running and jumping). We repeated our experiments 10 times. In each trial, we randomly split the 24 sequences into two disjoint sets of 12 sequences used for training and testing respectively. The training sequences $\{\mathbf{X}_i\}_i$ were represented using the binary silhouette (see left columns of Figs. 9a and 9b as examples) while the testing ones $\{\mathbf{Y}_j\}_j$ were encoded with the Euclidean distance features (see top rows of Figs. 9a and 9b as examples). Given a pair of sequences, $\mathbf{X}_i \in \mathbb{R}^{d_x \times n_{x_i}}$ and $\mathbf{Y}_j \in \mathbb{R}^{d_y \times n_{y_j}}$, we computed the (dis)similarity between videos using different methods: DTW, DDTW, IMW, CTW and GCTW. The warping path for each method is denoted as $\mathbf{W}_{x_i}^{alg} \in \mathbb{R}^{n_{x_i} \times l}$ and $\mathbf{W}_{y_j}^{alg} \in \mathbb{R}^{n_{y_j} \times l}$.

In order to provide a fair comparison with DTW, DDTW and mIMW (that cannot compute (dis)similarity between different features), we computed a pair of projection matrices, $\mathbf{V}_x \in \mathbb{R}^{d_x \times d}$ and $\mathbf{V}_y \in \mathbb{R}^{d_y \times d}$ for them. To do that, we took a subset of 1/3 of the training sequences encoded with both features $\{\mathbf{X}_i^{tr}\}$ and $\{\mathbf{Y}_i^{tr}\}$, and uniformaly aligned each pair of sequences of the same label. The aligned frames

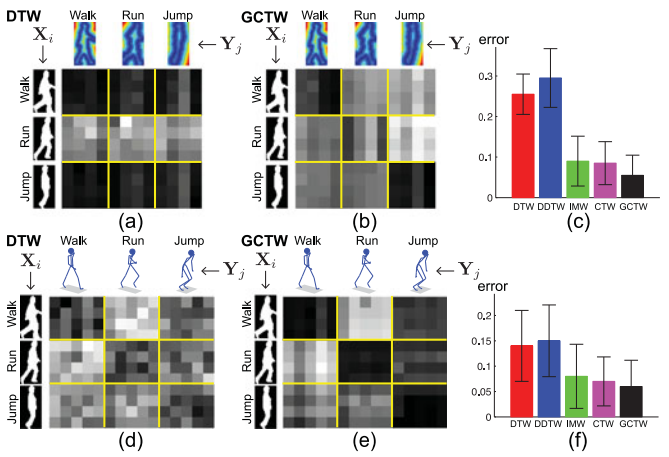

Fig. 9. Activity classification. (a) DTW (dis)similarity matrices computed between videos using different features: binary images $\{\mathbf{X}_i\}_i$ and Euclidean distance transforms $\{\mathbf{Y}_j\}_j$. A darker color indicates a smaller distance. (b) GCTW distance matrices. (c) Classification errors.

were concatenated in two matrices $\mathbf{X}^{tr}$ and $\mathbf{Y}^{tr}$, and the projections $\mathbf{V}_x$ and $\mathbf{V}_y$ were computed by CCA optimizing (1). Recall that the projection matrices were fixed for DTW, DDTW and IMW during testing, but for CTW and GCTW they were optimized by the algorithms. Then, the (dis)similarity between each pair of sequences was then computed as

$$dist(\mathbf{X}_i, \mathbf{Y}_j) = \frac{1}{l_{alg}} \left\| \mathbf{V}_x^T \mathbf{X}_i \mathbf{W}_{x_i}^{alg} - \mathbf{V}_y^T \mathbf{Y}_j \mathbf{W}_{y_j}^{alg} \right\|_F, \qquad (20)$$

where the alignment step l_{alg} was used to normalize the (dis)similarity. Given the (dis)similarity computed in (20), classification for a test sequence was done by finding the closest video in the training sequence. The overall classification error was averaged over all testing sequences.

