Self-Similarity Based Time Warping

Christopher J. Tralie

Duke University Department of Mathematics

ctralie@alumni.princeton.edu

Abstract

In this work, we explore the problem of aligning two time-ordered point clouds which are spatially transformed and re-parameterized versions of each other. This has a diverse array of applications such as cross modal time series synchronization (e.g. MOCAP to video) and alignment of discretized curves in images. Most other works that address this problem attempt to jointly uncover a spatial alignment and correspondences between the two point clouds, or to derive local invariants to spatial transformations such as curvature before computing correspondences. By contrast, we sidestep spatial alignment completely by using self-similarity matrices (SSMs) as a proxy to the time-ordered point clouds, since self-similarity matrices are blind to isometries and respect global geometry. Our algorithm, dubbed "Isometry Blind Dynamic Time Warping" (IBDTW), is simple and general, and we show that its associated dissimilarity measure lower bounds the L1 Gromov-Hausdorff distance between the two point sets when restricted to warping paths. We also present a local, partial alignment extension of IBDTW based on the Smith Waterman algorithm. This eliminates the need for tedious manual cropping of time series, which is ordinarily necessary for global alignment algorithms to function properly.

1. Introduction / Background

In this work, we address the problem of synchronizing sampled curves, which we refer to as "time-ordered point clouds" (TOPCs). The problem of synchronizing TOPCs which trace similar trajectories but which may be parameterized differently is usually approached with the Dynamic Time Warping (DTW) algorithm [31, 32]. Since sequential data can often be translated into a sequence of vectors in some feature space, this algorithm has found widespread use in applications such as spoken word synchronization [31, 32], gesture recognition [37], touch screen authentication [12], video contour shape sequence alignment [26], and general time series alignment [5], to name a few of
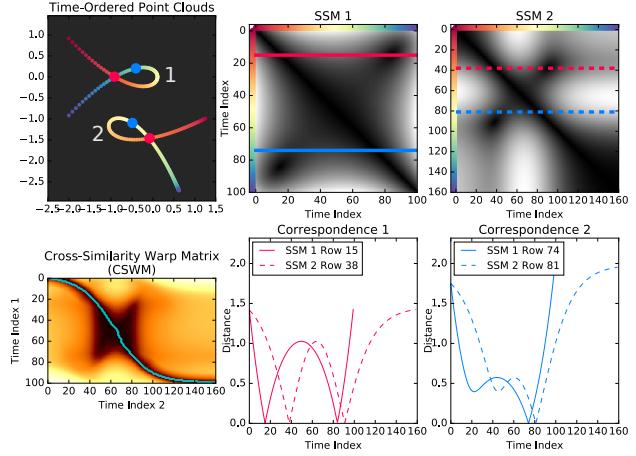

Figure 1. A concept figure for our technique of aligning time-ordered point clouds which are rotated/translated/flipped and re-parameterized versions of each other. Rows of self-similarity matrices (SSMs) of points which are in correspondence are re-parameterized versions of each other, which reduces the global alignment problem to a series of 1D time warping problems. This observation forms the basis of our algorithm, which returns "warping path," drawn in cyan in the lower left plot of this figure, that informs how to synchronize the point clouds.

the thousands of works that use it. The problem becomes substantially more difficult, however, when the point clouds undergo spatial transformations or dimensionality shifts in addition to re-parameterizations, which is more common *across* modalities. For instance, one may want to synchronize a motion capture sequence expressed with quaternions of joints with a video of a similar motion in some feature space (see Section 5.4). There is no apparent correspondence between these spaces a priori. This problem even arises within modalities, such as aligning gestures from different people who reside in different spatial locations. Thus, when synchronizing sampled curves, it is important to address not only re-parameterizations, but also spatial transformations such as maps between spaces or rotations/translations/flips within the same space.

In our work, we avoid explicitly solving for spatial maps by using *self-similarity matrices* (SSMs). Figure 1 shows

a sketch of the technique. Even if the curves have been rotated/translated/flipped and re-parameterized, rows of the SSMs which are in correspondence are re-parameterized versions of each other. Our technique is simple both conceptually and in implementation, it is fully unsupervised, and it is parameter free. There are also theoretical connections between our algorithm and metric geometry, as shown in Section 2.2. We also show an extension of our basic technique to *partially* align time series across modalities (Section 3), and this is the first known solution to that problem. Finally, with proper normalization (Section 4), these techniques can address cross-modal alignment. We show favorable results on a number of benchmark datasets (Section 5).

1.1. Self-Similarity Matrices (SSMs)

The main data structure we rely on in this work is the *self-similarity matrix*. Given a space curve parameterized by the unit interval $\gamma : [0, 1] \to (\mathcal{M}, d)$, a *Self-Similarity Image (SSI)* is a function $D : [0, 1] \times [0, 1] \to \mathbb{R}$ so that

$$D_\gamma(i, j) = d(\gamma(i), \gamma(j)) \quad (1)$$

The discretized version of an SSI corresponding to a sampled version of a curve is a *self-similarity matrix (SSM)*. SSIs and SSMs are naturally *blind to isometries* of the underlying space curve and time-ordered point cloud, respectively; these structures remain the same if the curve/point cloud is rotated/translated/flipped.

Time-ordered SSMs have been applied to the problem of human activity recognition in video [20], periodicity and symmetry detection in video motion [11], musical audio note boundary detection [14], music structure understanding and segmentation [4, 23, 35, 28], cover song identification [39], and dynamical systems [29], to name a few areas. In this work, we study more general properties of time-ordered SSMs that make them useful for alignment.

