2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Signal Alignment for Humanoid Skeletons via the Globally Optimal Reparameterization Algorithm

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

***Abstract*— The general ability to analyze and classify the 3D kinematics of the human form is an essential step in the development of socially adept humanoid robots. A variety of different types of signals can be used by machines to represent and characterize actions such as RGB videos, infrared maps, and optical flow. In particular, skeleton sequences provide a natural 3D kinematic description of human motions and can be acquired in real time using RGB+D cameras. Moreover, skeleton sequences can be generalized to characterize the motions of both humans and humanoid robots. The *Globally Optimal Reparameterization Algorithm* (GORA) is a novel, recently proposed algorithm for signal alignment in which signals are reparameterized to a globally optimal *universal standard timescale* (UST). Here, we introduce a variant of GORA for humanoid action recognition with skeleton sequences, which we call GORA-S. We briefly review the algorithm's mathematical foundations and contextualize them in the problem of action recognition with skeleton sequences. Subsequently, we introduce GORA-S and discuss parameters and numerical techniques for its effective implementation. We then compare its performance with that of the *DTW* and *FastDTW* algorithms, in terms of computational efficiency and accuracy in matching skeletons. Our results show that GORA-S attains a complexity that is significantly less than that of any tested DTW method. In addition, it displays a favorable balance between speed and accuracy that remains invariant under changes in skeleton sampling frequency, lending it a degree of versatility that could make it well-suited for a variety of action recognition tasks.**

I. INTRODUCTION

As the name suggests, humanoid robots inherently resemble human bodies and typically consist of a head, torso, two arms, and two legs. Many of these robots are designed to emulate human behaviors, such as walking or dancing, and to communicate with us both verbally and non-verbally, through conversation or universal gestures such a wave or a cheer [1]. Humanoid robots have the potential to perform a variety of challenging tasks normally reserved for their biological counterparts, such as providing office or administrative support in the role of a receptionist or caring for the elderly; as such, they have become a popular area of research in robotics.

Socially adept humanoid robots require a detailed knowledge of human actions in the context of daily life. A first step toward this end is the recognition and classification of specific actions using signals or sequences. With the recent emergence of new machine learning and computer vision techniques, new robotic action recognition methods have been developed for a variety of different types of signals, such as sequences of RGB images [2], infrared maps [3] and 3D skeletons [4]. The foundation of action recognition lies in the problem of signal alignment, in the sense that prior to categorizing sets of sequences, one should be able to temporally reparameterize the sequences in such a way that enables standardized comparisons. In particular, a robust action recognition algorithm should also be adaptable for use with different modalities of data. For example, such an algorithm should be able to accurately characterize an action, such as a wave from a NAO humanoid robot, by synthesizing and comparing multiple sequences of different data types, as shown in Fig. 1.

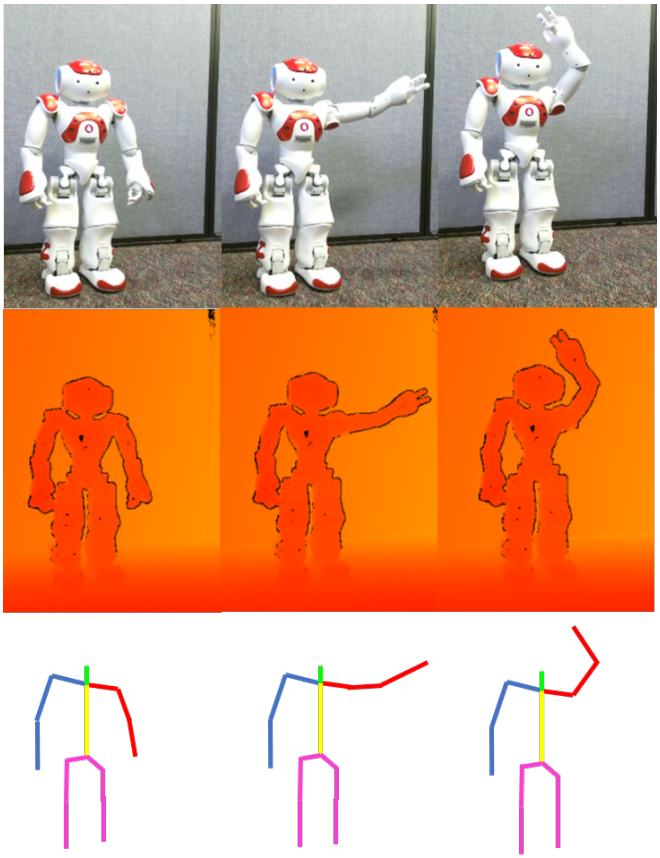


Fig. 1: A wave by a NAO robot represented in three different types of data. 1^{st} row: RGB video; 2^{nd} row: infrared depth map; 3^{rd} row: skeleton sequence. A robust action recognition framework should be able to synthesize and compare signals of different data types in order to correctly categorize a given action.