Figs. 9a and 9b display the 12-by-12 (dis)similarity matrices computed by DTW and GCTW respectively. Each element in the matrices encodes the (dis)similarity between a training sequence (row) and a testing sequence (column). We re-ordered the rows and columns of the matrices so that the sequences containing the same activities were grouped in consecutive rows and columns. We then divided the matrix into nine 4-by-4 blocks (yellow lines), where the block in the ith row and jth column contains the (dis)similarity between the training sequences of the ith action and the testing data of the jth action, where $i, j \in \{\text{walk}, \text{run}, \text{jump}\}$. Darker color denotes smaller (dis)similarity. Ideally, if the (dis)similarity would be able to capture perfectly the activity, the matrix will be perfect with zeros (black color) in the diagonal blocks and higher values (white color) elsewhere.

From Fig. 9a, we can observe that the DTW (dis)similarity values in the first and last row of blocks were smaller than the blocks in the middle row. This is because, DTW does not have a feature adaptation mechanism and fails to provide semantic similarity of videos. In comparison, Fig. 9b shows that GCTW captured better the (dis)similarity between actions. Fig. 9c shows the nearest-neighbor classification error using different (dis)similarity. Overall, CTW and GCTW achieved lower errors than others due to their feature selection in aligning videos with different features.

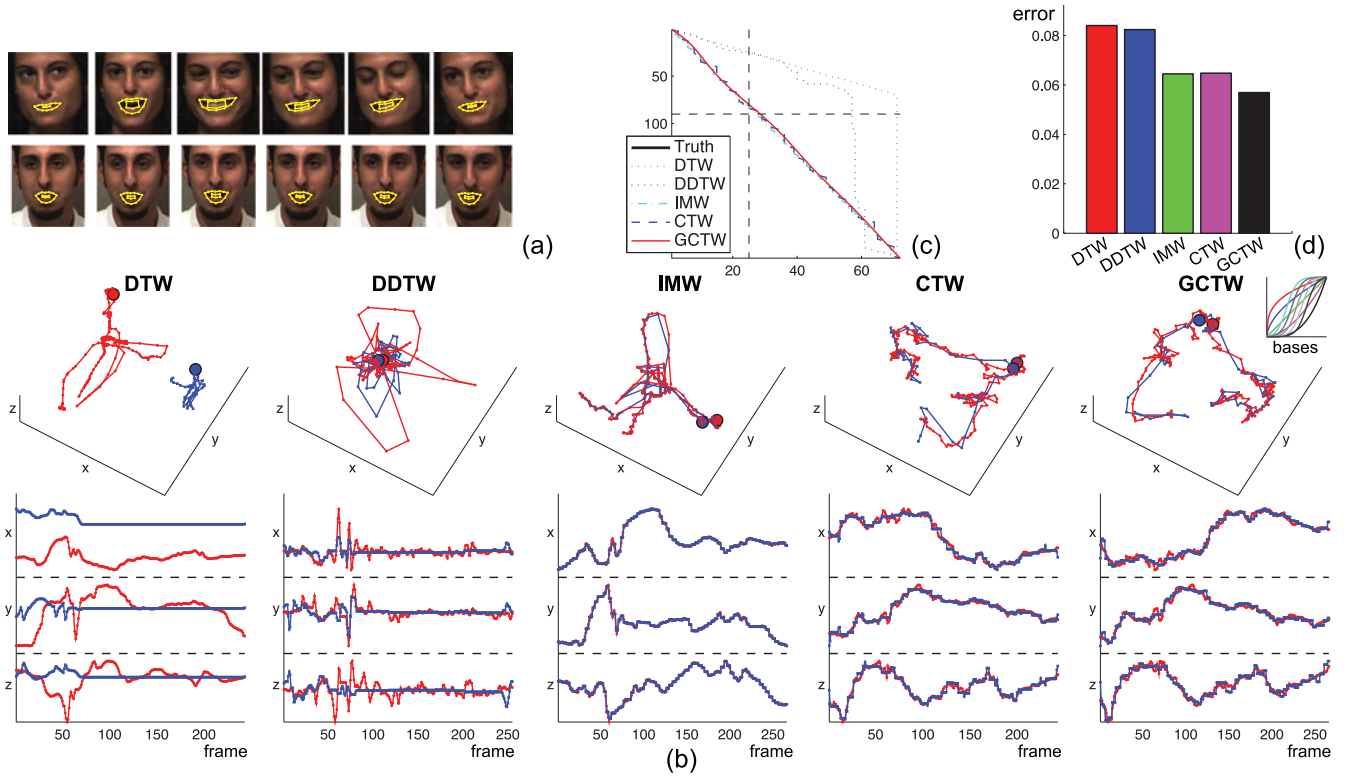

Fig. 10. Comparison of algorithms for aligning facial expression across different subjects. (a) An example of two smiling expression sequences aligned by GCTW. The features of the two sequences are computed as the 18 landmark coordinates of the mouth given by a face tracker. (b) Alignment results. The position that corresponds to the peak of the expression is indicated by points on the curves in the top row. (c) Comparison of time warping paths. The position that corresponds to the peak of the expression is indicated by the intersection of the dashed lines. (d) Alignment errors.