1.2. Warping Paths / Dynamic Time Warping

The warping path is the basic primitive object we seek to synchronize two time-ordered point clouds. It is a discrete version of an orientation preserving homeomorphism of the unit interval used to re-parameterize curves. In layman's terms, it provides a way to step forward along both point clouds jointly in a continuous way without backtracking, so that they are optimally aligned over all steps. More precisely, given two sets X and Y, a *correspondence* $\mathcal{C}$ between the two sets is such that $\mathcal{C} \subset X \times Y$ and $\forall x \in X \exists y \in y$ s.t. $(x, y) \in \mathcal{C}$ and $\forall y \in Y \exists x \in X$ s.t. $(x, y) \in \mathcal{C}$. In other words, a correspondence is a matching between two sets X and Y so that each element in X is matched to at least one element in Y, and each element of Y is matched to at least one element in X. Let X and Y be two sets whose elements are adorned with a time order:

$X = \{x_1, x_2, ..., x_M\}$ and $Y = \{y_1, y_2, ..., y_N\}$. A *warping path*, between X and Y is a correspondence $\mathcal{W}$ which can be put into the sequence $\mathcal{W} = (c_1, c_2, ..., c_K)$ satisfying the following properties

- *Monotonicity*: If $(x_i, y_j) \in \mathcal{W}$, then $(x_k, y_l) \notin \mathcal{W}$ for $k < i, l > j$

- *Boundary Conditions*: $(x_1, y_1), (x_M, y_N) \in \mathcal{W}$

- *Continuity*: $c_i - c_{i-1} \in \{(0, 1), (1, 0), (1, 1)\}$

Now suppose there are two time-ordered point clouds X and Y which both live in the same metric space $(\mathcal{M}, d)$. The *Dynamic Time Warping (DTW) Dissimilarity*[31, 32][1] between X and Y is

$$\text{DTW}(X, Y) = \min_{\mathcal{W} \in \Omega} \sum_{(i,j) \in \mathcal{W}} d(x_i, y_j)$$

where Ω is the set of all valid warping paths between X and Y. DTW satisfies the following *subsequence relation*

$$DTW_{ij} = \left\{ d(x_i, y_j) + \min \begin{array}{c} DTW_{i-1,j-1} \\ DTW_{i-1,j} \\ DTW_{i,j-1} \end{array} \right\} \quad (2)$$

where DTW_{ij} is the DTW dissimilarity between $\{x_1, x_2, ..., x_i\}$ and $\{y_1, y_2, ..., y_j\}$. This makes it possible to solve DTW with a dynamic programming algorithm which takes $O(MN)$ time. This algorithm computes the cost of the shortest path from the upper left to the lower right of the "cross-similarity matrix" (CSM), or the $M \times N$ matrix holding all distances d_{ij} between X_i and Y_j. A (not necessarily unique) shortest path realizing this distance is a warping path which can be used to align the two time series.

1.3. DTW After Spatial Transformations

In addition to synchronizing curves in the same ambient space which are approximately re-parameterizations of each other, there has also been some recent work on the more difficult problem of matching curves which live in different ambient spaces or which live in the same space but which may differ by a spatial transformation in addition to re-parameterization. One objective for spatial alignment is the optimal *rigid transformation* taking one set of points to another. More precisely, given two Euclidean point clouds $X, Y \in \mathbb{R}^d$, each with N points which are assumed to be in correspondence, the *Procrustes distance* [46, 22] is

$$d_P(X, Y) = \min_{R_x, R_y, t_x, t_y} \sum_{i=1}^{N} ||R_x(x_i - t_x) - R_y(y_i - t_y)||_2^2 \quad (3)$$

---

[1]Note that DTW is not a metric, as it fails to satisfy the triangle inequality. For an example, see [30] section 4.1

One issue with Procrustes is that not only do X and Y have to have the same number of points, but the correspondences must be known a priori. Often in practice, neither of these assumptions are true. To deal with this, one can use the "Iterative Closest Points" (ICP) algorithm [6, 8], which switches back and forth between finding correspondences with nearest neighbors and solving the Procrustes problem. The authors of [49] use a modified version of ICP, replacing the nearest neighbor correspondence step with DTW. This ensures that the time order will be respected, which is not guaranteed with nearest neighbors only[2].

There are also techniques which use canonical correlation analysis (CCA) instead of Procrustes analysis. Given two point clouds with N points represented by matrices $X \in \mathbb{R}^{d_1 \times N}$ and $Y \in \mathbb{R}^{d_2 \times N}$, assumed to be in correspondence, CCA is defined as

$$d_{\text{CCA}} = \min_{V_x \in \mathbb{R}^{d_x \times b}, V_y} ||V_x^T X - V_y^T Y||_F^2 \qquad (4)$$

for some chosen constant $b \leq \min(d_1, d_2)$, s.t. $V_x^T X X^T V_x = V_y^T Y Y^T V_y = I_b$. This is better suited to cross modal applications where scaling is involved. Like Procrustes, this assumes that the correspondences are known a priori. To find the correspondences, the authors in [53] take the same iterative approach as that authors in [49] did with ICP, but they alternate back and forth between DTW and CCA instead of DTW and Procrustes. An updated version of this algorithm known as "generalized time warping" (GTW) [51, 52] was developed which aligns multiple sequences using a single optimization objective where the spatial alignment and time warping are coupled. Finally, a recent work in [40, 41] takes a similar approach, but it replaces CCA by learning features in the projection stage with a deep neural network. Like all supervised learning approaches, however, this method requires training data with known correspondences. Furthermore, all of the techniques we have mentioned so far require a good initial guess to converge to a globally optimal solution.

As an alternative to solving for a spatial alignment explicitly, many works perform time warping on a surrogate function which is invariant to isometries of the input. A popular choice is to numerically estimate curvature [24, 33, 15]. Some works use the triangle area between triples of points as an invariant [1], and some use the turning angle of the curve [9], which is related to curvature. These techniques can suffer from numerical difficulties when estimating the invariants. Also, most of the invariants are local, so small differences can cause the curves "drift" over time (*e.g.* a U is similar to a 6 with local curvature [33]), though using integrated curvature [10] can ameliorate this.