The *Globally Optimal Reparameterization Algorithm*

*Equally contributing authors

Thomas W. Mitchel, Sipu Ruan and Gregory S. Chirikjian are with the Laboratory for Computational Sensing and Robotics, Johns Hopkins University, Baltimore, Maryland 21218 {tmitchel, ruansp, gchirik1}@jhu.edu

(**GORA**) is a novel, recently proposed algorithm for signal alignment in which signals are reparameterized to a *universal standard timescale* (**UST**) using principles of variational calculus [5], [6]. GORA has been initially applied to compare RGB video sequences [5], and has shown potential in providing a highly-effective framework for signal comparisons as an alternative to the well known DTW algorithm [7] and its variants [8]–[10].

In this paper, we introduce a variant of GORA for humanoid action recognition with skeleton sequences, which we call **GORA-S**. Skeleton sequences can be viewed as trajectories in the Lie group $\mathrm{SE}(3)^n \cong \mathrm{SE}(3) \times \mathrm{SE}(3) \times ... \times \mathrm{SE}(3)$ [11] and provide a natural 3D kinematic description of the motions, gestures, and actions of both natural and artificial humanoids. As such, action recognition with skeleton sequences is immediately relevant to current research in humanoid robotics as the general ability to analyze and classify the 3D kinematics of the human form will likely play an integral role in the development of socially adept machines.

Previously, GORA's effectiveness has only been evaluated with respect to signals that can be expressed as trajectories in $\mathbb{R}^n$ [5]. However, skeleton sequences' natural representations as trajectories in the Lie group $\mathrm{SE}(3)^n$ pose a distinct challenge. While the overall structure of GORA-S differs little from that of GORA, concepts integral to GORA such as differentiation and interpolation require fundamentally different treatments in $\mathrm{SE}(3)^n$ than in $\mathbb{R}^n$. In this context, the main contribution of GORA-S is *a framework for signal alignment and action recognition with skeleton sequences that attains a significantly lower computational complexity than DTW methods. In addition, it displays a favorable balance between speed and accuracy that remains almost invariant under changes in the temporal sampling frequency of the skeletons*.

The remainder of the paper is organized as follows: First, we provide a brief review of the mathematical foundations of the GORA framework and introduce GORA-S in the context of action recognition with skeleton sequences. Next, we review the algorithm and discuss the chosen parameters and numerical techniques used in its application. We then provide a verification of the algorithm by comparing its performance relative to the DTW and FastDTW algorithms [9] in terms of both computational efficiency and accuracy in matching skeleton sequences from the NTU RGB+D Action Recognition Dataset [12]. Other techniques such as the Longest Common Subseqeuence method have been used to to characterize actions using skeleton sequences [13], [14]. However, to the authors' knowledge these approaches have not previously been used to perform comparisons directly on $\mathrm{SE}(3)^n$. As such, we restrict our comparisons to the DTW family of algorithms, which can be naturally extended to perform signal alignments on $\mathrm{SE}(3)^n$. We then discuss the results and conclude with a short remark on the computational significance of the differences between GORA-S and DTW methods.

II. PROBLEM STATEMENT AND MATHEMATICAL FOUNDATIONS

The GORA algorithm is based on the notion that any temporal misalignment between two arbitrary signals can be compensated for by reparameterizing each to a universal standard timescale (**UST**). Such temporal parameterizations are differentiable strictly monotonically increasing functions on the unit interval. The set of all such functions forms the *Temporal Reparameterization Group* (**TRG**), denoted as $\mathcal{T}$, under the operation of composition of functions. [5], [6].

The foundation of this approach lies in the fact that for a cost function of the form

$$f(\tau, \dot{\tau}) = \dot{\tau}^2 \mathfrak{g}(\tau), \tag{1}$$

where $\tau \in \mathcal{T}$ and $\mathfrak{g} : [0,1] \to \mathbb{R}_{>0}$ is differentiable, the solution to the Euler-Lagrange equation

$$\frac{\partial f}{\partial \tau} - \frac{d}{dt}\left(\frac{\partial f}{\partial \dot{\tau}}\right) = 0, \tag{2}$$

globally minimizes the functional

$$J = \int_0^1 f\left(\tau, \dot{\tau}, t\right)\, dt, \tag{3}$$

where $\dot{\tau} = d\tau/dt$. In addition, the globally optimal solution, $\tau^* \in \mathcal{T}$, is unique and can be recovered by inverting the function

$$F(\tau^*) \doteq \frac{1}{c}\int_0^{x^*} \mathfrak{g}^{\frac{1}{2}}(\sigma)\, d\sigma = t, \tag{4}$$

where t is the temporal variable and

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

A full proof and numerical validation of these results can be found in [5], [6].