In the second example, we recognized actions from motion capture sequences $\{\mathbf{Y}_j\}_j$ given a training set containing videos with different visual features $\{\mathbf{X}_i\}_i$. From the Weizmann dataset [42] we selected 24 videos containing three actions (walking, running and jumping). From the CMU motion capture database we selected 30 sequences of subjects performing the same three actions (walking, running and jumping). For the mocap data $\mathbf{Y}$, we extracted quaternions on 20 joints [45] and use them as features. Each video frame $\mathbf{X}$ was encoded with a binary silhouette feature. The experimental setting was similar to previous experiments. We divided all sequences into two disjoint sets used for training and testing respectively. Each test motion capture sequence was classified with the label of the video sequence training sequence $\mathbf{X}$ with smallest (dis)similarity (20). As shown in Fig. 9f, GCTW achieved the lowest classification error compared to other methods. This was because our GCTW (dis)similarity captures between multimodal similarity between actions in videos.

5.6 Aligning Facial Expression Sequences

In the fourth experiment, we compared CTW and GCTW in the task of aligning unscripted facial expression sequences. The facial videos were taken from the RU-FACS database [46], which contains digitized videos of 29 young adults. The action units (AUs) in this database have been manually coded, and we randomly cropped video segments containing AU12 (smiling) for our experiments. Each event of AU12 is coded at its peak position. We used a person-specific active appearance model [47] to track 66 landmarks on the face. For the alignment of AU12, we used only the 18 landmarks that correspond to the outline of the mouth. See Fig. 10a for example frames where the mouth outlines are plotted.

The performance of CTW and GCTW were compared with DTW, DDTW and IMW. We initialized IMW, CTW and GCTW using the same uniform warping. Fig. 10b shows the alignment result obtained by different methods, where the three dimensions correspond to the first three principal components of the original signals. As an approximate ground-truth, the position of the peak frame of each AU12 event is indicated as the red and blue points on the curves in Fig. 10b and the intersection of the two dashed lines in Fig. 10c. As we can see from Figs. 10b and 10c, the two peaks in the low-dimensional projection found by CTW and GCTW are closer to the manually labeled peak than the ones in the original space used for DTW and DDTW. Finally, the distance between the peak point and the warping path is computed to quantitatively measure the performance. Fig. 10d shows the average error as the distance normalized by the sequence lengths over 20 random repetitions, where CTW and GCTW achieved better performance.

5.7 Aligning Large-Scale Motion Capture Sequences

This experiment illustrates the benefits of using GCTW for aligning two large-scale motion capture sequences. The two sequences were taken from the CMU multimodal activity dataset [48], which contains multi-sensor recordings (video, audio, motion capture data and accelerometers) of

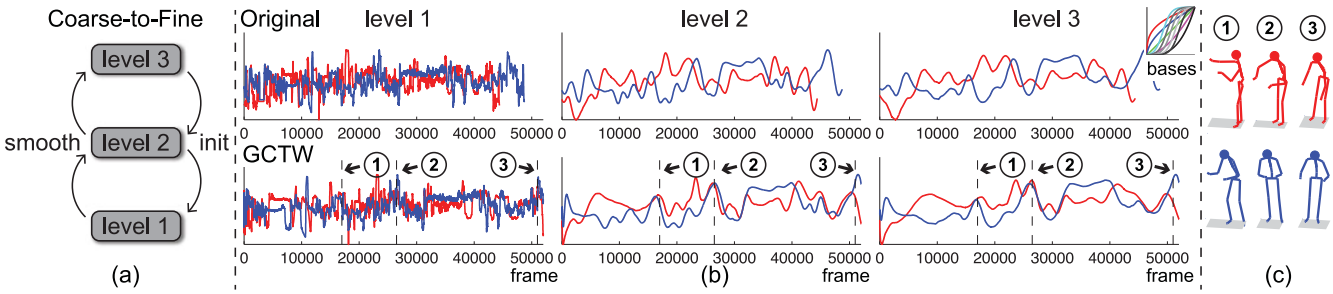

Fig. 11. Aligning two large-scale motion capture sequences using GCTW. (a) A coarse-to-fine strategy for improving the optimization performance of GCTW. (b) The first row shows the first principal components of the original sequences for three levels of the temporal pyramids. The second row corresponds to the aligned sequences using GCTW. (c) Key frames of similar body poses aligned by GCTW.