Figure 2. Self-similarity images of different parameterizations of a Figure 8. Rectangles in one image map to rectangles in the other image, and lines in one image map to lines in the other image.

Beyond spatial alignment and invariants, the authors of [47] address more general case of cross-domain object matching (CDOM) with general correspondences and address warping paths as a special case. However, their problem reduces to the quadratic assignment problem, which is NP-hard, and their iterative approximation requires a good initial guess. The authors of [42] address a special case where curves form closed loops, using cohomology to find maps from point clouds to the circle, where they are synchronized, but this only works for periodic time series. The authors of [16] jointly align curves on manifolds, which is effective but requires learning the manifolds. Perhaps the most similar to our approach is the action recognition work of [21], from which we drew much inspiration, which applies DTW to small patches of SSMs to align time warped actions from different camera views. However, they only use elements near the diagonal of SSMs, and their scheme does not extend across modalities.

2. Isometry Blind Dynamic Time Warping

Most of the approaches we reviewed to align time series which have undergone linear transformations try to explicitly factor out those transformations before doing an alignment, but this is not necessary if we build our algorithm on top a self-similarity matrix between two point clouds, which is already blind to isometries. To set the stage for our algorithms, we first study the maps that are induced between self-similarity images by re-parameterization functions, which will help in the algorithm design. For example, take the figure 8 curve, $\gamma_8(t) = (\cos(2\pi t), \sin(4\pi t))$. The bottom left of Figure 2 shows the SSM of a linearly parameterized sampled version of this curve, while the bottom right of Figure 2 shows the SSM corresponding to a re-parameterized sampled version. Maps between the domains of the SSMs shown are always rectangles, and they are independent of underlying curve being parameterized (they only depend on the relationship between two parameterizations). To see this, start with a space curve $\gamma : [0, 1] \to \mathbb{R}^d$ and its resulting self-similarity image D_γ. Given a homeomorphism $h : [0, 1] \to [0, 1]$, which yields a space curve $\gamma_h : [0, 1] \to \mathbb{R}^d$ and a corresponding self-similarity image D_{γ_h}, there is an induced homeomorphism, $h \times h$ from the

---

[2]This analogous to the difference between the Fréchet Distance [2] and the Hausdorff Distance between curves.

square to itself between the two domains of D_γ and D_{γ_h}; that is, $D_{\gamma_h} = D_\gamma(h(s), h(t))$. If we fix a correspondence $s \iff u = h(s)$, then this shows that row $h(s)$ of D_γ is a 1D re-parameterization of row s of D_{γ_h}, making rigorous the observation in Figure 1. Note that for a discrete version of these maps between time-ordered point clouds, one can replace the homeomorphism h with a warping path $\mathcal{W}$, and the relationships are otherwise the same.

2.1. IBDTW Algorithm

We now have the prerequisites necessary to define our main algorithm. The idea is quite simple. Based on our observations and Figure 1 and Figure 2, if we know that point i in a time-ordered point cloud (TOPC) X is in correspondence with a point j in TOPC Y, then we should match the ith row of X's SSM to the jth row of Y's SSM under the L_1 distance, enforcing the constraint that $(i, j) \in \mathcal{W}$. However, since it is unknown a priori which rows should be in correspondence, we try every row i of SSM A against every row j in SSM B, and we create a *cross-similarity time warping matrix (CSWM)* C so that C_{ij} contains L_1 DTW between row i of SSMA and row j of SSMB, *constrained to warping paths which include* (i, j). To enforce that (i, j) be in the optimal warping path, we exploit the boundary condition property of DTW by running the original DTW algorithm twice: once between $SSMAi_{1:i}, SSMBj_{1:j}$ and once between $SSMAi_{i:M}$ and $SSMBj_{j:N}$, summing the costs. After doing this $\forall i, j$, apply the ordinary DTW algorithm to C. Algorithm 1 summarizes this process. Note that a serial implementation of this algorithm takes $O(M^2N^2)$ time, since a 1D DTW is computed for every row pair. To mitigate this, we implement a linear systolic array[50] version of DTW in CUDA. With unlimited parallel processors, this reduces computation to $O(M + N)$. In practice, we witness a 30x speedup between point clouds with hundreds of samples.

Figure 3 shows an example of this algorithm on two rotated/translated/re-parameterized time-ordered point clouds in $\mathbb{R}^2$ (point clouds 1 and 2). As the colors show, IBDTW puts the points into correspondence correctly even without first spatially aligning them. We also show alignment to a third time-ordered point cloud, which is metrically distorted in addition to being rotated/translated/re-parameterized. The returned warping degrades gracefully. We will explore this more rigorously in Section 5.1.

2.2. Analysis: Lower Bounding L1 Metric Stress

IBDTW can be put into the *Gromov-Hausdorff Distance* framework, which describes how to "embed" one metric space into another. More formally, given two discrete metric spaces (X, d_X) and (Y, d_Y), and a correspondence $\mathcal{C}$ between X and Y, the *p-stress* is defined as
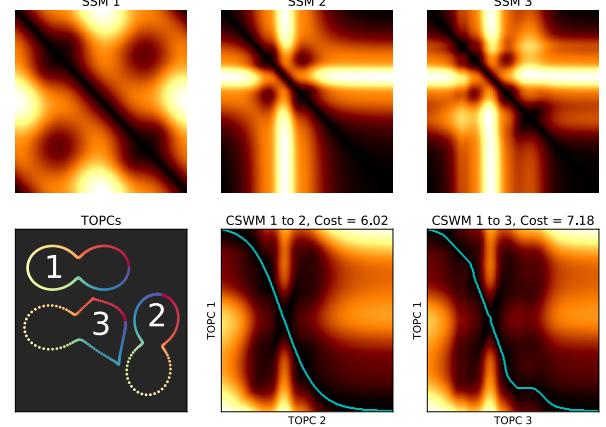

Figure 3. An example of IBDTW between 3 different samplings of a pinched ellipse. The optimal warping path found by Algorithm 1 is drawn in cyan on top of the CSWM in each case. Based on this, points which are in correspondence are drawn with the same color in the lower left figure. Though time-ordered point cloud 2 has more points towards the beginning and fewer points towards the end than time-ordered point cloud 1, correct regions are put into correspondence with each other. Furthermore, in addition to being parameterized this way, the time-ordered point could 3 is also distorted geometrically, but the correspondences are still reasonable.