In general, any kind of temporally evolving signal, $X(t)$, can be thought of as a mapping from the unit interval to the space S, i.e. $X : [0,1] \to S$, on which that particular type of signal evolves. Defining a metric d on S, (S, d) becomes a metric space. Here we consider signals of the form

$$X(t) = (g_1(t),\ g_2(t),\ \ldots,\ g_n(t)) \in \mathrm{SE}(3)^n, \tag{5}$$

which we call a *skeletons* when $n > 1$. These types of signals are typically acquired using RGB+D cameras, a well-known example being Microsoft's Kinect. Skeletons can provide detailed descriptions of humanoid motions in 3D space. As a result, they have become an increasingly popular way to characterize motions due to their ability to be acquired in real time [11], [12], [15].

The trick in applying the above results to the problem of action recognition lies in how the function $\mathfrak{g}(\tau)$ is defined. In particular, *it should measure the rate of change of a given signal along the temporal axis*. In this context, the solution to (2), $\tau^* \in \mathcal{T}$, is such that the reparameterization of $X(t)$ with respect to τ^*, $X(\tau^*(t))$, globally minimizes the rate of change of the signal. Since τ^* is unique, it serves the role of a parameterization to a UST, in the sense that for any

collection of signals, $X_1(t),\ \ldots,\ X_k(t)$, their corresponding globally optimal solutions, $\tau_1^*(t),\ \ldots,\ \tau_k^*(t)$, reparameterize each signal to the same timescale which minimizes their rate of change.

Given an arbitrary skeleton signal $X(t) \in \mathrm{SE}(3)^n$ and initial temporal variable t as inputs, GORA-S recovers the UST parameterization $\tau^*(t)$ corresponding to $X(t)$ and outputs the UST reparameterization of the skeleton, $X^*(t) = X(\tau^*(t))$. This UST reparameterization of the input skeleton can then be compared element-wise with other UST reparameterized skeletons to determine whether or not they represent the same action, resulting in a linear complexity of $O(T)$, where T is the total number of time instances in the sequence.

As an illustration of this method, suppose $X_1(t), X_2(t) \in \mathrm{SE}(3)^n$ are skeletons upon which minimal nuisance parameters or motion artifacts are acting and let d be an arbitrary metric on the $\mathrm{SE}(3)^n$. Subsequently GORA-S can be used to find the UST reparameterizations of $X_1(t)$ and $X_2(t)$, given by $X_1^*(t) = X_1(\tau_1^*(t))$ and $X_2^*(t) = X_2(\tau_2^*(t))$, respectively. Then, we can say that $X_1(t)$ and $X_2(t)$ represent the same action or gesture if

$$\int_0^1 d(X_1^*(t), X_2^*(t))\ dt \approx 0,$$

despite any initial temporal misalignment.

III. THE GLOBALLY OPTIMAL REPARAMETERIZATION ALGORITHM FOR SKELETON SEQUENCES (GORA-S)

This Globally Optimal Reparameterization Algorithm for Skeleton Sequences (GORA-S) is defined in Algorithm 1.

Algorithm 1: Globally Optimal Reparameterization Algorithm for Skeleton Sequences (GORA-S)

Input : Input skeleton sequence $X(t)$; Initial temporal variable t

Output: UST reparameterization of skeleton $X^*(t)$

1 Calculate $\dot{X}(t) = dX/dt$

2 Compute $\mathfrak{g}(t;\ \dot{X}(t))$;

3 $c = \mathit{NumericalIntegration}(\mathfrak{g}^{\frac{1}{2}}(\sigma),\ [0,1])$;

4 $F(\tau^*) = \frac{1}{c}\ \mathit{NumericalIntegration}(\mathfrak{g}^{\frac{1}{2}}(\sigma),\ [0,\tau^*])$;

5 $\tau^*(t) = F^{-1}(t)$;

6 $X^*(t) = \mathit{Interpolation}(X(t), \tau^*(t);\ \dot{X}(t))$;

Given $\mathfrak{g}(t)$, numerically calculating $\tau^*(t)$ is relatively straightforward and can be done efficiently by first computing $F(\tau^*)$ as in (4) then interpolating t as a function of $F(\tau^*)$. The UST reparameterization of the input signal, $X^*(t)$, is recovered by interpolating the input signal as function of t at the query points given by τ^*. Additionally, since the temporal derivative of the input signal,

$$\dot{X}(t) = (\dot{g}_1(t),\ \ldots, \dot{g}_n(t))$$

is necessary for both the computation of $\mathfrak{g}(t)$ and to perform our chosen method of interpolation in step 6, we choose to compute it once at the beginning of the algorithm for the sake of increased computational efficiency.

Overall, the algorithmic structure of GORA-S differs only slightly from that of GORA. However, skeleton sequences' representation as trajectories in $\mathrm{SE}(3)^n$ requires a fundamental reformulation of $\mathfrak{g}(\tau)$ and the method of interpolation used to recover the optimal reparameterization. The following sections describe these specific formulations unique to GORA-S.

