Multimed Tools Appl (2014) 72:3045–3062
DOI 10.1007/s11042-013-1591-9

Robust gesture recognition using feature pre-processing and weighted dynamic time warping

Tarik Arici · Sait Celebi · Ali S. Aydin · Talha T. Temiz

Published online: 17 July 2013

Abstract Gesture recognition is a technology often used in human-computer interaction applications. Dynamic time warping (DTW) is one of the techniques used in gesture recognition to find an optimal alignment between two sequences. Oftentimes a pre-processing of sequences is required to remove variations due to different camera or body orientations or due to different skeleton sizes between the reference gesture sequences and the test gesture sequences. We discuss a set of pre-processing methods to make the gesture recognition mechanism robust to these variations. DTW computes a dissimilarity measure by time-warping the sequences on a per sample basis by using the distance between the current reference and test sequences. However, all body joints involved in a gesture are not equally important in computing the distance between two sequence samples. We propose a weighted DTW method that weights joints by optimizing a discriminant ratio. Finally, we demonstrate the performance of our pre-processing and the weighted DTW method and compare our results with the conventional DTW and state-of-the-art.

Keywords Gesture recognition · Dynamic time warping · Kinect

T. Arici (✉) · S. Celebi · A. S. Aydin · T. T. Temiz
Department of Electrical Engineering, Istanbul Sehir University,
Kusbakisi Caddesi No: 27 34662, Uskudar, Istanbul, Turkey
e-mail: tarikarici@sehir.edu.tr

S. Celebi
e-mail: saitcelebi@std.sehir.edu.tr

A. S. Aydin
e-mail: aliaydin@std.sehir.edu.tr

T. T. Temiz
e-mail: talhatemiz@std.sehir.edu.tr

1 Introduction

Interacting with computers using human motion is commonly employed in human-computer interaction (HCI) applications. One way to incorporate human motion into HCI applications is to use a predefined set of human joint motions i.e., gestures. Gesture recognition has been an active research area [12, 19, 26, 39], and involves state-of-the-art machine learning techniques in order to work reliably in different environments. A variety of methods have been proposed for gesture recognition including Dynamic Time Warping [26], Hidden Markov Models [12], Finite State Machines [13], hidden Conditional Random Fields (CRFs) [35] and orientation histograms [11]. In addition to these, there are methods employed in gesture recognition that are not view-based. Examples of these are the use of Wii controller (Wiimote) [29] and DataGlove [23].

DTW measures similarity between two time sequences which might be obtained by sampling a source with varying sampling rates or by recording the same phenomenon occurring with varying speeds [37]. The conventional DTW algorithm is basically a dynamic programming algorithm, which uses an iterative update of DTW cost by adding the distance between mapped elements of the two sequences at each iteration step. The distance between two elements is oftentimes the Euclidean distance, which gives equal weights to all dimensions of a sequence sample. However, depending on the problem a weighted distance might perform better in assessing the similarity between a test sequence and a reference sequence. For example in a typical gesture recognition problem, body joints used in a gesture can vary from gesture class to gesture class. Hence, not all joints are equally important in recognizing a gesture.

We propose a weighted DTW algorithm that uses a weighted distance in the cost computation. The weights are chosen so as to maximize a discriminant ratio based on DTW costs. The weights are obtained from a parametric model which depends on how active a joint is in a gesture class. The model parameter is optimized by maximizing the discriminant ratio. By doing so, some joints will be weighted up and some joints will be weighted down to maximize between-class variance and minimize within-class variance. As a result, irrelevant joints of a gesture class (i.e., parts that are not involved in a gesture class) will contribute to the DTW cost to a lesser extent, while keeping the between-class variances large.

Our system first extracts body-joint features from a set of skeleton data that consists of six joint positions, which are left and right hands, wrists and elbows. We have observed that the gestures in our training set, which have quite different motion patterns, require the use of all or a subset of these six joints only. These obtained skeleton features are used to recognize gestures by matching them with pre-stored reference sequences. Pre-processing is needed to suppress the noise due to different body and camera orientations, and different body sizes. After pre-processing is done, the matching is performed by assigning a test sequence to a reference sequence with the minimum DTW cost. By removing the variations in the data, the DTW cost becomes more reliable in classification as demonstrated by the increase in the discriminant ratio values.