naturalistic behavior of 40 subjects cooking five different recipes. The two motion capture sequences used in this experiment contain 44,387 and 48,724 frames respectively from two subjects cooking brownies. See Fig. 11c for several key-frames of these two sequences. For each motion capture frame, we computed quaternions in four joints on the right hand, resulting in a 12-D feature vector. In this experiment, we only optimized the temporal component of GCTW, not the spatial component. We used five polynomial and five $\tanh(\cdot)$ functions as monotonic bases for the time warping function.

To avoid local minima in the alignment, we used a temporal coarse-to-fine strategy for the Gauss-Newton optimization in GCTW. As shown in Fig. 11a, the coarse-to-fine strategy proceeds in two steps: (1) In the pre-processing step, we obtained a three-level pyramid for each time series by recursively applying Gaussian smoothing with $\sigma = 200$. For instance, the first row of Fig. 11b illustrates the two sequences in three levels, where the ones in the first level correspond to the original signals, while the ones in the third level contain less detailed but much smoother signals. (2) In the optimization step, GCTW was first used to align the two sequences on the third level instead of the first level. The computed time warping result was then used to initialize GCTW on the second level. We repeated the same procedure to compute the final time warping result of the original sequences on the first level.

For this large-scale example, DTW was slow and computationally expensive. However, GCTW was able to efficiently find the temporal correspondence between the sequences in just a few seconds using Matlab on a regular laptop with a 2.5 GHz Intel CPU. Since the ground-truth is unknown, we qualitatively evaluated GCTW by showing the aligned key frames in Fig. 11c. Although the two subjects spent different amounts of time and followed different procedures in cooking, GCTW was able to align similar right hand actions.

5.8 Detection and Alignment of Similar Sub-Sequences

A major problem of DTW-type techniques is that they require an exact matching between the first frame and the last one, see (5). These boundary conditions are impractical and very restrictive when only a subset of the input sequence is similar to the sequence to be aligned. Fig. 12 illustrates this problem: how can we align four motion capture signals composed by different walking cycles? This problem is related to sub-sequence DTW [49] and temporal commonality discovery [50]. A major limitation of these methods is its inability to handle multiple sequences. This experiment shows how can we use GCTW for multiple sub-sequence alignment.

We selected four walking sequences from the CMU motion capture database [51]. For each motion capture frame, we computed the quaternions for 14 joints on the body, resulting in a 42-D feature vector that describes the human pose. Fig. 12a illustrates the first three principal components of the walking sequences. To allow for sub-sequence alignment, the warping path in GCTW is represented by a combination of a constant function and a linear one as the monotonic bases (see upper-left corner of Fig. 12c). Both GCTW and the baseline mDTW method are initialized by uniformly aligning the sequences.

A visual comparison between mDTW and GCTW is illustrated in Fig. 12. Without any manual cropping, most of the conventional DTW-based methods, such as mDTW, aligned

