Canonical Time Warping for Alignment of Human Behavior

Feng Zhou  
Robotics Institute  
Carnegie Mellon University  
www.f-zhou.com

Fernando de la Torre  
Robotics Institute  
Carnegie Mellon University  
ftorre@cs.cmu.edu

Abstract

Alignment of time series is an important problem to solve in many scientific disciplines. In particular, temporal alignment of two or more subjects performing similar activities is a challenging problem due to the large temporal scale difference between human actions as well as the inter/intra subject variability. In this paper we present canonical time warping (CTW), an extension of canonical correlation analysis (CCA) for spatio-temporal alignment of human motion between two subjects. CTW extends previous work on CCA in two ways: (i) it combines CCA with dynamic time warping (DTW), and (ii) it extends CCA by allowing local spatial deformations. We show CTW’s effectiveness in three experiments: alignment of synthetic data, alignment of motion capture data of two subjects performing similar actions, and alignment of similar facial expressions made by two people. Our results demonstrate that CTW provides both visually and qualitatively better alignment than state-of-the-art techniques based on DTW.

1 Introduction

Temporal alignment of time series has been an active research topic in many scientific disciplines such as bioinformatics, text analysis, computer graphics, and computer vision. In particular, temporal alignment of human behavior is a fundamental step in many applications such as recognition [1], temporal segmentation [2] and synthesis of human motion [3]. For instance consider Fig. 1a which shows one subject walking with varying speed and different styles and Fig. 1b which shows two subjects reading the same text.

Previous work on alignment of human motion has been addressed mostly in the context of recognizing human activities and synthesizing realistic motion. Typically, some models such as hidden Markov models [4, 5, 6], weighted principal component analysis [7], independent component analysis [8, 9] or multi-linear models [10] are learned from training data and in the testing phase the time series is aligned w.r.t. the learned dynamic model. In the context of computer vision a key aspect for successful recognition of activities is building view-invariant representations. Junejo et al. [1] proposed a view-invariant descriptor for actions making use of the affinity matrix between time instances. Caspi and Irani [11] temporally aligned videos from two closely attached cameras. Rao et al. [12, 13] aligned trajectories of two moving points using constraints from the fundamental matrix. In the literature of computer graphics, Hsu et al. [3] proposed the iterative motion warping, a method that finds a spatio-temporal warping between two instances of motion captured data. In the context of data mining there have been several extensions of DTW [14] to align time series. Keogh and Pazzani [15] used derivatives of the original signal to improve alignment with DTW. Listgarten et al. [16] proposed continuous profile models, a probabilistic method for simultaneously aligning and normalizing sets of time series.

A relatively unexplored problem in behavioral analysis is the alignment between the motion of the body of face in two or more subjects (e.g., Fig. 1). Major challenges to solve human motion align-
Figure 1: Temporal alignment of human behavior. (a) One person walking in normal pose, slow speed, another viewpoint and exaggerated steps (clockwise). (b) Two people reading the same text.

ment problems are: (i) allowing alignment between different sets of multidimensional features (e.g., audio/video), (ii) introducing a feature selection or feature weighting mechanism to compensate for subject variability or irrelevant features and (iii) execution rate [17]. To solve these problems, this paper proposes canonical time warping (CTW) for accurate spatio-temporal alignment between two behavioral time series. We pose the problem as finding the temporal alignment that maximizes the spatial correlation between two behavioral samples coming from two subjects. To accommodate for subject variability and take into account the difference in the dimensionality of the signals, CTW uses CCA as a measure of spatial alignment. To allow temporal changes CTW incorporates DTW. CTW extends DTW by adding a feature weighting mechanism that is able to align signals of different dimensionality. CTW also extends CCA by incorporating time warping and allowing local spatial transformations.

The remainder of the paper is organized as follows. Section 2 reviews related work on dynamic time warping and canonical correlation analysis. Section 3 describes the new CTW algorithm. Section 4 extends CTW to take into account local transformations. Section 5 provides experimental results.

2 Previous work

This section describes previous work on canonical correlation analysis and dynamic time warping.

