

The Globally Optimal Reparameterization Algorithm: an Alternative to Fast Dynamic Time Warping for Action Recognition in Video Sequences

Thomas W. Mitchel*, Sipu Ruan*, Yixin Gao, Gregory S. Chirikjian

Abstract—Signal alignment has become a popular problem in robotics due in part to its fundamental role in action recognition. Currently, the most successful algorithms for signal alignment are Dynamic Time Warping (DTW) and its variant ‘Fast’ Dynamic Time Warping (FastDTW). Here we introduce a new framework for signal alignment, namely the Globally Optimal Reparameterization Algorithm (GORA). We review the algorithm’s mathematical foundation and provide a numerical verification of its theoretical basis. We compare the performance of GORA with that of the DTW and FastDTW algorithms, in terms of computational efficiency and accuracy in matching signals. Our results show a significant improvement in both speed and accuracy over the DTW and FastDTW algorithms and suggest that GORA has the potential to provide a highly effective framework for signal alignment and action recognition.

I. INTRODUCTION

With the recent emergence of new machine learning techniques, there has been an increasing interest in robotic action recognition. The foundation of action recognition lies in the problem of signal alignments, in the sense that prior to categorizing or identifying sets of sequences, one must establish a method to temporally parameterize the sequences that enables standardized comparisons.

Currently the most successful techniques for signal alignment are based on the well-known method of *Dynamic Time Warping* (DTW) [1], which matches two time series with a monotonically increasing optimal warping path satisfying boundary conditions. Since its introduction almost 40 years ago, DTW has been applied to a variety of fields including speech recognition [2], action recognition [3], data mining [4], and motion perception [5]. Due to DTW’s $O(T^2)$ time complexity, many variants have been introduced with the goal of striking a balance between accuracy and computational efficiency [6]–[8], the most widely-used of these being the *Fast Dynamic Time Warping* (FastDTW) algorithm, which achieves a time complexity of $O(T)$ [7]. One of the most recent developments in the DTW family has been the introduction of *Generalized Time Warping* [9], which aligns multiple multi-modal sequences with linear time complexity.

Recently, a novel, alternative mathematical framework for signal alignment has been proposed, in which signals are reparameterized to a *universal standard timescale* (UST)

using principles of variational calculus [10]. The goal of this paper is to introduce an efficient numerical algorithm for signal alignment based on this framework, which we will henceforth refer to as the *Globally Optimal Reparameterization Algorithm* (GORA), and to provide an initial numerical validation of this approach.

Given two or more time-evolving signal sequences, GORA temporally reparameterizes each to a UST that allows for pairwise comparison at each instance in time. Reparameterizations are found via variational calculus to produce mappings to new temporal variables that globally minimize the amount of change in the sequences, representing a new approach to the problem of signal alignment. The major advantages of this approach are:

1. *It achieves linear time complexity of $O(T)$, where T is the number of time instances in a signal;*
2. *It can simultaneously reparameterize multiple signal sequences to a universal time scale; and*
3. *It can potentially be built-upon to allow for the effects of nuisance parameters such as noise or motion artifacts to be minimized or eliminated [10].*

The remainder of the paper is organized as follows. First, we review GORA’s mathematical foundations, as described in [10], and define and introduce GORA itself. We then discuss the settings for its application to signals in the form of both real trajectories and video sequences. This is followed by a numerical verification of GORA’s ability to find the globally optimal temporal reparameterization of a given signal. We then provide an initial verification of the algorithm by comparing its performance, in terms of both computational efficiency and accuracy in matching signals, relative to DTW methods. Our results show a significant improvement in both speed and accuracy over the DTW and FastDTW algorithms. We conclude with a short discussion on the computational significance of the differences between GORA and DTW methods, in addition to the authors’ plans for the continued development of the GORA framework.

II. PROBLEM STATEMENT

Without loss of generality, any kind of temporally evolving signal, $X(t)$, can be considered as a mapping from the unit interval to the space S on which that particular type of signal evolves, i.e. $X : [0, 1] \rightarrow S$. Defining a metric d on S , (S, d) becomes a metric space. In general, given any two signals $X_1(t)$, $X_2(t)$ it is likely that

$$\int_0^1 d(X_1(t), X_2(t)) dt \gg 1,$$

*Equally contributing authors

Thomas W. Mitchel, Sipu Ruan and Gregory S. Chirikjian are with the Laboratory of Computational Sensing and Robotics, Johns Hopkins University, Baltimore, Maryland 21218 {tmitchel, ruansp, gchirikj1}@jhu.edu

Yixin Gao is with the Department of Computer Science, Johns Hopkins University, Baltimore, Maryland 21218

even if it is suspected that both signals portray similar dynamic phenomena, a major reason being that each signal could have a different temporal parameterization on the the unit interval.