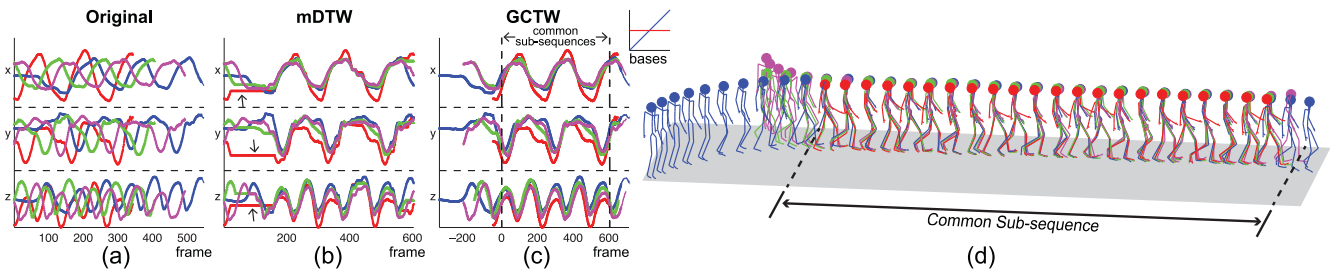

Fig. 12. Aligning similar sub-sequences of four walking motion capture signals. (a) Original features of four mocap walking sequences. (b) Alignment achieved by mDTW. mDTW alings the sequences end-to-end and hence it has to stretch some parts of the sequences (flat lines indicated by arrows). (c) Alignment using GCTW. GCTW efficiently aligns the sub-sequences and also finds the boundaries of the sub-sequences containing similar motions. (d) Key frames aligned by GCTW.

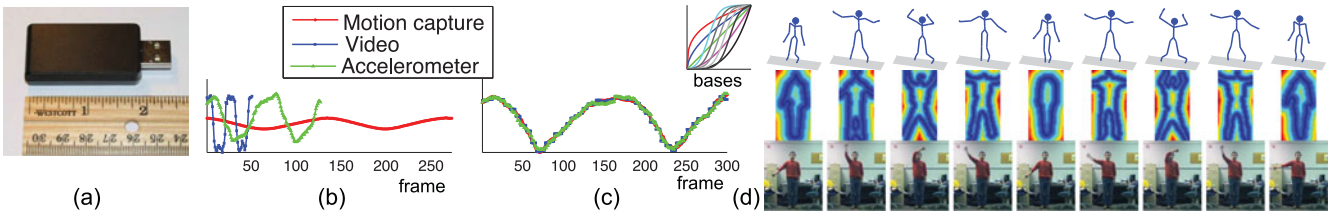

Fig. 13. Example of aligning multi-modal sequences. (a) Accelerometer. (b) Projection onto the first principal component for the motion capture data, video and accelerometers respectively. (c) GCTW. (d) Key frames aligned by GCTW. Notice that similar hand gestures have been aligned. From the top to bottom, we show mocap data, video, and accelerometer data respectively.

the sequences by matching the first and the last frame, which results in incorrect alignments (see Fig. 12b). Some parts (noted by arrows) of the sequences with fewer cycles have to be stretched into flat lines in order to match the other sequences with more cycles. Unlike conventional DTW-based methods built on dynamic programming, GCTW uses the Gauss-Newton method, which allows for a more flexible time warping. By incorporating a constant function in the set of bases, GCTW can naturally be generalized to deal with the sub-sequence alignment problem across multiple sequences. As shown in Figs. 12c and 12d, GCTW is not only able to align the sequences in time, but also locate the boundaries of the sub-sequences that contain similar motions. This experiment demonstrates the benefits of GCTW in controlling the warping path.

5.9 Aligning Multi-Modal Sequences

The last experiment evaluates CTW and GCTW in the task of aligning multi-modal sequences recorded with different sensors. We selected one motion capture sequence from the CMU motion capture database, one video sequence from the Weizmann database [42], and we collected an accelerometer signal of a subject performing jumping jacks. Some instances of the multi-modal data can be seen in Fig. 13d. Note that, to make the problem more challenging, the two subjects in the mocap (top row) and video (middle row) sequences are performing the same activity, but in the accelerometer sequence (bottom row) the subject only moves one hand and not the legs. Even in this challenging scenario, GCTW is able to solve for the temporal correspondence that maximizes the correlation between signals. For the mocap data, we computed the quaternions for the 20 joints. In the case of the Weizmann dataset, we computed the Euclidean distance transform as described earlier. The X, Y and Z axis accelerometer data (40 Hz) was collected using an X6-2 mini USB accelerometer (Fig. 13a). Fig. 13b shows the first components of the three sequences projected separately by PCA. As shown in Fig. 13c, GCTW found an accurate temporal correspondence between the three sequences. Unfortunately, we do not have ground-truth for this experiment. However, visual inspection of the video suggests that the results are consistent with human labeling. Fig. 13d shows several frames that have aligned by GCTW.