Algorithm 1 Isometry Blind Dynamic Time Warping

1: **procedure** IBDTW(X, Y, d_X, d_Y)
2:     ▷ TOPCs X and Y with M and N points, metrics d_X and d_Y, respectively
3:     ▷ Initialize cross-similarity warp matrix (CSWM)

4:   C ←
$$\left.\begin{array}{|ccccc|} \hline 0 & \infty & \infty & \dots & \infty \\ \infty & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \infty & 0 & 0 & \dots & 0 \\ \hline \end{array}\right\} M$$
$$\underbrace{}_{N}$$

5:     **for** $i = 1 : M$ **do**
6:         ▷i^{th} row of d_X
7:         $A \leftarrow [d_X(x_i, x_1), d_X(x_i, x_2), \dots, d_X(x_i, x_M)]$
8:         **for** $j = 1 : N$ **do**
9:             ▷j^{th} row of d_Y
10:             $B \leftarrow [d_Y(y_j, y_1), d_Y(y_j, y_2), \dots, d_Y(y_j, y_N)]$
11:             $C_{ij} \leftarrow$ ConstrainedDTW(A, B, L_1, i, j)
12:         **end for**
13:     **end for**
14:     ▷ Use the CSWM C in ordinary DTW
15:     $D \leftarrow \frac{1}{2}\text{DTW}(X, Y, C)$
      return (D, C) ▷ Return the cost and the CSWM
16: **end procedure**

$$S_p(X, Y, \mathcal{C}) = \left(\sum_{(x,y),(x',y')\in\mathcal{C}} (d_X(x, x') - d_Y(y, y'))^p \right)^{1/p}$$

(5)

Intuitively, the p-stress measures how much one has to stretch one metric space when moving it to another. The Gromov-Hausdorff Distance d_{GH} uses $\mathcal{S}_\infty$ specifically:

$$d_{\text{GH}}(X, Y) = \frac{1}{2} \inf_{\mathcal{C} \in \Pi} \mathcal{S}_\infty(X, Y, \mathcal{C}) \qquad (6)$$

where Π is the set of all correspondences between X and Y. In other words, the Gromov-Hausdorff Distance measures the smallest possible *distortion* between a pair of points over all possible embeddings of one metric space into another. Unfortunately, the Gromov-Hausdorff Distance is NP-complete, but we can still connect Algorithm 1 to the Gromov-Hausdorff Distance via the following lemma:

Lemma 1. *The cost returned by Algorithm 1 lower bounds* $\mathcal{S}_1(X, Y, \mathcal{W})$, *or the 1-stress restricted to warping paths.*

Proof: Note that the optimal IBDTW warping path $\mathcal{W}^*$ has the following cost $c(\mathcal{W}^*)$

$$c(\mathcal{W}^*) = \sum_{(x_i, y_j), (x', y') \in \mathcal{W}^*} |d_X(x_i, x') - d_Y(y_j, y')| \quad (7)$$

which can be rewritten as

$$c(\mathcal{W}^*) = \frac{1}{2} \sum_{(x_i, y_j) \in \mathcal{W}^*} \sum_{(x', y') \in \mathcal{W}^*} |d_X(x_i, x'), -d_Y(y_j, y')| \tag{8}$$

since if $(x', y') = (x_i, y_j)$ then the cost is zero, all other terms counted twice. Now fix an x_i and y_j. Then the sum of the terms of the form $|d_X(x_i, x') - d_Y(y_j, y')|$ is simply the L_1 warping distance between 1D time series which are the ith row of d_X, $d_X[i, :]$ and the jth row of d_Y, $d_Y[j, :]$ under the warping $\mathcal{W}^*$. Also, the DTW Distance between $d_X[i, :]$ and $d_Y[j, :]$ is at most the L_1 warping distance under $\mathcal{W}^*$, and is potentially lower since we are computing them greedily only between x_i and y_j, ignoring all other constraints. Hence, the sum of the terms $|d_X(x_i, x') - d_Y(y_j, y')|$ is lower bounded by Line 11 in Algorithm 1. ∎

For a more direct analogy with DTW, Algorithm 1 was designed to lower bound the 1-stress restricted to warping paths. We note that a similar technique could be used to lower bound the Gromov-Hausdorff Distance restricted to warping paths by replacing constrained DTW in the inner loop in Line 11 with a constrained version of the discrete Fréchet Distance [13] to find the maximum distortion induced by putting two points in correspondence. In this work, however, we stick to the 1-stress, since it gives a more informative overall picture of the full metric space.