A. Formulation of $\mathfrak{g}(t)$

For skeleton sequences, a natural choice for the definition of $\mathfrak{g}(t)$ consistent with (1) is based on the body velocities of the joint trajectories in $\mathrm{SE}(3)$ comprising the signal. For a skeleton with n joint trajectories as in (5), we defined $\mathfrak{g}(t)$ as

$$\mathfrak{g}(t) = \mathfrak{g}(t; \dot{X}(t)) = \sum_{j=1}^{n} \left\| g_j^{-1} \frac{\partial g_j}{\partial \tau} \right\|_W^2. \tag{6}$$

Here $\|\cdot\|_W$ denotes the weighted Frobenius norm defined such that for any $A \in \mathbb{R}^{4\times4}$,

$$\|A\|_W = \sqrt{\mathrm{tr}\,(A^T W A)}, \tag{7}$$

for some symmetric 4×4 matrix W. Here, we defined W as

$$W = \begin{bmatrix} J & \mathbf{0} \\ \mathbf{0}^T & m \end{bmatrix} \tag{8}$$

with $m = 1$ and

$$J = \frac{1}{2}\mathrm{tr}(I)\mathbb{I} - I \tag{9}$$

where I is the 3×3 diagonal inertia tensor corresponding to a solid sphere of unit mass and $\mathbb{I}$ denotes the 3×3 identity matrix. Further details of this norm can be found in [16].

B. Interpolation on $SE(3)^n$

Given a set of T time instances $\{t_i\}$, the corresponding values of a skeleton sequence

$$\{X(t_i)\} = \{(g_1(t_i),\ \ldots, g_n(t_i))\}$$

and the values of its temporal derivative

$$\{\dot{X}(t_i)\} = \{(\dot{g}_1(t_i),\ \ldots, \dot{g}_n(t_i))\},$$

we can construct a piecewise interpolating curve for each joint trajectory, $g_j(t),\ 1 \le j \le n$, such that

$$\bar{X}(t) = (g_1(t),\ \ldots, g_n(t))$$

passes through $X(t_i)$ at time instance $t_i, 1 \le i \le T$, as follows [17]:

For a given joint trajectory $g_j \in \mathrm{SE}(3)$, we can define a cubic minimum acceleration curve in $\mathrm{Aff}^+(4,\mathbb{R})$ as

$$\begin{aligned} M(t) &= \begin{bmatrix} M_{3\times3}(t) & \mathbf{m}(t) \\ \mathbf{0}^T & 1 \end{bmatrix} \\ &= M_3 t^3 + M_2 t^2 + M_1 t + M_0, \quad t \in [t_i, t_{i+1}] \end{aligned}$$

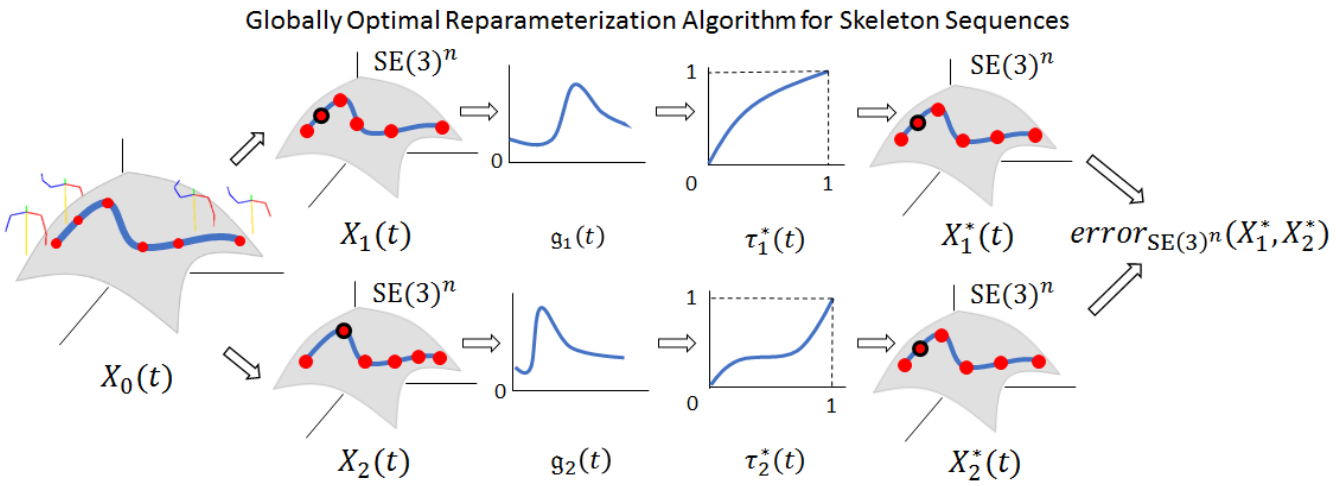


Fig. 2: The experimental work flow for evaluating the computational efficiency and accuracy of GORA-S in matching skeleton sequences: A template skeleton $X_0(t)$ is selected and parameterized with respect to two randomly generated functions in the TRG (e.g. $\tau_1(t)$ and $\tau_2(t)$) to create two skeletons $X_1(t), X_2(t) \in \mathrm{SE}(3)^n$, identical up to their temporal alignment. For each skeleton, $\tau^*(t)$ is computed from $\mathfrak{g}(t)$ and used to find the UST reparameterzaion as in Algorithm 1. The algorithm then returns the error between the UST reparameterizations computed using the metric in (11).