6 Conclusions

This paper proposes CTW and GCTW, two new techniques for spatio-temporal alignment of multiple multi-modal time series. CTW extends DTW by adding a feature selection mechanism and enabling alignment of signals with different dimensionality. CTW extends CCA by adding temporal alignment and allowing temporally local projections. To improve the efficiency of CTW, allow a more flexible time-warping, and align multiple sequences, GCTW extends CTW by parameterizing the warping path as a combination of monotonic functions. Inspired by existing work on image alignment, GCTW is optimized using coarse-to-fine Gauss-Newton updates, which allows for efficient alignment of long sequences.

Although CTW and GCTW have shown promising preliminary results, there are still unresolved issues. First, the Gauss-Newton algorithm used in GCTW for time warping converges poorly in areas where the objective function is non-smooth. Second, both CTW and GCTW are subject to local minima. The effect of local minima can be partially alleviated using a temporal coarse-to-fine approach as in the case of image alignment. In future work, we also plan to explore better initialization strategies. Third, although the experiments have shown good results using manually designed bases, an obvious extension is to learn a set of monotonic bases from training data, so the basis is adapted to a particular alignment problem. Finally, kernelization of CTW and GCTW might lead to improvements in alignment error.

Acknowledgments

This work was partially supported by the National Science Foundation under Grant No. EEEC-0540865, RI-1116583 and CPS-0931999. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. F. Zhou is the corresponding author.

References

[1] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
[2] A. Bruderlin and L. Williams, "Motion signal processing," in *Proc. ACM 22nd Annu. Conf. Comput. Graphics Interactive Techn.*, 1995, pp. 97–104.
[3] Y. Caspi and M. Irani, "Aligning non-overlapping sequences," *Int. J. Comput. Vis.*, vol. 48, no. 1, pp. 39–51, 2002.
[4] J. Aach and G. M. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, vol. 17, no. 6, pp. 495–508, 2001.
[5] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 116–125, Jan. 2003.