2.1 Canonical correlation analysis

Canonical correlation analysis (CCA) [18] is a technique to extract common features from a pair of multivariate data. CCA identifies relationships between two sets of variables by finding the linear combinations of the variables in the first set\(X \in \mathbb{R}^{d_x \times n}\) that are most correlated with the linear combinations of the variables in the second set\(Y \in \mathbb{R}^{d_y \times n}\). Assuming zero-mean data, CCA finds a combination of the original variables that minimizes:

\[J_{\text{cca}}(V_x, V_y) = \| V_x^T X - V_y^T Y \|^2_F \quad \text{s.t.} \quad V_x^T X X^T V_x = V_y^T Y Y^T V_y = I_b, \tag{1}\]

where \(V_x \in \mathbb{R}^{d_x \times b}\) is the projection matrix for \(X\) (similarly for \(V_y\)). The pair of canonical variates \((v_x^T X, v_y^T Y)\) is uncorrelated with other canonical variates of lower order. Each successive canonical variate pair achieves the maximum correlation orthogonal to the preceding pairs. Eq. 1 has a closed form solution in terms of a generalized eigenvalue problem. See [19] for a unification of several component analysis methods and a review of numerical techniques to efficiently solve the generalized eigenvalue problems.

In computer vision, CCA has been used for matching sets of images in problems such as activity recognition from video [20] and activity correlation from cameras [21]. Recently, Fisher et al. [22]...
proposed an extension of CCA with parameterized warping functions to align protein expressions. The learned warping function is a linear combination of hyperbolic functions with non-negative coefficients, ensuring monotonicity. Unlike our method, the warping function is unable to deal with feature weighting.

2.2 Dynamic time warping

Given two time series, $X = [x_1, x_2, \ldots, x_{n_x}] \in \mathbb{R}^{d \times n_x}$ and $Y = [y_1, y_2, \ldots, y_{n_y}] \in \mathbb{R}^{d \times n_y}$, dynamic time warping [14] is a technique to optimally align the samples of $X$ and $Y$ such that the following sum-of-squares cost is minimized:

$$J_{dtw}(P) = \sum_{t=1}^{m} \|x_{p_t^x} - y_{p_t^y}\|^2,$$

where $m$ is the number of indexes (or steps) needed to align both signals. The correspondence matrix $P$ can be parameterized by a pair of path vectors, $P = [p_x^T, p_y^T] \in \mathbb{R}^{2 \times m}$, in which $p_x^T \in \{1 : n_x\}^{m \times 1}$ and $p_y^T \in \{1 : n_y\}^{m \times 1}$ denote the composition of alignment in frames. For instance, the $i^{th}$ frame in $X$ and the $j^{th}$ frame in $Y$ are aligned if there exists $p_t = [p_t^x, p_t^y]^T = [i, j]^T$ for some $t$. $P$ has to satisfy three additional constraints: boundary condition ($p_1 \equiv [1, 1]^T$ and $p_m \equiv [n_x, n_y]^T$), continuity ($0 \leq p_t - p_{t-1} \leq 1$) and monotonicity ($t_1 \geq t_2 \Rightarrow p_{t_1} - p_{t_2} \geq 0$).

Although the number of possible ways to align $X$ and $Y$ is exponential in $n_x$ and $n_y$, dynamic programming [23] offers an efficient ($O(n_x n_y)$) approach to minimize $J_{dtw}$ using Bellman’s equation:

$$L^*(p_t) = \min_{\pi(p_t)} \|x_{p_t^x} - y_{p_t^y}\|^2 + L^*(p_{t+1}),$$

where the cost-to-go value function, $L^*(p_t)$, represents the remaining cost starting at $t^{th}$ step to be incurred following the optimum policy $\pi^*$. The policy function, $\pi : \{1 : n_x\} \times \{1 : n_y\} \rightarrow \{[1, 0]^T, [0, 1]^T, [1, 1]^T\}$, defines the deterministic transition between consecutive steps, $p_{t+1} = p_t + \pi(p_t)$. Once the policy queue is known, the alignment steps can be recursively constructed from the starting point, $p_1 = [1, 1]^T$. Fig. 2 shows an example of DTW to align two 1-D time series.

3 Canonical time warping (CTW)

This section describes the energy function and optimization strategies for CTW.