where

$$M_3 = 6\,\frac{\dot{g}_j(t_i) + \dot{g}_j(t_{i+1})}{(\Delta t)^2} - 12\frac{\Delta x}{(\Delta t)^3}$$
$$M_2 = \frac{\Delta v}{\Delta t} - M_3\frac{t_i + t_{i+1}}{2}$$
$$M_1 = \dot{g}_j(t_i) - M_3\frac{t_i^2}{2} - M_2 t_i$$
$$M_0 = g_j(t_i) - M_3\frac{t_i^3}{6} - M_2\frac{t_i^2}{2} - M_1 t_i$$

and

$$\Delta t = t_{i+1} - t_i$$
$$\Delta x = g_j(t_{i+1}) - g_j(t_i)$$
$$\Delta v = \dot{g}_j(t_{i+1}) - \dot{g}_j(t_i)$$

Taking J as defined in (9) we can find the singular value decomposition of $M_{3\times3}(t)J$ to recover the matrices $U(t), \Sigma(t), V(t)$ such that

$$M_{3\times3}(t)J = U(t)\Sigma(t)V^H(t).$$

Then, the curve interpolating the joint trajectory $g_j \in \mathrm{SE}(3)$ on the interval $[t_i, t_{i+1}]$ is given by

$$g_j(t) = \begin{bmatrix} R(t) & \mathbf{r}(t) \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad t \in [t_i, t_{i+1}]$$

where

$$R(t) = U(t)V^H(t) \in \mathrm{SO}(3),$$
$$\mathbf{r}(t) = \mathbf{m}(t) \in \mathbb{R}^3.$$

C. Numerical differentiation of skeleton sequences

We use Fornberg's method to numerically calculate $\dot{X}(t)$ for skeleton sequences by high order finite difference approximation [18], [19]. This method can compute derivatives on both regularly and irregularly spaced grids, making it potentially advantageous in real world scenarios in the case where an RGB+D camera might fail to track a skeleton at certain time instances. In our experiments, we use Bjorn Dahlgren's `finitediff` package in Python 2.7, which enables efficient numerical differentiation over arrays via Fornberg's method [20].

D. Error metric

Given two skeleton sequences,

$$X_1(t) = (g_1(t), \ldots, g_n(t))$$
$$X_2(t) = (h_1(t), \ldots, h_n(t)),$$

both in $\mathrm{SE}(3)^n$, we defined the element-wise distance between them at an arbitrary time instance t_0 as

$$d_{\mathrm{SE}(3)^n}(X_1(t_0), X_2(t_0)) = \frac{1}{n}\sum_{j=1}^{n}\left\|\log_{\mathrm{SE}(3)}\left([g_j(t_0)]^{-1}[h_j(t_0)]\right)\right\|_W, \qquad (10)$$

where $\log_{\mathrm{SE}(3)}(\cdot)$ denotes the logarithm operation which maps elements in $\mathrm{SE}(3)$ to their corresponding elements in the Lie algebra, $\mathfrak{se}(3)$ [21]. Similarly, we defined the average distance or error between two skeletons across all time instances $\{t_i\}$ to be

$$error_{\mathrm{SE}(3)^n} = \frac{1}{T}\sum_{i=1}^{T} d_{\mathrm{SE}(3)^n}(X_1(t_i), X_2(t_i)), \qquad (11)$$

where T is the total number of the time instances.

IV. Algorithm Performance and Comparisons

GORA's demonstrated effectiveness in matching signals on $\mathbb{R}^n$ [5] does not automatically guarantee the effectiveness of GORA-S in aligning skeleton sequences on $\mathrm{SE}(3)^n$. While the overall structure of GORA-S remains similar to that of GORA, GORA-S is characterized by fundamental reformulations of concepts integral to the GORA framework required to adapt the algorithm for skeleton sequences (Sections III-A, III-B, III-C). As such, these differences necessitate a separate evaluation of the effectiveness of GORA-S relative to other algorithms.

