

# Time Alignment of Trajectories Using Local Time Warping

Guilherme Maeda

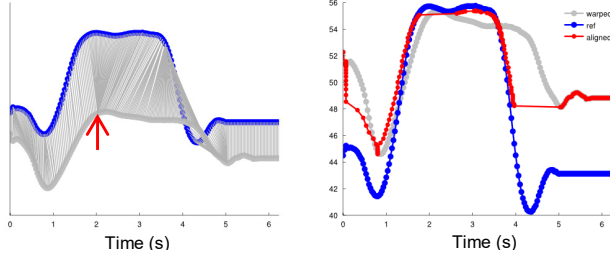


Fig. 1. Vanilla implementation of dynamic time warping applied to very dissimilar trajectories. The plot at the left shows how the time indexes of the reference and warped trajectories relate to each other. Note that for a single index in one trajectory, multiple indexes of the other trajectory are allocated due to the minimum distance. One case is indicated by the arrow at time 2 seconds. The consequence of such repetitive correspondence is that the resulting aligned trajectory may present discontinuities (right).

**Abstract**—This code does time alignment of trajectories by using local optimization.

## I. INTRODUCTION

Time alignment is usually achieved with Dynamic Time Warping (DTW) [1]. As shown in Fig. 1, one issue is that of repeated indexes being attributed to different indexes of the other trajectory. Although DTW provides the global optimal solution with respect to a distance cost, it does not take into account that trajectories generated by dynamical systems must be continuous.

As shown in Fig. 2(a) shows the mapping between the time indexes of two trajectories. At the location indicated by the arrow, the vertical segment indicates the issue of discontinuity, which leads to unnatural alignment. This issue is usually solved with a slope constraint heuristic. In this paper we propose solving this issue by forcing a continuous and smooth warping function as shown in Fig. 2(b). This approach generates a one-to-one association of the time indexes which is naturally fit for local optimization of the function  $f(\theta)$ .

## II. THE ALGORITHM

Consider the mapping  $F : t_{1:T}^r \mapsto t_{1:T}^w$  that maps the time indexes of a reference trajectory with associated time vector  $t_{1:T}^r$  to a query trajectory whose associated time vector  $t_{1:T}^w$  is to be found. Note that one of the trajectories must be resampled such that both query and reference trajectories have the same number of time steps  $T$ . The algorithm does not modify the number of time steps, but the duration

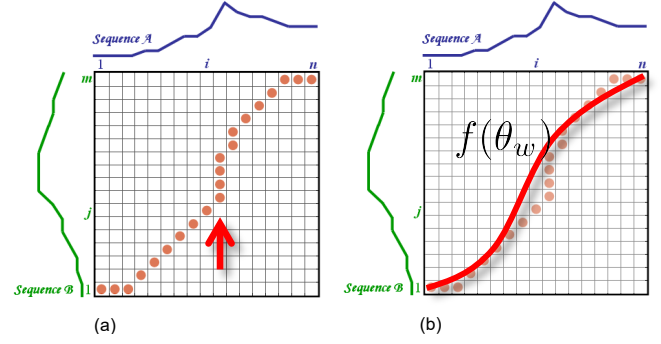


Fig. 2. (a) Index mapping with DTW [2]. The part indicated by the arrow causes discontinuities in the trajectory, which is unnatural for physical dynamical systems. (b) We propose solving the slope problem by using a warping function that is continuous and smooth by construction. The algorithm forces a one-to-one association of indexes which is more suitable for dynamical systems.

between each time step. Define the cumulative cost based on DTW as

$$C = \sum_{t=1}^T \|\mathbf{x}^r(t_t^r) - \mathbf{x}^q(t_t^w)\|, \quad (1)$$

where  $\mathbf{x}^r$  is the reference trajectory for time alignment, and  $\mathbf{x}^q$  is the query trajectory to be aligned.

The goal of the method is to find a warping mapping function  $F$  such that the cumulative cost is minimized. We initially parameterize the warping function with parameters  $\theta = \{\theta_0, \theta_w\}$  such that

$$\mathbf{s}_{1:T} = f(\theta_w), \quad (2)$$

and

$$\mathbf{t}_{1:T}^w = \theta_0 + \mathbf{s} \odot \mathbf{t}_{1:T}^r. \quad (3)$$

The function  $f(\theta_w)$  is the local warping function, previously illustrated in Fig. 2(b) and returns a vector of scaling values for each time step of the original reference time. In (3), the second term is the element wise product of  $\mathbf{s}_{1:T}$  and the reference time  $\mathbf{t}_{1:T}^r$ .

The scalar parameter  $\theta_0$  changes the start instant of  $\mathbf{t}_{1:T}^w$ . This is useful when both reference and query trajectories are identically in shape, but start at different times. In the particular case when the reference and query trajectories are already aligned,  $\theta_0 = 0$  and (2) returns a vector of ones. This is usually the standard initial guess of parameters to initialize the optimization.

To enforce that  $f(\theta_w)$  is a smooth function, we use the

parameterization

$$\mathbf{s}_t = \sum_{n=1}^N (\Phi_t)_n (\theta_w)_n. \quad (4)$$

where  $(\Phi_t)_n$  is the value of the  $n$ -th radial basis function at time  $t$ . The dimension of  $\theta_w \in \mathbb{R}^N$  is defined by the number of  $N$  basis functions to be used.

The method consists in using a local optimization method, such as gradient descent, to update  $\theta = \{\theta_0, \theta_w\}$  such that the cost (1) is minimized. In the current implementation, gradient descent with Jacobian computed by finite differences is used. The source code with an example can be found in

### III. EXPERIMENTS

### IV. ACKNOWLEDGMENTS

### REFERENCES

- [1] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978.
- [2] "DTW algorithm," <http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm>, accessed: 2016-06-01.
- [3] C. Bishop, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.