2 Related work

One commonly used technique for gesture recognition is using HMMs for modeling gesture sequences. HMMs are especially known for their application to speech recognition, gesture recognition, etc. HMMs are statistical models for sequential data [3, 4], and therefore can be used in gesture recognition [12, 18, 32]. The states of an HMM are hidden and state transition probabilities are to be learned from the training data. However, defining states for gestures is not an easy task since gestures can be formed by a complex interaction of different joints. Also, learning the model parameters i.e., transition probabilities, requires large training sets, which may not always be available. On the other hand, DTW does not require training but needs good reference sequences to align with.

After DTW was introduced in 1960s [5], it has been used in solving different problems such as speech recognition to warp speech in time to be able to cope with different speaking speeds [2, 22, 28], data mining and information retrieval to deal with time-dependent data [1, 24], curve matching [10], online handwriting recognition [34], hand shape classification [17]. In gesture recognition, DTW time-warps an observed motion sequence of body joints to pre-stored gesture sequences [9, 17, 25, 36]. Although we present the theory of the general DTW and its implementation issues, in this paper we focus more on its application to gesture recognition. Comprehensive surveys about the general DTW algorithm can be found in [21, 30]. This work is the extended version of our work in [7].

Using a weighting scheme in DTW cost computation has been proposed for gesture recognition [26]. The method proposed in [26] uses DTW costs to compute between-class and within-class variations to find a weight for each body joint. These weights are global weights in the sense that there is only one weight computed for a body joint. However, our proposed method computes a weight for each body joint and for each gesture class. This boosts the discriminative power of DTW costs since a joint that is active in one gesture class may not be active in another gesture class. Hence weights has to be adjusted accordingly. This helps especially dealing with within-class variation. To avoid reducing the between-class variance, we compute weights by optimizing a discriminant ratio using a parametric model that depends on body joint activity. Another type of weighting in DTW for aligning time series is proposed in [15]. Their goal is to modify DTW so that the similarity between two 1D time series is robust to outliers. An outlier at a particular time instant can create a large error, which dominates distances in other time instants. To avoid this, a robust distance function instead of the L_1 norm (i.e., absolute distance), or the L_2 norm (i.e., Euclidean distance) is used. Hence, the weighting is for different distance values between the two samples of 1D time series. However, we propose to weight each dimension of a multi-dimensional signal, where each dimension is a joint position. This work is complimentary to our work in the sense that a robust distance function that penalizes the outliers to a lesser degree, can also be used in our method.

The goal of dynamic time warping is the alignment of two time sequences via a dynamic cost minimization. The final DTW cost, which is a dissimilarity measure between two time sequences, is also used for classification. Oftentimes, a test sequence

is aligned to a set of templates via DTW and the test sequence is matched to the minimum cost template. A novel approach is presented in [20], to separate the alignment and classification tasks. First, alignment is performed using DTW. The aligned sequences are classified by using feature generation methods from Machine Learning theory. Our proposed method can be used in the alignment phase of the technique proposed in [20].

With Microsoft's launch of Kinect in 2010, and release of Kinect SDK in 2011, numerous applications and research projects exploring new ways in human-computer interaction have been enabled. Some examples are gesture recognition [26], touch detection using depth data [38], human pose estimation [14], implementation of real-time virtual fixtures [27], real-time robotics control applications [33] and the physical rehabilitation of young adults with motor disabilities [8]. In the next section we discuss data acquisition and feature pre-processing.

3 Data acquisition and feature pre-processing

We use Microsoft Kinect sensor [31] to obtain joint positions. Kinect SDK tracks 3D coordinates of 20 body joints given in Fig. 1 in real time (30 frames per second). The Kinect algorithm uses depth images to predict joint positions and the predicted joint positions are quite robust to color, texture, and background.