The GORA algorithm is based on the notion that the temporal misalignment between two arbitrary signals can be compensated for by reparameterizing each to a UST. In other words, assuming that nuisance parameters or motion artifacts such as variations in perspective are not significantly affecting the signals, if we can find two strictly monotonically increasing functions, $\tau_1^*, \tau_2^* \in C^1[0, 1]$ such that

$$\int_0^1 d(X_1(\tau_1^*(t)), X_2(\tau_2^*(t))) dt \ll 1,$$

then we can say that $X_1(t)$ and $X_2(t)$ are fundamentally the same.

Let \mathcal{T} represent the set of all such C^1 monotonically increasing functions on the unit interval. Denoting \circ as the operation of composition of functions, namely

$$(\tau_1 \circ \tau_2)(t) = \tau_1(\tau_2(t)) \quad \forall \tau_1, \tau_2 \in \mathcal{T},$$

(\mathcal{T}, \circ) forms a group, which we refer to as the *temporal reparameterization group (TRG)*. For a given signal, $X(t) \in S$, one can use the succeeding variational calculus formulation to find a globally optimal $\tau^* \in \mathcal{T}$ such that $X(\tau^*(t))$ is the UST parameterization of X , reducing a search for this mapping from S to the quotient space S/\mathcal{T} .

III. MATHEMATICAL FORMULATIONS

Suppose one wants to find a function, $\mathbf{x}(t)$, that extremizes a functional of the form

$$J = \int_0^1 f(\mathbf{x}, \dot{\mathbf{x}}, t) dt \quad (1)$$

where $\dot{\mathbf{x}} = d\mathbf{x}/dt$. This type of problem can be addressed by the application of Calculus of Variations, and the desired $\mathbf{x}(t)$ is the solution to the Euler-Lagrange equations:

$$\frac{\partial f}{\partial \mathbf{x}} - \frac{d}{dt} \left(\frac{\partial f}{\partial \dot{\mathbf{x}}} \right) = 0 \quad (2)$$

In general, there are no guarantees that the solution to the preceding equations will be globally optimal, however, in certain situations (including optimal temporal reparameterization), the structure of the function $f(\cdot)$ will guarantee that the solution generated by the equations is in fact a globally optimal solution. The following theorem is an example of one such case.

A. Theorem and Proof of Global Optimality

THEOREM 1: If $x : [0, 1] \rightarrow \mathbb{R}$ and the integrand in the cost functional (1) is of the form

$$f(x, \dot{x}) = \dot{x}^2 \mathbf{g}(x) \quad (3)$$

where $\mathbf{g}(x) : \mathbb{R} \rightarrow \mathbb{R}_{>0}$ is C^1 , then the solution generated by (2) subject to the boundary conditions $x(0) = 0$ and $x(1) = 1$ is globally minimal.

The proof of this theorem was first demonstrated in [10], however we choose to re-demonstrate it here as it illuminates the fundamental structure of GORA.

Proof: Evaluating (2) with (3) gives

$$2\dot{x}\mathbf{g} + \dot{x}^2 \frac{\partial \mathbf{g}}{\partial x} = 0. \quad (4)$$

Multiplying both sides by \dot{x} and integrating yields the exact differential

$$\frac{d}{dt}(\dot{x}^2 \mathbf{g}) = 0.$$

Integrating both sides with respect to t and isolating \dot{x} yields

$$\dot{x} = c \mathbf{g}^{-\frac{1}{2}}(x)$$

where c is the arbitrary constant of integration. With the boundary conditions $x(0) = 0$ and $x(1) = 1$, we can then write

$$F(x^*) \doteq \frac{1}{c} \int_0^{x^*} \mathbf{g}^{\frac{1}{2}}(\sigma) d\sigma = t,$$

where

$$c = \int_0^1 \mathbf{g}^{\frac{1}{2}}(\sigma) d\sigma.$$

The notation x^* indicates that this is the unique solution obtained from the Euler-Lagrange equations that satisfies the boundary conditions.

The function $F(x^*) = t$ can be inverted (F is monotonically increasing since $\mathbf{g}(x) > 0$) to yield $x^* = F^{-1}(t)$.

To see that this solution is globally optimal, substitute

$$\dot{x}^* = \mathbf{g}^{-\frac{1}{2}}(x^*) \int_0^1 \mathbf{g}^{\frac{1}{2}}(\sigma) d\sigma \quad (5)$$

into the cost functional

$$J(y) = \int_0^1 \mathbf{g}(y) \dot{y}^2 dx$$

where $y(t)$ is any function in \mathcal{T} . Then

$$J(x^*) = \left(\int_0^1 \mathbf{g}^{\frac{1}{2}}(x^*) dx^* \right)^2 = \left(\int_0^1 \mathbf{g}^{\frac{1}{2}}(y) dy \right)^2,$$

where the second equality is simply a change of the name of the dummy variable of integration. Furthermore, since x^* and y are both functions of time, we can change the domain of integration as

$$J(x^*) = \left(\int_0^1 \mathbf{g}^{\frac{1}{2}}(y(t)) \dot{y} dt \right)^2.$$

Since in general, from the Cauchy-Schwarz inequality,

$$\left(\int_0^1 f(t) dt \right)^2 \leq \int_0^1 [f(t)]^2 dt,$$

we see that by letting $f(t) = \mathbf{g}^{\frac{1}{2}}(y(t)) \dot{y}$ that

$$\left(\int_0^1 \mathbf{g}^{\frac{1}{2}}(y(t)) \dot{y} dt \right)^2 \leq \int_0^1 \mathbf{g}(y) (\dot{y})^2 dt$$

and hence

$$J(x^*) \leq J(y)$$

where $x^*(t)$ is the solution generated by the Euler-Lagrange equation and $y(t)$ is any function in \mathcal{T} . Therefore $x^*(t)$ is a globally minimal solution. \blacksquare