3. Isometry Blind Partial Time Warping

One of the drawbacks of IBDTW is that it requires a global alignment. However, if the sequences only partially
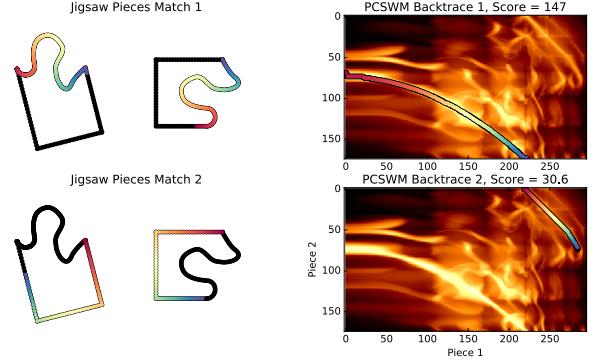

Figure 4. Partial alignment with IBPTW on jigsaw puzzle pieces. The middle row shows the optimal partial alignment. The bottom row shows a locally optimally partial alignment with a lower score. Please refer to color version of this figure for full detail.

overlap, forcing a global alignment leads to poor result, unless manual cropping is done to ensure that sequences start and end at the same place [53]. To automate cropping, we design an isometry blind time warping algorithm that can do partial alignment. This algorithm is like IBDTW, except DTW is replaced with the "Smith Waterman" algorithm [36, 45], which seeks the best contiguous *subsequences* in each time series which match each other [3]. Unlike dynamic time warping, Smith Waterman seeks to *maximize* an alignment *score*, and the alignment does not have to start on the first element of each sequence or end on the last element on each sequence. To solve this, the exact same dynamic programming algorithm is used, except there is one extra "restart" condition if a local alignment has become sufficiently poor. The recurrence is

$$SW_{ij} = \max \left\{ \begin{array}{c} SW_{i-1, j-1} + m(x_i, y_j) \\ SW_{i-1, j} + g \\ SW_{i, j-1} + g \\ 0 \end{array} \right\} \qquad (9)$$

where $m(x_i, y_j)$ is a matching score between points x_i and y_j, which is positive for a match and negative for a mismatch, and g is a gap penalty.

Like DTW, we may modify Smith Waterman to return the best subsequence constrained to match the ith point in the first sequence to the jth point in the second sequence by runing Smith Waterman between $\{X_1, X_2, ..., X_i\}$ and $Y_1, Y_2, ..., Y_j$, and then again between the reversed sequences $\{X_M, X_{M-1}, X_{M-2}, ..., X_i\}$ and $\{Y_N, Y_{N-1}, Y_{N-2}, ..., Y_1\}$. Then, the *Isometry Blind Partial Time Warping (IBPTW)* algorithm is exactly like Algorithm 1, except we replace Line 11 with constrained Smith

---

[3]This algorithm was originally developed for gene sequence alignment, but it has been adapted to multimedia problems such as music alignment [34] and video copyright infringement detection [7].
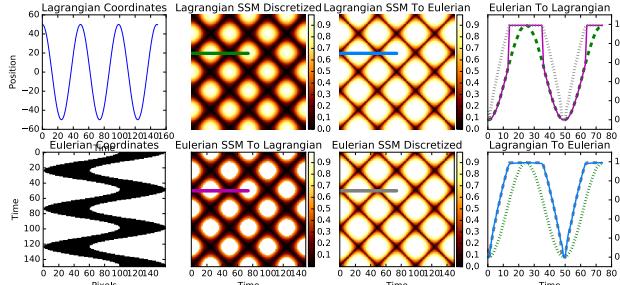
Figure 5. An example of matching the SSM of an oscillating line segment captured with Lagrangian coordinates to an SSM of the oscillating line segment captured with Eulerian coordinates, and vice versa. The right column shows an example of a row from each of the matrices in different cases. The stipple line pattern shows the original row, the line segment shows the corresponding row from the SSM with the target distribution, and the solid row shows the remapped version of the original row. In this case, it is easier to remap the Lagrangian coordinates to Eulerian coordinates, though both remappings are closer to the target than the original.

Waterman, and we replace Line 15 with unconstrained Smith Waterman. We refer to the matrix C (Line 4) as the "partial cross-similarity warp matrix (PCSWM)." In practice, we define $m_1(a, b) = \exp(-|d(a,b)|/\sigma) - 0.6$ for Line 11. If $d(a, b)$ is the L1 distance between elements of two histogram normalized SSM rows, then it ranges between 0 and 1. Thus, there is a positive matching score of up to 0.4 for the most similar SSM values and a negative matching score of -0.6 between the most dissimilar values. We also choose a gap penalty of -0.4 to promote diagonal matches. Otherwise, the warping path maximizing the alignment score contains longer horizontal and vertical lines, leading to undesirable pauses of one time series with respect to the other. For the outer loop (Line 15), we use

$$m_2(a_i, b_j) = \frac{(S_{ij} - \mathrm{md}(S))}{\max(|S - \mathrm{md}(S)|)}$$

where md is the median operation. This will give a high score up to ≤ 1 to row pairs which have a high subsequence score in common, and a low score ≥ -1 to rows which do not have a good subsequence. Figure 4 shows an example of this algorithm on two jigsaw puzzle pieces which should fit together with m_1 and m_2 defined as above, $\sigma = 0.09$, and $g_1, g_2 = -0.4$. The longest subsequence is indeed along the cutouts where they match together. It is also possible to backtrace from anywhere in the PCSWM to find other subsequences which match in common, so Figure 4 also shows an example of suboptimal but good local alignment.

4. Cross-Modal Histogram Normalization

The schemes we have presented work well for point clouds sampled from isometric curves, but the isometry

assumption does not usually hold in cross-modal applications. Not only can the scales be drastically different between modalities, but it is unlikely that a uniform re-scaling will fix the problem. For instance, consider a 1D oscillating bar of length B oscillating sinusoidally over the interval $[0, A]$ with a period of T. Its center position is measured as $c_t = (A/2) + (A/2)\cos(2\pi t/T)$. This type of measurement, which follows the object in question, is in *Lagrangian coordinates*. By contrast, suppose we take a 1D video of the bar with A pixels, where each pixel in each frame measures occupancy by the bar at that frame in the video. Then pixel i in video X_t is parameterized by time as

$$X_t[i] = \left\{ \begin{array}{cc} 1 & |c_t + B/2 - i| < B/2 \\ 0 & \text{otherwise} \end{array} \right\} \quad (10)$$

These pixel by pixel measurements at fixed positions are referred to as *Eulerian coordinates*. Let the Lagrangian SSM D_1 be the 1D metric between two different centers, $D_1[s, t] = |c_s - c_t|$, and let the Eulerian SSM D_2 be the Euclidean metric $D_2[s, t] = ||X_s - X_t||_2$ between each frame of the video. Although they are measuring the same process, the SSMs have a locally different character. Rows of D_1 are perfect sinusoids, while rows of D_2 are more like square waves, since there are sharp transitions from foreground to background in Eulerian coordinates. Figure 5 summarizes all of this visually.