[6] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, Mar. 2013.
[7] J. M. Winters and Y. Wang, "Wearable sensors and telerehabilitation," *IEEE Eng. Med. Biol. Mag.*, vol. 22, no. 3, pp. 56–65, May-Jun. 2003.
[8] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "View-independent action recognition from temporal self-similarities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 172–185, Jan. 2011.
[9] E. Hsu, K. Pulli, and J. Popovic, "Style translation for human motion," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 1082–1089, 2005.
[10] W. Pan and L. Torresani, "Unsupervised hierarchical modeling of locomotion styles," in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 785–792.
[11] A. Veeraraghavan, A. Srivastava, A. K. R. Chowdhury, and R. Chellappa, "Rate-invariant recognition of humans and their activities," *IEEE Trans. Image Process.*, vol. 18, no. 6, pp. 1326–1339, Jun. 2009.
[12] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[13] F. Zhou and F. De la Torre, "Canonical time warping for alignment of human behavior," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 2286–2294.
[14] F. Zhou and F. De la Torre, "Generalized time warping for alignment of human motion," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2012, pp. 1282–1289.
[15] M. Brand and A. Hertzmann, "Style machines," in *Proc. ACM 27th Annu. Conf. Comput. Graphics Interactive Techn.*, 2000, pp. 183–192.
[16] A. Shapiro, Y. Cao, and P. Faloutsos, "Style components," in *Proc. Graphics Interface*, 2006, pp. 33–39.
[17] C. Rao, A. Gritai, M. Shah, and T. Fathima, "View-invariant alignment and matching of video sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 939–945.
[18] D. Gong and G. G. Medioni, "Dynamic manifold warping for view invariant action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 571–578.
[19] M. A. Nicolaou, V. Pavlovic, and M. Pantic, "Dynamic probabilistic CCA for analysis of affective behavior and fusion of continuous annotations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1299–1311, Jul. 2014.
[20] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proc. SIAM Int. Conf. Data Mining*, 2001, pp. 5–7.
[21] J. Listgarten, R. M. Neal, S. T. Roweis, and A. Emili, "Multiple alignment of continuous time series," in *Proc. Neural Inf. Process. Syst.*, 2005, pp. 817–824.
[22] J. Ham, D. Lee, and L. Saul, "Semisupervised alignment of manifolds," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2005, pp. 120–127.
[23] C. Wang and S. Mahadevan, "Manifold alignment using Procrustes analysis," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1120–1127.
[24] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Hoboken, NJ, USA: Wiley, 2003.
[25] F. De la Torre, "A least-squares framework for component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1041–1055, Jun. 2012.
[26] T. K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1415–1428, Aug. 2009.
[27] C. C. Loy, T. Xiang, and S. Gong, "Time-delayed correlation analysis for multi-camera activity understanding," *Int. J. Comput. Vis.*, vol. 90, no. 1, pp. 106–129, 2010.
[28] B. Fischer, V. Roth, and J. Buhmann, "Time-series alignment by non-negative multiple generalized canonical correlation analysis," *BMC Bioinformatics*, vol. 8, no. 10, pp. 1–10, 2007.
[29] S. Shariat and V. Pavlovic, "Isotonic CCA for sequence alignment and activity recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2572–2578.
[30] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 1995.
[31] S. Chu, E. Keogh, D. Hart, and M. Pazzani, "Iterative deepening dynamic time warping for time series," in *Proc. SIAM Int. Conf. Data Mining*, 2002, pp. 195–212.
[32] A. Heloir, N. Courty, S. Gibet, and F. Multon, "Temporal alignment of communicative gesture sequences," *J. Vis. Comput. Animation*, vol. 17, no. 3-4, pp. 347–357, 2006.
[33] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.
[34] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM Conf. Knowl. Discovery Data Mining*, 2012, pp. 262–270.
[35] M. A. Hasan, "On multi-set canonical correlation analysis," in *Proc. Int. Joint Conf. Neural Netw.*, 2009, pp. 1128–1133.
[36] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, 2nd ed. New York, NY, USA: Springer, 2005.
[37] J. O. Ramsay, "Estimating smooth monotone functions," *J. Roy. Statist. Soc.: Series B Statist. Methodol.*, vol. 60, pp. 365–375, 1998.
[38] T. Robertson, F. T. Wright, and R. L. Dykstra, *Order Restricted Statistical Inference*. New York, NY, USA: Wiley, 1988.
[39] I. W. Wright and E. J. Wegman, "Isotonic, convex and related splines," *Ann. Statist.*, vol. 8, no. 5, pp. 1023–1035, 1980.
[40] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.
[41] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. New York, NY, USA: Wiley, 1998.
[42] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
[43] C. R. Maurer, R. Qi, and V. V. Raghavan, "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 265–270, Feb. 2003.
[44] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the Poisson equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1991–2005, Dec. 2006.
[45] J. Barbic, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proc. Graphics Interface*, 2004, pp. 185–194.
[46] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. Fasel, and J. R. Movellan, "Automatic recognition of facial actions in spontaneous expressions," *J. Multimedia*, vol. 1, no. 6, pp. 22–35, 2006.
[47] I. Matthews and S. Baker, "Active appearance models revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, 2004.
[48] F. De la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel, "Detailed human data acquisition of kitchen activities: The CMU-multimodal activity database (CMU-MMAC)," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. RI-TR-08-22, 2009.
[49] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, "Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation," *Artif. Intell. Med.*, vol. 45, no. 1, pp. 11–34, 2009.
[50] W.-S. Chu, F. Zhou, and F. De la Torre, "Unsupervised temporal commonality discovery," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 373–387.
[51] (2015). Carnegie Mellon University Motion Capture Database. [Online]. Available: http://mocap.cs.cmu.edu

Feng Zhou received the BS degree in computer science from Zhejiang University in 2005, the MS degree in computer science from Shanghai Jiao Tong University in 2008, and the PhD degree in robotics from Carnegie Mellon University in 2014. He is now a researcher in the Media Analytics group at NEC Laboratories America. His research interests include machine learning and computer vision.

Fernando de la Torre received the BSc degree in telecommunications, as well as the MSc and PhD degrees in electronic engineering from the La Salle School of Engineering at Ramon Llull University, Barcelona, Spain in 1994, 1996, and 2002, respectively. He is an associate research professor in the Robotics Institute at Carnegie Mellon University. His research interests are in the fields of computer vision and machine learning. Currently, he is directing the Component Analysis Laboratory (http://ca.cs.cmu.edu) and the Human Sensing Laboratory (http://humansensing.cs.cmu.edu) at Carnegie Mellon University. He has more than 150 publications in referred journals and conferences. He has organized and co-organized several workshops and has given tutorials at international conferences on the use and extensions of Component Analysis.