B. The Globally Optimal Reparameterization Algorithm (GORA)

In the context of signal alignment, the solution to the preceding variational problem provides a method for finding the UST parameterization of a given signal. In particular, taking $x \in \mathcal{T}$ we have $x^* = \tau^*$, subject to the definition of $g(x)$, which measures the rate of change of the given signal along the temporal axis. This is the backbone of Globally Optimal Reparameterization Algorithm (GORA), defined in Algorithm 1.

Algorithm 1: Globally Optimal Reparameterization Algorithm (GORA)

Input : Input signal $X(t)$; Initial temporal variable t
Output: UST reparameterization of signal $X^*(t)$

- 1 Compute $g(t)$;
- 2 $c = \text{NumericalIntegration}(g^{\frac{1}{2}}(\sigma), [0, 1])$;
- 3 $F(\tau^*) = \frac{1}{c} \text{NumericalIntegration}(g^{\frac{1}{2}}(\sigma), [0, \tau^*])$;
- 4 $\tau^*(t) = F^{-1}(t)$;
- 5 $X^*(t) = \text{Interpolation}(X(t), \tau^*(t))$;

Given $g(t)$, calculating $\tau^*(t)$ is relatively straightforward and follows the first part of the proof of Theorem 1. For a given signal, the function $g(t)$ should be defined analogous to the squared magnitude of the temporal derivative of the signal. For example, in the case of a video signal, an appropriate definition of $g(t)$ could be based on the temporal derivatives of the matrices of pixel values representing each frame.

It should be noted that steps 2-3 in GORA can be performed simultaneously. Additionally, the method of interpolation through which $X^*(t)$ is recovered from $X(t)$ and $\tau^*(t)$ in step 5 should be chosen based on the properties of the input signal.

IV. NUMERICAL VERIFICATION SETTINGS

In this paper, we provide a validation of GORA using discretized signals, in the form of both synthetically generated trajectories in \mathbb{R}^3 and video sequences from the Weizmann Action Recognition Classification Database [11], [12]. The following section describes our experimental regime and results.

A. Signal structure

The version of GORA implemented in our experiments is designed for signals in the form of real trajectories. Through vectorization, each frame of a video sequence can be represented as an $n \times 1$ array of pixel values where $n = \text{width} \times \text{height}$. As such, any video sequence can be

described by a temporally-evolving curve in \mathbb{Z}^n . If we imagine video sequences as collections of discretized samplings of continuous phenomena at arbitrary time instances, any resampling at new time instances produces a curve in \mathbb{R}^n .

Additionally, it is important to note that for the sake of sampling consistency, we trimmed all video sequences in the Weizmann Database such that each trimmed video showed only a *single instance of an action being performed*. For example, videos of a person walking were trimmed to show only a single stride (two successive placements of the same foot) and videos of a person waving multiple times were trimmed to show only a single wave.

B. Formulation of $g(t)$

For signals of the form $X(t) \in \mathbb{R}^n$, a natural choice for the definition of $g(t)$ consistent with (3) is

$$g(t) = \left\| \frac{dX}{dt} \right\|^2, \quad (6)$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector. In practice, we computed dX/dt using a high order finite difference method.

C. Error metric

Given two signals $X_1(t), X_2(t) \in \mathbb{R}^n$ sampled at discrete time instances $\{t_i\}$, we defined the distance or error between them as the average euclidean distance over all time instances, namely,

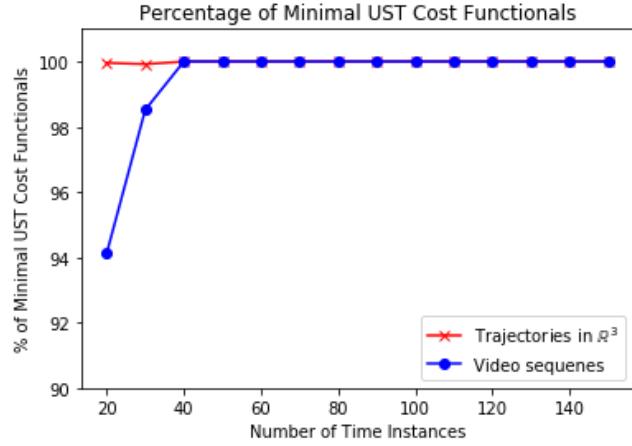
$$\text{error}_{\mathbb{R}^n} = \frac{1}{T} \sum_{i=1}^T \|X_1(t_i) - X_2(t_i)\|, \quad (7)$$

where T is the number of time instances.