3.1 Energy function for CTW

In order to have a compact and compressible energy function for CTW, it is important to notice that Eq. 2 can be rewritten as:

$$J_{dtw}(W_x, W_y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_i^x w_j^y \|x_i - y_j\|^2 = \|XW_x^T - YW_y^T\|_F^2,$$

where $W_x \in \{0, 1\}^{m \times n_x}, W_y \in \{0, 1\}^{m \times n_y}$ are binary selection matrices that need to be inferred to align $X$ and $Y$. In Eq. 4 the matrices $W_x$ and $W_y$ encode the alignment path. For instance,
The dimension generalized eigenvalue problem is solved by regularizing the covariance matrices adding a scaled multiple local spatial deformations. This section extends CTW by allowing CTW to align in space and time two time series of different dimensionality. However, there are many situations (e.g., aligning long sequences) where a global transformation of the whole time series is not accurate. For these cases, local models have been shown to provide better performance [3, 24, 25]. This section extends CTW by allowing multiple local spatial deformations.

3.2 Optimization for CTW

Algorithm 1: Canonical Time Warping

```
input : X, Y
output: V_x, V_y, W_x, W_y
begin
  Initialize V_x = I_{d_x}, V_y = I_{d_y}
  repeat
    Use dynamic programming to compute, W_x, W_y, for aligning the sequences, V_x^T X, V_y^T Y
    Set columns of, V^T = [V_x^T, V_y^T], be the leading b generalized eigenvectors of:
    \[
    \begin{bmatrix}
      0 & X W_y^T X^T \\
      Y W_x^T Y^T & 0
    \end{bmatrix}
    \]
    \[ V = \begin{bmatrix}
      X D_x X^T & 0 \\
      0 & Y D_y Y^T
    \end{bmatrix} V A
    \]
  until J_{ctw} converges
end
```

Optimizing $J_{ctw}$ is a non-convex optimization problem with respect to the alignment matrices $(W_x, W_y)$ and projection matrices $(V_x, V_y)$. We alternate between solving for $W_x, W_y$ using DTW, and optimally computing the spatial projections using CCA. These steps monotonically decrease $J_{ctw}$ and since the function is bounded below it will converge to a critical point.

Alg. 1 illustrates the optimization process (e.g., Fig. 3e). The algorithm starts by initializing $V_x$ and $V_y$ with identity matrices. Alternatively, PCA can be applied independently to each set, and used as initial estimation of $V_x$ and $V_y$ if $d_x \neq d_y$. In the case of high-dimensional data, the generalized eigenvalue problem is solved by regularizing the covariance matrices adding a scaled identity matrix. The dimension $b$ is selected to preserve 90% of the total correlation. We consider the algorithm to converge when the difference between two consecutive values of $J_{ctw}$ is small.

4 Local canonical time warping (LCTW)

In the previous section we have illustrated how CTW can align in space and time two time series of different dimensionality. However, there are many situations (e.g., aligning long sequences) where a global transformation of the whole time series is not accurate. For these cases, local models have been shown to provide better performance [3, 24, 25]. This section extends CTW by allowing multiple local spatial deformations.
4.1 Energy function for LCTW
Let us assume that the spatial transformation for each frame in X and Y can be model as a linear combination of \( k_x \) or \( k_y \) bases. Let be \( V_x = [V_1^T, \cdots, V_{k_x}^T]^T \in \mathbb{R}^{k_x \times d_x \times b} \), \( V_y = [V_1^T, \cdots, V_{k_y}^T]^T \in \mathbb{R}^{k_y \times d_y \times b} \) and \( b \leq \min(k_x d_x, k_y d_y) \). CTW allows for a more flexible spatial warping by minimizing:

\[
J_{ctw}(W_x, W_y, V_x, V_y, R_x, R_y)
\]

\[
= \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_i^x w_j^y \left( \sum_{c_x=1}^{k_x} r_i^{c_x} V_{c_x}^T \right) x_i - \sum_{c_y=1}^{k_y} \sum_{j=1}^{n_y} y_j \left( \sum_{c_x=1}^{k_x} \| F_x r_i^{c_x} \|_F^2 + \sum_{c_y=1}^{k_y} \| F_y r_j^{c_y} \|_F^2 \right)
\]