To address this kind of local rescaling between an SSM D_1 and an SSM D_2, we first divide each SSM by its respective max, and we quantize each to L levels evenly spaced in $[0, 1]$. We then apply a monotonic, one-to-one map f to each pixel in D_1 so that the CDF of D_1 approximately matches the CDF of D_2 (see, *e.g.*, [17] ch. 3.3). Note that this process can be done from D_1 to D_2 or from D_2 to D_1, as shown in Figure 5. Since this process is not necessarily symmetric, we perform both sets of normalizations, and we choose the one which yields a better alignment score.

5. Experimental Results

In this section, we will quantitatively compare the IB-DTW algorithm for global alignment with several other techniques in the literature, including ordinary dynamic time warping (DTW), derivative dynamic time warping (DDTW) [24] (a curvature-based version), canonical time warping (CTW)[53], Generalized Time Warping (GTW)[51, 52], and Iterative Motion Warping (IMW)[19] (a simpler version of CTW which is restricted to the same space). We use code from [53] and [51, 52] to compute all of these alignments[4]. We use the default parameters provided in this code, and, as in [53] and [51, 52], we use the results of DTW to initialize CTW and GTW. In all of our experiments, we show results both from IBDTW and
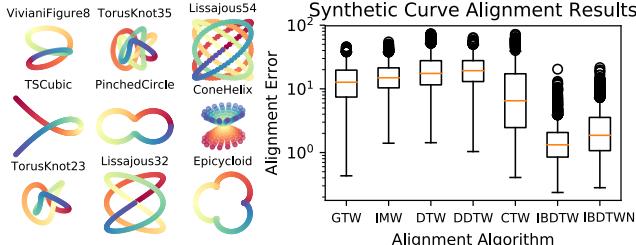

Figure 6. Comparisons of alignment error distributions for different techniques on synthetic rotated/translated/re-paramterized/distorted 2D/3D curves drawn from the classes shown on the left. Log plot shown for contrast since IBDTW performs so well relative to other methods.
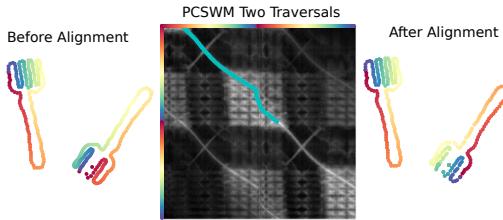

Figure 7. Two closed loops in the shape of a fork which have been rotated/translated/re-parameterized, in addition to starting at different points. The left plot shows the forks before alignment. The center plot shows the PCSWM resulting from aligning both point clouds each repeating themselves twice, as shown by the colors along the rows (left fork) and columns (right fork), which correspond to the colors in the left plot. The optimal partial warping path truncated to the first repetition of the left fork is superimposed in cyan. The forks in the right plot are put into correct correspondence by this truncated partial warping path.

IBDTW after SSM rank normalization, which we refer to as "IBDTWN." When a ground truth warping path exists, we report the alignment error as in [52] and [40]. Given a warping path $\mathcal{W} = (x_1, x_2, ..., x_M)$ and a ground truth path $\mathcal{W}_{GT} = (y_1, y_2, ..., y_N)$, the alignment error is

$$\frac{1}{M+N} \left(\sum_{i=1}^{M} \min_{j=1}^{N} ||x_i - y_j||_2 + \sum_{j=1}^{N} \min_{i=1}^{M} ||x_i - y_j||_2 \right) \tag{11}$$

which is (roughly) the average number of samples by which $\mathcal{W}$ is shifted from $\mathcal{W}_{GT}$ at any point in time.

5.1. Curve Alignment

We first perform an experiment aligning a series of rotated/translated/flipped and re-parameterized sampled curves. As in [52], we re-parameterize the curves with random convex combinations of polynomial, logarithmic, exponential, and hyperbolic tangent functions. To distort the curves, we move random control points in random directions after spatial transformation and re-parameterization. Let X be the TOPC
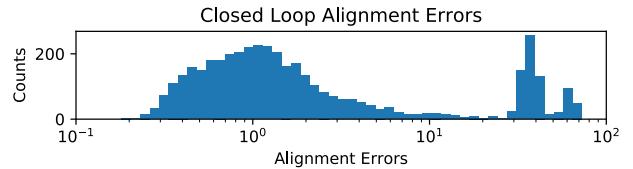

Figure 8. Distribution of IBPTW alignment errors for circularly shifted/warped/distorted MPEG-7 loops.

before transformation/re-paramterization/distortion and let Y be the curve afterwards. The average ratio of $d_{GH}(X, Y)/\text{diam}(X)$, where diam is the "diameter" of X (the maximum inter point distance), is 0.18. Figure 6 shows the results. IBDTW performs the best, while doing the normalization for IBDTWN only degrades the results slightly.