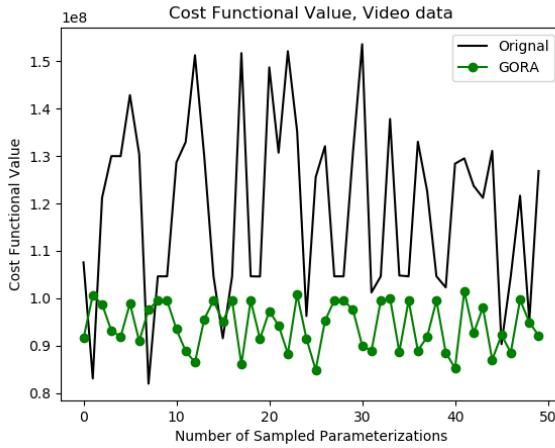
V. VERIFICATIONS OF GLOBAL OPTIMALITY

One of the major advantages of GORA is its ability to reparameterize multiple signals to their corresponding USTs in parallel, which allows for pairwise comparison between signals. Here we verify the global optimality of UST parameterizations computed by GORA using both synthetic curves in \mathbb{R}^3 and video sequences in the form of vectorized curves in \mathbb{R}^n .

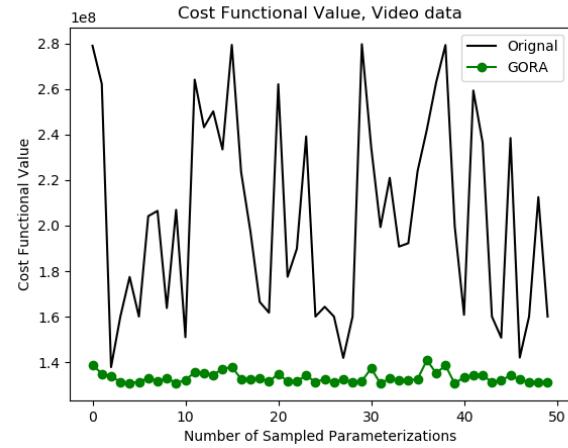
In principle, the UST reparameterization found by GORA corresponding to an arbitrary input signal minimizes the integrand in (1) with respect to the cost function given by (3). Our experimental procedures for validation are summarized as follows: For a given number of time instances, we randomly selected 50 template signals. For each template signal, $X_0(t)$, we randomly generated 50 functions in the TRG and reparameterized $X_0(t)$ with respect to each to create 50 different input signals. We then used GORA to obtain the UST parameterization, $\tau^*(t)$, and recover the the UST reparameterization of each input signal, $X^*(t)$. We then computed the value of the cost functional with respect to $\tau^*(t)$ and $X^*(t)$ and compared this with the value of the cost functional computed using the input signal and original timescale.



(a) Percentages of UST cost functional values lower than initial cost functional values for increasing numbers of time instances in input signals.



(b) The cost functional values for 50 different initial parameterizations of a template video sequence with 20 times instances and their corresponding UST reparameterizations.



(c) The cost functional values for 50 different initial parameterizations of the same template video sequence with 100 times instances and their corresponding UST reparameterizations.

Fig. 1: Comparisons between the cost functional values of randomly parameterized input signals and their UST reparameterized counterparts found by GORA.

The results of our global optimality experiments are displayed in Fig. 1. Fig. 1a shows the percentages of UST computed cost functional values lower than the cost functional values computed using the initial signals and initial timescales from 20 to 150 time instances. At each time instance, the percentages were computed with respect to the 2500 (50×50) input and corresponding UST signal pairs generated using the experimental procedures detailed in the preceding paragraph. The red and blue lines show the results for synthetic trajectories in \mathbb{R}^3 and for vectorized video sequences, respectively.

As an example, Figs. 1b and 1c show the values of the cost functionals for fifty different pairs of randomly parameterized input signals and their corresponding UST reparameterizations found by GORA with 20 and 100 time instances, respectively. For both Figs. 1b and 1c, all input sig-

nals were generated from a single template video sequence in the Weizmann Database. The black and green lines represent the cost functional values for the input signals and their UST reparameterizations, respectively.

The results indicate that in general, GORA does a remarkably good job of finding UST parameterizations that are globally minimal (or at least very close, depending on numerical precision) in the sense of (1). GORA's failure to do so consistently in the case of signals with low numbers of time instances (e.g. Fig. 1b) can likely be explained by its reliance on numerical differentiation of the input signal (in the computation of $g(t)$) and on numerical integration of $g(t)$, both of which become less accurate with larger temporal step sizes corresponding to lower numbers of time instances.