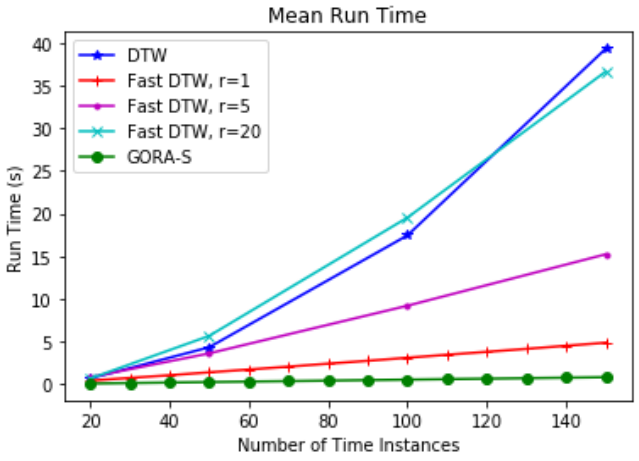
This section summarizes our comparisons between GORA-S and that of the DTW and FastDTW [9] algorithms in the context of action recognition with skeleton sequences. Specifically, we evaluated the performance of each of the above algorithms in terms of both accuracy in matching skeleton sequences and computational efficiency. For the sake of consistency, our comparison regime mirrors the regime we used to evaluate GORA against other algorithms [5]. All comparisons were performed in Python 2.7 and the DTW and FastDTW implementations used in our experiments were from the official Python package [22]. The experiments were performed on an Intel Core i7-7600U CPU @ 2.80GHz.

A. Pre-processing

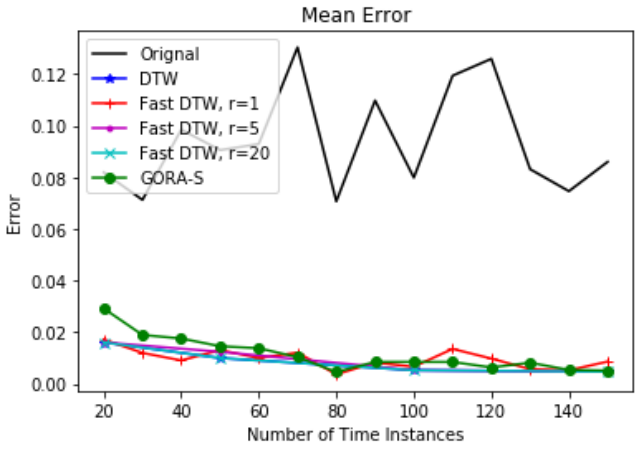
Due to the large size of the NTU Action Recognition Dataset, we randomly chose 50 skeleton sequences from each of the drinking (A001), clapping (A010), cheering (A022), and waving (A023) action classes to create a pool of skeletons with which to compare algorithms. It is important to note that for the sake of sampling consistency, we trimmed all skeleton sequences in this pool such that each sequence depicted only a *single instance of an action being performed*. For example, skeleton sequences of a person waving multiple times were trimmed to show only a single wave. Since each of the four chosen action classes were characterized by upper body movements, we pruned each skeleton in the pool to include only the joints comprising the torso, arms, and head to eliminate any added noise from irrelevant joints. We also excluded any joints that did not contain rotational components in SO(3). Additionally, we applied a normalization procedure to each skeleton to account for differences in physical size between subjects and body orientation with respect to camera coordinates. The complete details of the procedure can be found in [12].

B. Comparison regime

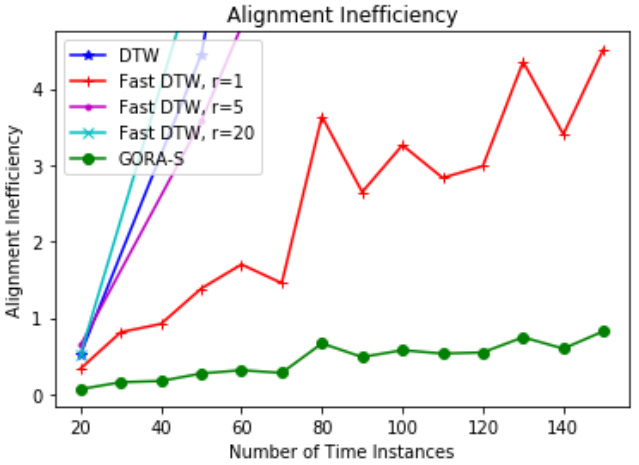
We compared the performance of GORA-S with the DTW algorithm and implementations of the FastDTW algorithm with radii of 1, 5, and 20. The procedures with which we performed action recognition comparisons with skeleton sequences are as follows: For a given number of time instances, we randomly selected 50 different template skeletons from the pool. For each template skeleton, two initial parameterizations in the TRG were randomly generated and used to parameterize the original signal, creating 50 pairs of input signals with different temporal alignments, which were then fed to GORA-S and the DTW and FastDTW algorithms.



(a) Mean run time



(b) Mean error



(c) Alignment inefficiency

Fig. 3: Algorithm performance in aligning skeleton sequences in $\mathrm{SE}(3)^n$.