\[
= \left\| V_x \left( 1_{k_x} \otimes X \right) \circ \left( R_x^T \otimes 1_{d_x} \right) \right\|^2_F - \left\| V_y \left( 1_{k_y} \otimes Y \right) \circ \left( R_y^T \otimes 1_{d_y} \right) \right\|^2_F
\]

where \( R_x \in \mathbb{R}^{n_x \times k_x}, R_y \in \mathbb{R}^{n_y \times k_y} \) are the weighting matrices, \( r_i^{c_x} \) denotes the coefficient (or weight) of the \( c_x \)th basis for the \( i \)th frame of X (similarly for \( r_j^{c_y} \)). We further constrain the weights to be positive (i.e., \( R_x, R_y \geq 0 \)) and the sum of weights to be one (i.e., \( R_x 1_{k_x} = 1_{n_x}, R_y 1_{k_y} = 1_{n_y} \)) for each frame. The last two regularization terms, \( F_x \in \mathbb{R}^{n_x \times n_x}, F_y \in \mathbb{R}^{n_y \times n_y} \), are 1st order differential operators of \( r_i^{c_x} \in \mathbb{R}^{n_x \times 1}, r_j^{c_y} \in \mathbb{R}^{n_y \times 1} \), encouraging smooth solutions over time. Observe that \( J_{ctw} \) is a special case of \( J_{ctw} \) when \( k_x = k_y = 1 \).

4.2 Optimization for LCTW

**Algorithm 2: Local Canonical Time Warping**

**Input:** \( X, Y \)

**Output:** \( W_x, W_y, V_x, V_y, R_x, R_y \)

**Begin**

**initialize**

\[ V_x = 1_{k_x} \otimes I_{d_x}, \quad V_y = 1_{k_y} \otimes I_{d_y} \]

\[ r_i^{c_x} = 1 \text{ for } \left( \frac{c_x - 1}{k_x} \right) n_x < i \leq \left( \frac{c_x}{k_x} \right) n_x \]

\[ r_j^{c_y} = 1 \text{ for } \left( \frac{c_y - 1}{k_y} \right) n_y < j \leq \left( \frac{c_y}{k_y} \right) n_y \]

**repeat**

**Denote**

\[ Z_x = (1_{k_x} \otimes X) \circ (R_x^T \otimes 1_{d_x}), \quad Z_y = (1_{k_y} \otimes Y) \circ (R_y^T \otimes 1_{d_y}) \]

\[ Q_x = V_x (I_{k_x} \otimes X), \quad Q_y = V_y (I_{k_y} \otimes Y) \]

**Use dynamic programming to compute,** \( W_x, W_y, \) **between the sequences,** \( V_x \circ Z_x, V_y \circ Z_y \)

**Set columns of,** \( V = [V_x^T, V_y^T] \), **be the leading b generalized eigenvectors,**

\[ \begin{bmatrix} 0 & Z_x W_x Z_x^T \\ Z_y W_y Z_y^T \end{bmatrix} \quad \text{VA} \]

**Set,** \( r = \text{Vec}([R_x, R_y]) \), **be the solution of the quadratic programming problem,**

\[ \min_r \quad r^T \begin{bmatrix} 1_{k_x \times k_x} \otimes D_x \circ Q_x^T Q_x + I_{k_x} \otimes F_x^T F_x & -1_{k_x \times k_y} \otimes W \circ Q_y^T Q_y \\ -1_{k_y \times k_x} \otimes W^T \circ Q_y^T Q_y & 1_{k_y \times k_y} \otimes D_y \circ Q_y^T Q_y + I_{k_y} \otimes F_y^T F_y \end{bmatrix} r \]

**s.t.**

\[ \begin{bmatrix} 1_{k_x} \otimes I_{n_x} \\ 0 \end{bmatrix} \]