Additionally, this type of numerical evaluation between

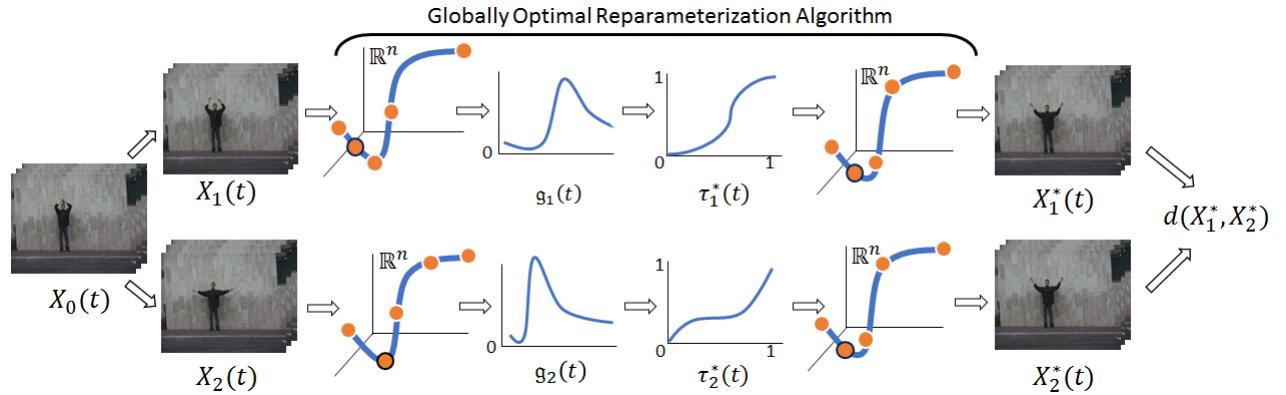


Fig. 2: The experimental work flow for evaluating GORA’s computational efficiency and accuracy in matching signals: A template signal is selected and parameterized with respect to two randomly generated functions in the TRG to create two input signals, represented by curves in \mathbb{R}^n . For each input signal, $\tau^*(t)$ is computed from $g(t)$ and used to recover the UST parameterization of the sequence. The optimally reparameterized signals can then be compared element-wise using the error metric in (7).

the values of the UST computed cost functionals and cost functionals computed using the initial sequences and timescales might provide a template for finding a lower bound of GORA’s effectiveness with a given signal type. When the UST parameterization found by GORA is clearly not globally optimal, as is the case when the cost functional computed with respect to the input signal has a lower value than that computed using the UST reparameterization, it cannot be considered to be accurate. When using GORA for pairwise signal comparisons, failure to well-approximate UST parameterizations would likely lead to a greater degree of induced error. Depending on the properties of the input signals and chosen methods of derivation, integration, and interpolation, one might be able to probe for a lower bound on the acceptable degree of coarseness for a type of signal based on a desired accuracy threshold.

VI. ALGORITHM PERFORMANCE AND COMPARISONS

This section summarizes our comparisons between the performance of GORA and that of the DTW and FastDTW [7] algorithms. Specifically, we evaluate the performance of each of the above algorithms in terms of both accuracy in matching signals and computational efficiency. All comparisons were performed in Python 2.7 and the DTW and FastDTW implementations we used in our experiments were from the official Python package [13].

A. Comparison regime

We compared the performance of GORA with the DTW algorithm and implementations of the FastDTW algorithm with radii of 1, 5, and 20. The procedures with which we performed comparisons using both synthetic trajectories in \mathbb{R}^3 and video sequences are described as follows: For a given number of time instances, we randomly selected 50 different template signals. For each template signal, two initial parameterizations in the TRG were randomly generated and used to parameterize the original signal, creating 50 pairs of input

signals, which were then fed to GORA and the DTW and FastDTW algorithms.

To ensure fair comparisons between algorithms, we use a modified version of GORA designed for the pairwise comparison of two signals, which is outlined in Fig. 2. This version accepts two input signals, $X_1(t)$ and $X_2(t)$, computes in parallel to their respective UST reparameterizations as defined in Algorithm 1, i.e. $X_1^*(t)$ and $X_2^*(t)$, and outputs the error between the two UST reparameterizations given by (7). Similarly, we normalized the accumulated cost error output by the DTW and FastDTW algorithms under the Euclidean norm by dividing it by the length of the optimal warping path. Run time comparisons were performed using the `clock` module in Python’s `time` package. Given two input signals, we defined the run time (what we called computational efficiency) to be the time it took each algorithm to output the error between them.

B. Results

Fig. 3 and Fig. 4 compare the performance of GORA and the DTW and FastDTW algorithms using signals in the form of trajectories in \mathbb{R}^3 and vectorized video sequences, respectively. Figs. 3a and 4a show the mean run time of each algorithm from 20 to 150 time instances. Figs. 3c and 4c show the corresponding standard deviations from the mean run times for each algorithm.