To ensure fair comparisons between algorithms, we use a modified version of GORA-S designed for pairwise comparisons between skeleton sequences, outlined in Fig. 2. This version accepted two input skeletons, $X_1(t), X_2(t) \in \mathrm{SE}(3)^n$, computed in parallel their respective UST reparameterizations, i.e. $X_1^*(t)$ and $X_2^*(t)$, as in Algorithm 1 and output the average error between the two UST reparameterizations given by (11). An example of this approach can be seen in the supplementary video accompanying this paper. Similarly, we implemented the DTW and FastDTW algorithms such that the metric given in (10) was used to compute the element-wise distance between skeletons at arbitrary time instances. Furthermore, we normalized the accumulated cost error output by the DTW and FastDTW algorithms under (10) 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 two skeleton sequences.

To quantify the relationship between run time and accuracy, we introduce a quantity we call the *alignment inefficiency*, denoted as $\mathcal{I}$. For an arbitrary signal alignment algorithm and m pairs of input skeletons, let μ_0 denote the mean error between initial skeletons pairs given by (11) and σ_0 denote the corresponding standard deviation. Additionally, let E_i denote the error between the i^{th} skeleton pair found by the algorithm, and T_i denote the run time of the algorithm in aligning the i^{th} pair. Subsequently, we can define the alignment inefficiency as

$$\mathcal{I} = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\mu_0 - E_i}{\sigma_0} \right) T_i \tag{12}$$

The alignment inefficiency gives a spatially normalized measure of an algorithm's performance that equally weights run time and accuracy. For a given set of skeleton pairs, an algorithm with a small run time and a larger error would have a similar alignment inefficiency as that of another algorithm with a larger run time and a smaller error. However, a 'better' algorithm with both a small run time and small error would have the smallest alignment inefficiency. In addition, the rate of change of the alignment inefficiency with respect to the temporal sampling frequency of the input skeletons provides a measure of the sensitivity of the algorithm's performance to the degree of coarseness or fineness in the sampling of the signals.

C. Results

Fig. 3. compares the performance of GORA-S and the DTW and FastDTW algorithms. Fig. 3a. shows the mean run time of each algorithm from 20 to 150 time instances. For skeleton 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 GORA-S's run time is less than that of all DTW methods, and its complexity grows more slowly than the fastest implementation of FastDTW (radius $= 1$).

Fig. 3b. shows the mean error between skeleton pairs identical up to their temporal alignment given by each algorithm from 20 to 150 time instances. In the sense that the computed error between skeleton pairs representing the same action is small, the accuracy of GORA-S is comparable to that of the DTW algorithm and all implementations of the FastDTW algorithm. However, it was slightly larger when comparing very coarsely sampled skeleton sequences. It was often the case that the DTW algorithm and the implementations of the FastDTW algorithm gave identical errors, since it is possible for both to construct the same accumulated cost matrix.

Fig. 3c. compares the alignment inefficiencies of GORA-S and the DTW and FastDTW algorithms across skeleton pairs with 20 to 150 time instances. The alignment inefficiencies of the DTW algorithm and implementations of the FastDTW algorithm with radii of 5, 20 grow much faster than those of GORA-S and FastDTW algorithm with r $= 1$. As such, the larger values of these quantities are not shown.

In the opinion of the authors, the disparity between the alignment inefficiency of the FastDTW algorithm and that of GORA-S is especially significant. While GORA-S is slightly less accurate than FastDTW with radius $= 1$ when comparing very coarsely sampled skeleton sequences, its lower alignment inefficiency implies that the loss in accuracy is outweighed by the relative increase in computational efficiency gained due to its faster run time.

More importantly, GORA-S's alignment inefficiency grows very slowly as the sampling frequency of the input skeletons increases. This implies that the overall performance of GORA-S changes very little with respect to the sampling frequency of the skeleton sequences. In other words, any increases in run time when comparing more finely sampled skeletons are offset by directly proportional decreases in the computed error and vice-versa. Moreover, this suggests that GORA-S has the potential to be an extremely 'versatile' algorithm that can perform efficiently in a variety of roles. Its faster run times and competitive accuracy relative to DTW methods when comparing coarsely sampled skeletons make it well-suited for real-time action recognition. Alternatively, its ability to remain efficient when comparing finely sampled skeletons could make it an effective signal alignment algorithm for deep learning applications. However, it is important to keep in mind that these results represent only an initial validation of GORA-S using skeletons depicting four elementary humanoid actions. Further analysis is needed to properly contextualize the strengths and weakness of the algorithm.