**until** \( J_{ctw} \) **converges**

**End**

As in the case of CTW, we use an alternating scheme for optimizing \( J_{ctw} \), which is summarized in Alg. 2. In the initialization, we assume that each time series is divided into \( k_x \) or \( k_y \) equal parts, being the identity matrix the starting value for \( V_{c_x}, V_{c_y} \) and block structure matrices for \( R_x, R_y \).
The main difference between the alternating scheme of Alg. 1 and Alg. 2 is that the alternation step is no longer unique. For instance, when fixing $V_x, V_y$, one can optimize either $W_x, W_y$ or $R_x, R_y$. Consider a simple example in warping $\sin(t_1)$ towards $\sin(t_2)$, one could shift the first sequence along time axis by $\delta t = t_2 - t_1$ or do the linear transformation, $a_t \sin(t_1) + b_t$, where $a_t = \cos(t_2 - t_1)$ and $b_t = \cos(t_1) \sin(t_2 - t_1)$. In order to better control the trade-off between warping and spatial transformation, we propose a stochastic selection process. Let us denote $p_{w|v}$ the conditional probability of optimizing $W$ when fixing $V$. Given the prior probabilities $[p_w, p_v, p_r]$, we can derive the conditional probabilities using Bayes’ theorem and the fact that, $[p_{v|w}, p_{r|v}, p_{w|r}] = 1 - [p_{v|w}, p_{w|v}, p_{w|r}]$. $[p_{v|w}, p_{w|v}, p_{w|r}]^T = \mathbf{A}^{-1}\mathbf{b}$, where $\mathbf{A} = 
abla \mathbf{b} \begin{bmatrix} p_{w} & -p_{v} & 0 \\ p_{w} & 0 & p_{r} \\ 0 & -p_{v} & p_{r} \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 0 \\ p_{w} \\ -p_{v} + p_{r} \end{bmatrix}$. Fig. 3f (right-lower corner) shows the optimization strategy, $p_w = .5, p_v = .3, p_r = .2$, where the time warping process is more often optimized.

5 Experiments

This section demonstrates the benefits of CTW and LCTW against state-of-the-art DTW approaches to align synthetic data, motion capture data of two subjects performing similar actions, and similar facial expressions made by two people.

5.1 Synthetic data

In the first experiment we synthetically generated two spatio-temporal signals (3-D in space and 1-D in time) to evaluate the performance of CTW and LCTW. The first two spatial dimensions and the time dimension are generated as follows: $X = U_x Z M_x^T$ and $Y = U_y Z M_y^T$, where $Z \in \mathbb{R}^{1\times n_y}$ is a curve in two dimensions (Fig. 3a). $U_x, U_y \in \mathbb{R}^{2\times 2}$ are randomly generated affine transformation matrices for the spatial warping and $M_x \in \mathbb{R}^{n_x \times m}, M_y \in \mathbb{R}^{n_y \times m}, m \geq \max(n_x, n_y)$ are randomly generated matrices for time warping. The third spatial dimension is generated by adding a $(1 \times n_x)$ or $(1 \times n_y)$ extra row to $X$ and $Y$ respectively, with zero-mean Gaussian noise (see Fig. 3a-b).

We compared the performance of CTW and LCTW against three other methods: (i) dynamic time warping (DTW) [14], (ii) derivative dynamic time warping (DDTW) [15] and (iii) iterative time warping (IMW) [3]. Recall that in the case of synthetic data we know the ground truth alignment matrix $W_{\text{truth}} = M_x M_y^T$. The error between the ground truth and a given alignment $W_{\text{alg}}$ is computed by the area enclosed between both paths (see Fig. 3g).

Fig. 3c-f show the spatial warping estimated by each algorithm. DDTW (Fig. 3c) cannot deal with this example because the feature derivatives do not capture well the structure of the sequence. IMW (Fig. 3d) warps one sequence towards the other by translating and re-scaling each frame in each dimension. Fig. 3h shows the testing error (space and time) for 100 new generated time series. As it can be observed CTW and LCTW obtain the best performance. IMW has more parameters ($O(dn)$) than CTW ($O(dn)$) and LCTW ($O(kcnb + knl)$), and hence IMW is more prone to overfitting. IMW tries to fit the noisy dimension (3rd spatial component) biasing alignment in time (Fig. 3g), whereas CTW and LCTW have a feature selection mechanism which effectively cancels the third dimension. Observe that the null space for the projection matrices in CTW is $v_x = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, $v_y = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$.