For both trajectories in \mathbb{R}^3 and video sequences, as the total number of time instances increases, DTW’s run time grows quadratically (i.e. $O(T^2)$ complexity) while all iterations of the FastDTW algorithm and GORA achieve linear complexity (i.e. $O(T)$). However, in both cases GORA’s run time is less than that of all the DTW methods, and GORA’s complexity grows more slowly than the fastest implementation of FastDTW (radius = 1). In addition, GORA’s run time has a similar degree of stability (in the sense of smaller deviations from the overall mean run time) as that of

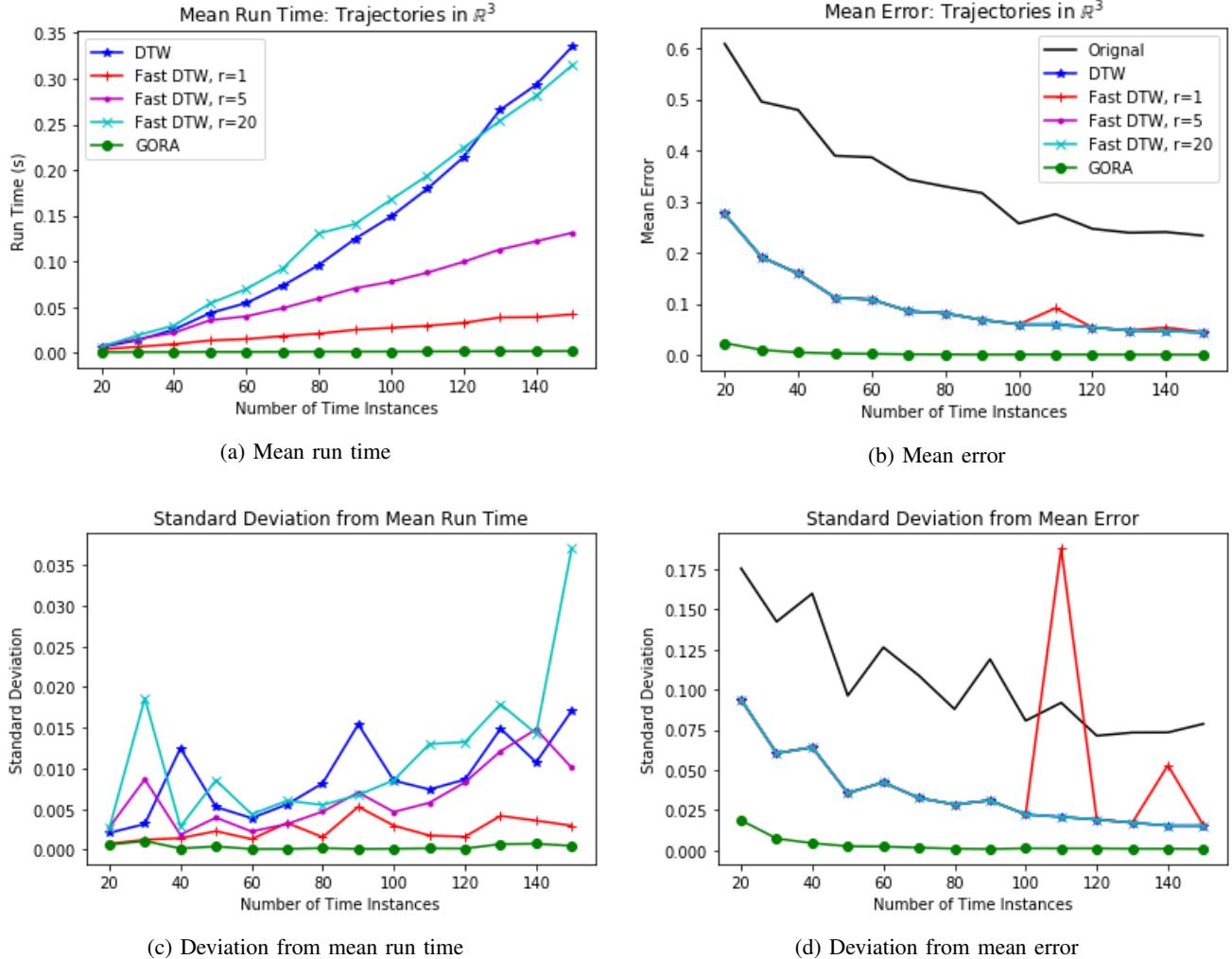


Fig. 3: Algorithm performance: synthetic trajectories in \mathbb{R}^3 .

the FastDTW implementation with radius = 1, and remains significantly more stable than other DTW methods.

Figs. 3b and 4b show the mean error between signal pairs given by each algorithm from 20 to 150 time instances. Figs. 3d and 4d show the corresponding standard deviations from the mean error for each algorithm. In both cases, GORA is significantly more accurate (in the sense that the computed error between signal pairs known to represent the same dynamic phenomena is small) than the DTW algorithm and all implementations of the FastDTW algorithm. It was often the case that the DTW algorithm and the implementations of the FastDTW algorithm gave identical errors, since it is possible for the FastDTW algorithm to construct the same accumulated cost matrix as the DTW algorithm.