We also showcase our IBPTW algorithm by aligning 2D loops, or curves $\gamma : [0, 1] \to \mathbb{R}^2$ so that $\gamma(0) = \gamma(1)$, which is useful in recognizing boundaries of foreground objects in video. Note that a geometrically equivalent loop can be parameterized starting at a different point in the interior of the first loop: $\gamma'(t) = \gamma(t - \tau(\text{mod}1))$. Also, it is possible that this loop is parameterized differently: $\gamma'_h(t) = \gamma'(h(t))$ for some orientation preserving homeomorphism $h : [0, 1] \to [0, 1]$, and $\gamma'_h(t)$ may also be transformed spatially. To demonstrate how our algorithm is able to align such curves, we use examples from the MPEG-7 dataset of 2D contours [25]. Given a point cloud A and a point cloud B, we partially align the concatenated point clouds AA and BB. If A starts T samples later than B, then there will be a partial warping path which starts at sample 1 in AA and sample T in BB. Figure 7 shows an example where two forks are successfully put into correspondence this way. Figure 8 shows a histogram of alignment errors for distorted/circularly shifted/re-parameterized curves over 7 classes from the MPEG-7 dataset, using $\sigma = 0.01$ and $m_1, m_2 = -0.4$, and with a mean $d_{GH}/\text{diam} = 0.11$. IBPTW returns excellent alignments for most loops, though there are a cluster of outliers that occur due to (near) symmetries of some loops.

5.2. Weizmann Walking Videos

For our first cross modal experiment, we align 4 videos of people walking from the Weizmann dataset [18] cropped to 4 walking cycles each. As in [53], we use different features between each pair of videos we align. On one video, we use the binary mask of the foreground object of the person walking, where every pixel is a dimension, and every frame is a point in the TOPC. On the second video, we use the Euclidean distance transform (EDT) [27]. To assess performance rigorously, we create a more controlled experiment where the second video is the same as the first video after a time warp and applying EDT, so that we have access to the ground truth. Figure 9 shows the results. CTW
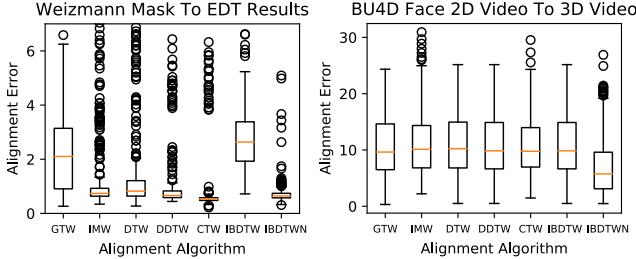
Figure 9. Comparison of IBDTW with different time warping techniques on walking videos from the Weizmann action dataset[18] (left) and on 2D/3D facial expression videos from the BU4D dataset [48] (right).
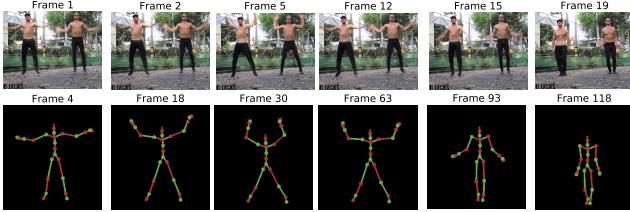

Figure 10. Example aligning MOCAP data expressed in the product space of quaternions with videos in raw pixel space.

performs slightly better than IBDTWN, but not appreciably. Furthermore, IBTW without normalization has the worst average alignment error, demonstrating how normalization is needed in cross-modal applications.

5.3. BU4D 2D/3D Facial Expressions

To show a more difficult cross-modal application, we also synchronize expressions drawn from the BU4D face dataset [48], which includes RGB videos of people making different facial expressions and a 3D triangle mesh corresponding to each frame. For our RGB features, we simply take each channel and each pixel to be a dimension, so a video with $W \times H$ pixels lives in $\mathbb{R}^{3WH}$. For the 3D mesh features, we use shape histograms [3] after mean centering and RMS normalizing each mesh. Our shape histograms have 20 radial shells, each with 66 sectors per shell with centers equally distributed across the sphere. We perform an experiment where we take the 2D features from a face and the 3D features of the same face which has been warped in time. We perform 10 such warpings and alignments for 9 faces from 6 different expression types. Figure 9 shows the aggregated results. The performance is strikingly worse than the Weizmann dataset, though the normalized IBDTW has about half of the alignment error of other techniques.

5.4. Non-Euclidean Examples

One strength of our algorithms is that they run without modification for features in arbitrary metric spaces. For example, Figure 10 shows IBDTW between motion capture data expressed in a product space of quaternions of N joints
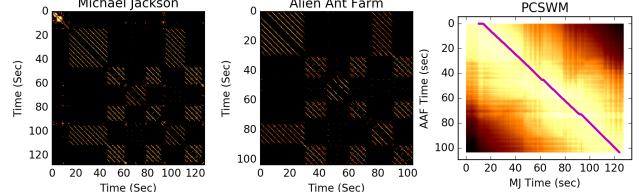

Figure 11. Example aligning "Smooth Criminal" by Michael Jackson to a faster tempo cover by Alien Ant Farm. The optimal warping path is superimposed in magenta over the PCSWM. MJ's version has an intro which is not present in the AAF version, and which is properly skipped by the warping path.

with video data expressed as raw pixels. The product space of quaternions we choose is $\sum_{i=1}^{N} \cos^{-1}(|q_i \cdot p_i|)$ between two sets of quaternions $(q_1, q_2, ..., q_N)$ and $(p_1, p_2, ..., p_N)$. For our second example, Figure 11 shows IBPTW used to align two "cover songs" (two versions of the same song with different instruments / tempos) after fusing note-based and timbral features using similarity network fusion [43, 44] to learn an improved metric for self-similarity (see [38] for more info on this process). Please also refer to supplementary material for video and audio for these examples.

6. Conclusions

In this work, we have shown it is possible to synchronize time-ordered point clouds that are spatially transformed without explicitly uncovering the spatial transformation. As we have shown by our experiments, our algorithms perform excellently when aligning nearly isometric sampled curves, and, with proper normalization, we are competitive with state of the art unsupervised techniques for cross-modal applications. Furthermore, IBDTW requires no parameters, making it approachable "out of the box," while, in our experience, CTW and GTW require many parameters that make or break performance. Finally, we have opened the door for straightforward non-Euclidean time warping, and we hope to see more such applications (*e.g.* time series of graphs).