V. DISCUSSION

As noted in [5], an important difference between GORA-S and the DTW and FastDTW algorithms is the reliance of GORA-S on interpolation to recover the UST reparameterization of the input skeleton. Here we use the method described in Section III-B. This requires computing the singular value decomposition factorization of the rotational

component of the interpolated curve in $\mathrm{GL}(3,\mathbb{R})$ at each of the desired points of evaluation, for a total of nT SVD computations per sequence. While GORA-S is faster than all implementations of the FastDTW algorithm, a simpler and less computationally expensive method could be used to interpolate the joint trajectories of skeletons if increased computational efficiency is desired. For example, one could use the well-known minimum geodesic method on $\mathrm{SE}(3)$ [23]. However, the increase in computational efficiency would likely be offset by a significant decrease in accuracy.

As a next step, we plan to explore capabilities of GORA framework in providing a foundation for a more robust action recognition algorithm able to minimize or eliminate noise and nuisance parameters from signals while simultaneously reparameterizing them to a UST [6]. The development of such an algorithm able to inherently compensate for perturbations such as noise or motion artifacts while maintaining a linear complexity similar to that of GORA would mark an important milestone toward the goal of robust robotic action recognition in real-time.

VI. CONCLUSIONS

In this paper, we introduced a variant of the *Globally Optimal Reparameterization Algorithm* for signal alignment and action recognition with skeleton sequences, which we call GORA-S. This algorithm reparameterizes skeletons to a *universal standard timescale* (UST), allowing for element-wise comparisons between skeletons at each time instance with a linear complexity of $O(T)$. Additionally, we review the parameters and numerical techniques used in its application.

Our experimental results suggest that GORA-S has the potential to become a viable alternative to DTW methods for signal alignment and action recognition with skeleton sequences. In particular, we show that the computational complexity of GORA-S is less than that of the FastDTW algorithm with radius $= 1$ while maintaining a similar degree of accuracy. More importantly, GORA-S displays a favorable balance between speed and accuracy that remains approximately invariant under changes in the temporal sampling frequency of the skeletons, suggesting it has the potential to be a versatile algorithm well-suited for a variety of different action recognition related tasks.

VII. ACKNOWLEDGMENTS

The authors would like to thank Dr. Jin Seob Kim for useful discussions, Mr. Can Kocabalkanli and Ms. Yuqing (Eva) Pan for pictures of the NAO robot, and Mr. Mark Shifman and Mr. Okan Kocabalkanli for help in pre-processing data. This work was performed under National Science Foundation grant IIS-1619050 and Office of Naval Research Award N00014-17-1-2142.

REFERENCES

[1] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, "Humanoid robot HRP-3," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2471–2478.

[2] M. Ma, N. Marturi, Y. Li, A. Leonardis, and R. Stolkin, "Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos," *Pattern Recognition*, vol. 76, pp. 506–521, 2018.

[3] A. Akula, A. K. Shah, and R. Ghosh, "Deep learning approach for human action recognition in infrared images," *Cognitive Systems Research*, vol. 50, pp. 146–154, 2018.

[4] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2018.

[5] T. W. Mitchel, S. Ruan, Y. Gao, and G. S. Chirikjian, "The Globally Optimal Reparameterization Algorithm: an alternative to Fast Dynamic Time Warping for action recognition in video sequences," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov 2018, pp. 1–8.

[6] 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

[7] 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.

[8] 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.

[9] 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.

[10] 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

[11] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a Lie group," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 588–595.

[12] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+ D: A large scale dataset for 3D human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

[13] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Data Engineering, 2002. Proceedings. 18th International Conference on*. IEEE, 2002, pp. 673–684.

[14] S.-Y. Jin and H.-J. Choi, "Essential body-joint and atomic action detection for human activity recognition using longest common subsequence algorithm," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 148–159.

[15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Ieee, 2011, pp. 1297–1304.

[16] G. S. Chirikjian and A. B. Kyatkin, *Harmonic Analysis for Engineers and Applied Scientists: Updated and Expanded Edition*. Courier Dover Publications, 2016.

[17] C. Belta and V. R. Kumar, "An SVD-based projection method for interpolation on SE(3)," *IEEE Trans. Robotics and Automation*, vol. 18, pp. 334–345, 2002.

[18] B. Fornberg, "Generation of finite difference formulas on arbitrarily spaced grids," *Mathematics of computation*, vol. 51, no. 184, pp. 699–706, 1988.

[19] ——, "Classroom note: Calculation of weights in finite difference formulas," *SIAM review*, vol. 40, no. 3, pp. 685–691, 1998.

[20] B. Dahlgren and O. Certik, "bjodah/finitediff: finitediff-0.6.2," Jun. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1299173

[21] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[22] K. Tanida, "slaypni/fastdtw: fastdtw-0.3.2," Jul. 2017. [Online]. Available: https://pypi.org/project/fastdtw/

[23] M. Zefran, V. Kumar, and C. Croke, "Choice of Riemannian metrics for rigid body kinematics," in *ASME 24th Biennial Mechanisms Conference*, vol. 2, 1996.