The authors believe that the disparity in accuracy between GORA and the implementation of the FastDTW algorithm with radius = 1 is especially significant. Since the error produced by the implementation of the FastDTW algorithm with radius = 1 is both highly inaccurate and unstable (in the sense of large deviations from the mean error), especially for input signals in the form of video sequences, this suggests

that an effective implementation of the FastDTW algorithm requires a larger radius. As such, the run time disparity between an effective FastDTW implementation and GORA is likely somewhere between the implementations of FastDTW with radius = 1 and radius = 5.

That being said, these results constitute only an initial analysis with two types of elementary data. However, they do suggest that GORA has potential to be a highly effective framework for signal comparison and action recognition.

VII. DISCUSSION

A crucial difference between GORA and the DTW and FastDTW algorithms is GORA's reliance on interpolation to recover the UST reparameterization of the input signal. Depending on the context, this can be an advantage or disadvantage for the GORA framework. For example, consider the problem of signal comparison or action recognition over a space of signals where computing the error between signals at a given time instance is itself computationally expensive to perform. By interpolating, GORA only has to compute the pairwise error between signals no more than T times, where

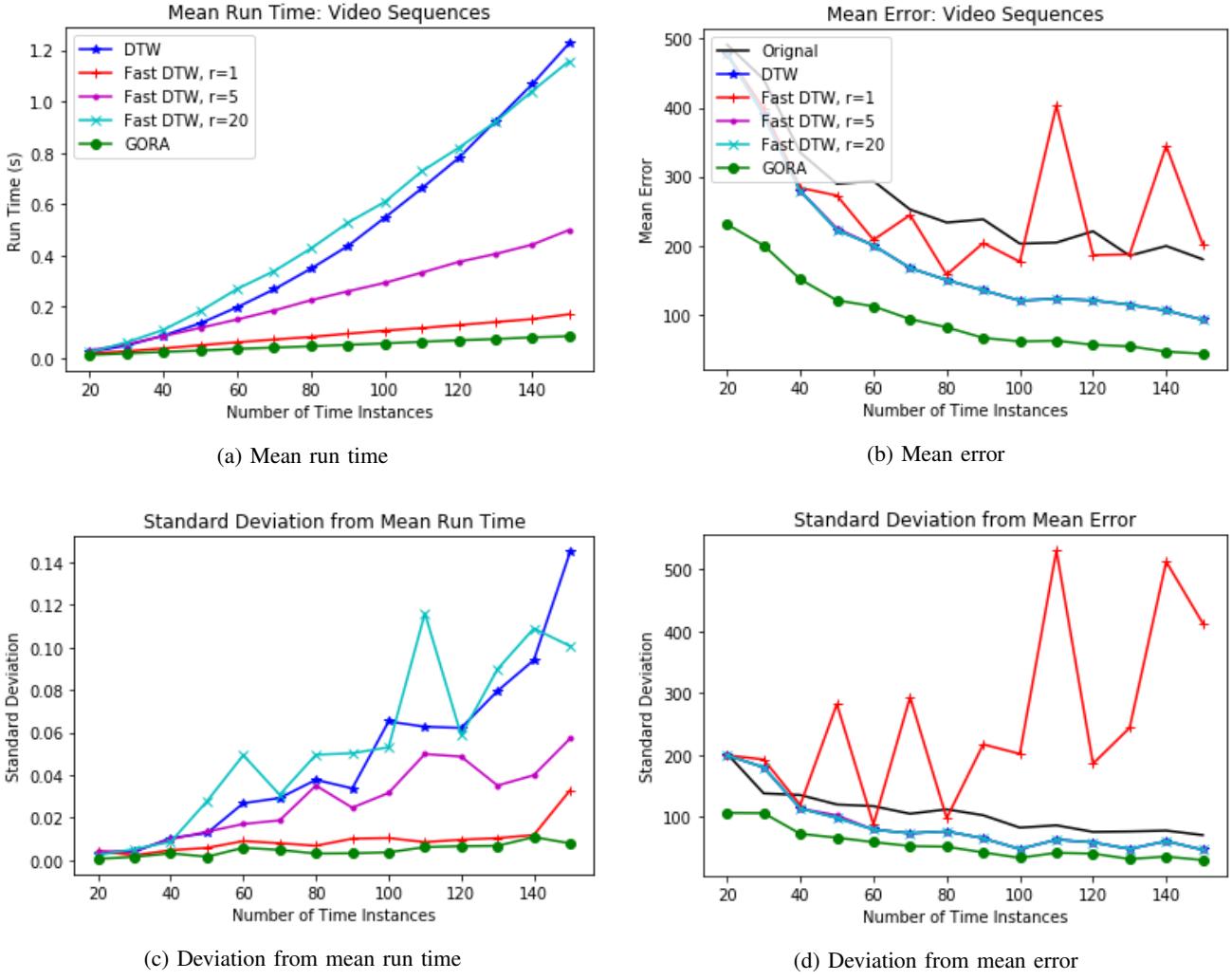


Fig. 4: Algorithm performance: vectorized video sequences from the Weizmann Database.