5.2 Motion capture data

In the second experiment we apply CTW and LCTW to align human motion with similar behavior. The motion capture data is taken from the CMU-Multimodal Activity Database [26]. We selected a pair of sub-sequences from subject 1 and subject 3 cooking brownies. Typically, each sequence contains 500-1000 frames. For each instance we computed the quaternions for the 20 joints resulting in a 60 dimensional feature vector that describes the body configuration. CTW and LCTW are initialized as described in previous sections and optimized until convergence. The parameters of LCTW are manually set to $k_x = 3, k_y = 3$ and $p_w = .5, p_v = .3, p_r = .2$.

$^2$The generation of time transformation matrix $M_x$ (similar for $M_y$) is initialized by setting $M_x = I_{n_x}$. Then, randomly pick and replicate $m - n_x$ columns of $M_x$. We normalize each row $M_x 1_m = 1_{n_x}$ to make the new frame to be an interpolation of $z_x$. 

6
Figure 3: Example with synthetic data. Time series are generated by (a) spatio-temporal transformation of 2-D latent sequence (b) adding Gaussian noise in the 3rd dimension. The result of space warping is computed by (c) derivative dynamic time warping (DDTW), (d) iterative time warping (IMW), (e) canonical time warping (CTW) and (f) local canonical time warping (LCTW). The energy function and order of optimizing the parameters for CTW and LCTW are shown in the top right and lower right corner of the graphs. (g) Comparison of the alignment results for several methods. (h) Mean and variance of the alignment error.

Figure 4: Example of motion capture data alignment. (a) PCA. (b) CTW. (c) LCTW. (d) Alignment path. (e) Motion capture data. 1st row subject one, rest of the rows aligned subject two.

Fig. 4 shows the alignment results for the action of opening a cabinet. The projection on the principal components for both sequences can be seen in Fig. 4a. CTW and LCTW project the sequences in a low dimensional space that maximizes the correlation (Fig. 4b-c). Fig. 4d shows the alignment path. In this case, we do not have ground truth data, and we evaluated the results visually. The first row of Fig. 4e shows few instances of the first subject, and the last three rows the alignment of the third subject for DTW, CTW and LCTW. Observe that CTW and LCTW achieve better temporal alignment.
5.3 Facial expression data

In this experiment we tested the ability of CTW and LCTW to align facial expressions. We took 29 subjects from the RU-FACS database [27] which consists of interviews with men and women of varying ethnicity. The action units (AUs) in this database have been manually coded, and we selected AU12 (smiling) to run our experiments. Each event of AU12 is coded with an onset (start), peak and offset (end). We used person-specific AAM [28] to track 66 landmark points on the face. For the alignment of AU12 we only used 18 landmarks corresponding to the outline of the mouth, so for each frame we have a vector \( (\mathbb{R}^{36 \times 1}) \) with \((x, y)\) coordinates.

We took subject 14 and 30 and ran CTW and LCTW on the segments where the AU12 was coded. The parameters of LCTW are manually set to \( k_x = 3, k_y = 3 \) and \( p_w = .5, p_v = .3, p_r = .2 \). Fig. 5 shows the results of the alignment. Fig. 5b-c shows that the low dimensional projection obtained with CTW and LCTW has better alignment than DTW in Fig. 5a. Fig. 5d shows the position of the peak frame as the intersection of the two dotted lines. As we can observe from Fig. 5d, the alignment paths found by CTW and LCTW are closer to the manually labeled peak than the ones found by DTW. This shows that CTW and LCTW provide better alignment because the manually labeled peaks in both sequences should be aligned. Fig. 5e shows several frames illustrating the alignment.

6 Conclusions

In this paper we proposed CTW and LCTW for spatio-temporal alignment of time series. CTW integrates the benefits of DTW and CCA into a clean and simple formulation. CTW extends DTW by adding a feature selection mechanism and enables alignment of signals with different dimensionality. CTW extends CCA by adding temporal alignment and allowing temporal local projections. We illustrated the benefits of CTW for alignment of motion capture data and facial expressions.

7 Acknowledgements

This material is based upon work partially supported by the National Science Foundation under Grant No. EEC-0540865.
References