Acknowledgments

References

[1] N. Alajlan, I. El Rube, M. S. Kamel, and G. Freeman. Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition*, 40(7):1911–1920, 2007. 3

[2] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995. 3

[3] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *International Symposium on Spatial Databases*, pages 207–226. Springer, 1999. 8

[4] J. P. Bello. Grouping recorded music by structural similarity. *Int. Conf. Music Inf. Retrieval (ISMIR-09)*, 2009. 2

[5] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994. 1

[6] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992. 3

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. The video genome. *arXiv preprint arXiv:1003.5320*, 2010. 5

[8] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 3

[9] S. D. Cohen and L. J. Guibas. Partial matching of planar polylines under similarity transformations. In *SODA*, pages 777–786, 1997. 3

[10] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan. Curve matching for open 2d curves. *Pattern Recognition Letters*, 30(1):1–10, 2009. 3

[11] R. Cutler and L. S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000. 2

[12] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 987–996. ACM, 2012. 1

[13] T. Eiter and H. Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994. 5

[14] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000. 2

[15] M. Frenkel and R. Basri. Curve matching using the fast marching method. In *EMMCVPR*, pages 35–51. Springer, 2003. 3

[16] D. Gong and G. Medioni. Dynamic manifold warping for view invariant action recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 571–578. IEEE, 2011. 3

[17] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley world student series. Addison-Wesley, 1992. 6

[18] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007. 7, 8

[19] E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 1082–1089. ACM, 2005. 6

[20] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 293–306. Springer-Verlag, 2008. 2

[21] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):172–185, 2011. 3

[22] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 2

[23] F. Kaiser and T. Sikora. Music structure discovery in popular music using non-negative matrix factorization. In *ISMIR*, pages 429–434, 2010. 2

[24] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2001. 3, 6

[25] L. Latecki. Shape data for the mpeg-7 core experiment ce-shape-1. *Web Page: http://www. cis. temple. edu/˜ latecki/TestData/mpeg7shapeB. tar. g z*, 2002. 7

[26] P. Maurel and G. Sapiro. Dynamic shapes average. 2003. 1

[27] C. R. Maurer, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003. 7

[28] B. McFee and D. P. Ellis. Analyzing song structure with spectral clustering. In *15th International Society for Music Information Retrieval (ISMIR) Conference*, 2014. 2

[29] G. McGuire, N. B. Azar, and M. Shelhamer. Recurrence matrices and the preservation of dynamical properties. *Physics Letters A*, 237(1-2):43–47, 1997. 2

[30] M. Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007. 2

[31] H. Sakoe and S. Chiba. A similarity evaluation of speech patterns by dynamic programming. In *Nat. Meeting of Institute of Electronic Communications Engineers of Japan*, page 136, 1970. 1, 2

[32] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978. 1, 2

[33] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE transactions on pattern analysis and machine intelligence*, 25(1):116–125, 2003. 3

[34] J. Serra, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(6):1138–1151, 2008. 5

[35] J. Serra, M. Müller, P. Grosche, and J. L. Arcos. Unsupervised detection of music boundaries by time series structure features. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012. 2

[36] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981. 5

[37] G. A. ten Holt, M. J. Reinders, and E. Hendriks. Multidimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, volume 300, 2007. 1

[38] C. J. Tralie. Early mfcc and hpcp fusion for robust cover song identification. In *18th International Society for Music Information Retrieval (ISMIR)*, 2017. 8

[39] C. J. Tralie and P. Bendich. Cover song identification with timbral shape. In *16th International Society for Music Information Retrieval (ISMIR) Conference*, 2015. 2

[40] G. Trigeorgis, M. Nicolaou, S. Zafeiriou, and B. Schuller. Deep canonical time warping for simultaneous alignment and representation learning of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3, 7

[41] G. Trigeorgis, M. A. Nicolaou, S. Zafeiriou, and B. W. Schuller. Deep canonical time warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5110–5118, 2016. 3

[42] M. Vejdemo-Johansson, F. T. Pokorny, P. Skraba, and D. Kragic. Cohomological learning of periodic motion. *Applicable Algebra in Engineering, Communication and Computing*, 26(1-2):5–26, 2015. 3

[43] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2997–3004. IEEE, 2012. 8

[44] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3):333–337, 2014. 8

[45] M. S. Waterman. *Introduction to computational biology: maps, sequences and genomes*. CRC Press, 1995. 5

[46] G. Whaba. A least squares estimate of spacecraft attitude. *SIAM Review*, 7(3):409, 1965. 2

[47] M. Yamada, L. Sigal, M. Raptis, M. Toyoda, Y. Chang, and M. Sugiyama. Cross-domain matching with squared-loss mutual information. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1764–1776, 2015. 3

[48] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008. 8

[49] R. Ying, J. Pan, K. Fox, and P. K. Agarwal. A simple efficient approximation algorithm for dynamic time warping. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '16, pages 21:1–21:10, New York, NY, USA, 2016. ACM. 3

[50] C. W. Yu, K. Kwong, K.-H. Lee, and P. H. W. Leong. A smith-waterman systolic cell. In *New Algorithms, Architectures and Applications for Reconfigurable Computing*, pages 291–300. Springer, 2005. 4

[51] F. Zhou and F. De la Torre. Generalized time warping for multi-modal alignment of human motion. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1282–1289. IEEE, 2012. 3, 6

[52] F. Zhou and F. De la Torre. Generalized canonical time warping. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):279–294, 2016. 3, 6, 7

[53] F. Zhou and F. Torre. Canonical time warping for alignment of human behavior. In *Advances in neural information processing systems*, pages 2286–2294, 2009. 3, 5, 6, 7