T is the total number of time instances. On the other hand, all DTW methods will have to compute this error between $O(T)$ and $O(T^2)$ times. If GORA's chosen method of interpolation is relatively inexpensive, this could give it a significant run time advantage over DTW methods.

However, this could easily become a disadvantage for GORA if the chosen method of interpolation is relatively expensive compared to the computation of the pairwise error between signals at a given time instance. In particular, this might serve to explain why GORA's run time advantage over the FastDTW implementation with radius = 1 is smaller with video sequences than with trajectories in \mathbb{R}^3 . For trajectories in \mathbb{R}^3 , GORA only performs three instances of linear interpolation — one in each dimension of the trajectory. In contrast, for video sequences GORA performs $n = \text{width} \times \text{height}$ instances of linear interpolation along each dimension of the trajectory in \mathbb{R}^n representing the vectorized video sequence. While this also means that DTW methods have to perform pairwise error computations with larger signals, it's likely that the cost due to increasing the instances of interpolation outweighs the cumulative cost of

computing the euclidean norm in \mathbb{R}^n . The videos in the Weizmann Database are relatively small (180×144 pixels) and it is unknown with video sequences of larger dimensions whether GORA's mean run time would remain favorable relative to other methods.

The GORA framework is very much an active work-in-progress. Currently, we are exploring the GORA's potential in providing a foundation for a more robust algorithm able to minimize or eliminate nuisance parameters while simultaneously reparameterizing signals to a UST [10]. The development of such an algorithm able to inherently compensate for perturbations such as noise or motion artifacts while maintaining a similar linear complexity to GORA would mark an important milestone toward the goal of robust robotic action recognition of human motions in real-time. Additionally, a well-known strength of DTW methods is their ability to effectively compare signals with different numbers of time instances. This is something we have yet to consider in our implementations of GORA and a topic we plan to address in our future work.

VIII. CONCLUSIONS

In this paper, we introduced the *Globally Optimal Reparameterization Algorithm* (GORA) for signal alignment and comparison. This algorithm reparameterizes signals to a *universal standard timescale* (UST), allowing for element-wise comparisons between multiple signals at each instance of time with linear time complexity of $O(T)$. In particular, we defined procedures for applying this algorithm to characterize and compare signals in the form of real trajectories and video sequences.

Our experimental results provide both a numerical validation of GORA's theoretical basis and suggest that the GORA framework has the potential to become a viable alternative to DTW methods for signal comparison and action recognition purposes. For signals in the form of real trajectories in \mathbb{R}^3 and vectorized video sequences with a fixed number of time instances, GORA's computational complexity is less than that of the FastDTW algorithm with radius= 1 and GORA's accuracy in matching signals representing fundamentally the same phenomena exceeds that of both the DTW algorithm and implementations of the FastDTW algorithm with radii of 1, 5, and 20.

IX. ACKNOWLEDGMENTS

The authors would like to thank Dr. Jin Seob Kim and Ms. Mengdi Xu for useful discussions that contributed to this work. This work was performed under National Science Foundation grant IIS-1619050 and Office of Naval Research Award N00014-17-1-2142.

REFERENCES

- [1] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb 1978.
- [2] H. Shaikh, L. C. Mesquita, S. D. C. S. Araujo, P. Student, and A. P. Professor, "Recognition of isolated spoken words and numeric using MFCC and DTW," *International Journal of Engineering Science*, vol. 10539, 2017.
- [3] S. Semenza, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using Dynamic Time Warping," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, July 2011, pp. 1–5.
- [4] E. J. Keogh and M. J. Pazzani, "Scaling up Dynamic Time Warping to massive datasets," in *Principles of Data Mining and Knowledge Discovery*, J. M. Zytkow and J. Rauch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 1–11.
- [5] T. Zhu, Q. Zhao, W. Wan, and Z. Xia, "Robust regression-based motion perception for online imitation on humanoid robot," *International Journal of Social Robotics*, vol. 9, no. 5, pp. 705–725, Nov 2017. [Online]. Available: <https://doi.org/10.1007/s12369-017-0416-9>
- [6] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 2001 SIAM International Conference on Data Mining*. SIAM, 2001, pp. 1–11.
- [7] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.
- [8] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, Mar 2005. [Online]. Available: <https://doi.org/10.1007/s10115-004-0154-9>
- [9] F. Zhou and F. D. la Torre, "Generalized time warping for multi-modal alignment of human motion," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1282–1289.
- [10] G. S. Chirikjian, "Signal classification in quotient spaces via globally optimal variational calculus," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 735–743, July 2017. [Online]. Available: <https://arxiv.org/abs/1705.03744>
- [11] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *The Tenth IEEE International Conference on Computer Vision (ICCV'05)*, 2005, pp. 1395–1402.
- [12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, December 2007.
- [13] K. Tanida, "slaypni/fastdtw: fastdtw-0.3.2," Jul. 2017. [Online]. Available: <https://pypi.org/project/fastdtw/>