In our experiments we have focused on hand-arm gestures. Six out of the 20 joints available in Kinect's skeleton model are informative in recognizing a hand-arm gesture, which are left hand, right hand, left wrist, right wrist, left elbow and right elbow joints. However, there is no limitation on the number of joints used in our proposed method. For hand-arm gestures the relevant body joints are obvious,

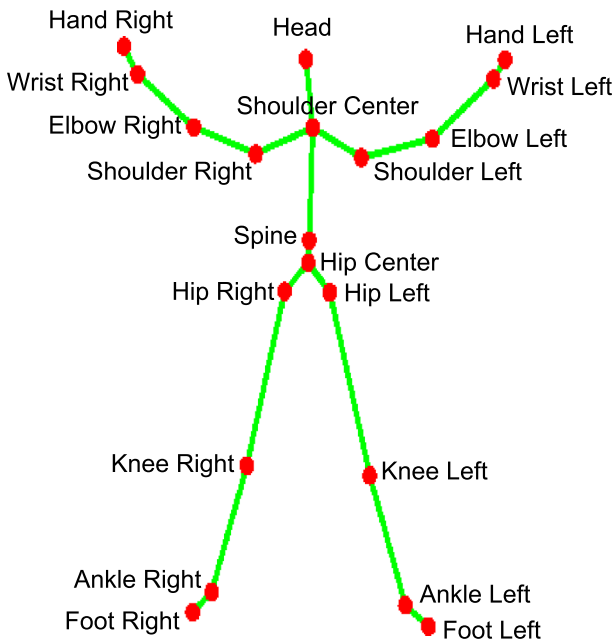

Fig. 1 Kinect joints

but for more complex gestures the most informative joints for the recognition task can be selected by using feature selection techniques from the classification literature in machine learning [6]. For example Sequential Backward Elimination (SBE) technique, which starts with all the 20 joints, and eliminates joints one by one based on the discriminant ratio change can be utilized.

In our method, a feature vector consists of 3D coordinates of these six joints and is of dimension of 18 as given below

$$\mathbf{f}_n = [X_1, Y_1, Z_1, X_2, Y_2, Z_2, \ldots X_6, Y_6, Z_6], \tag{1}$$

where n is the index of the skeleton frame at time t_n. A gesture sequence is the concatenation of N such feature vectors.

After N feature vectors are concatenated to create the gesture sequence, they are pre-processed before the DTW cost computation. The pre-processing consists of three stages. First stage is the normalization stage which translates all skeletons to the center of the field of view. This could be done by subtracting the hip center joint position from the other joint positions. Note that the reference frames are already recorded at the center of the field of view. The second pre-processing stage removes the rotational distortion caused by different orientations of human bodies. Contrary to the reference gestures, where trained performers are used, it is highly possible to have different orientations or positionings of users with respect to camera in real-life cases. Such occasions are problematic for gesture recognition since they will result in rotationally distorted skeleton frames (See Fig. 2). To cope with these occasions, our pre-processing system rotates the skeleton frames if necessary, such that the skeleton frames will be orthogonal to the principal axis of the camera. To this end, we define two vectors by using spatial coordinates of the right shoulder, left shoulder and hip center which are obtained from Kinect sensor. One of the vectors is defined from the midpoint of right and left shoulder to hip center, while the other vector is defined from the same midpoint to the right shoulder. Using these two vectors, we calculate the three angles, α, β, θ, of the skeleton with respect to the camera's coordinate system, and compute the rotation matrices $\mathbf{R}_\mathbf{x}^\alpha$, $\mathbf{R}_\mathbf{y}^\beta$, $\mathbf{R}_\mathbf{z}^\theta$, respectively. The

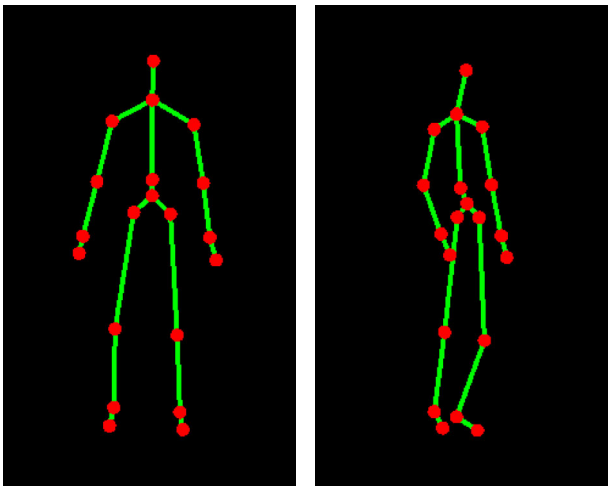

Fig. 2 Two skeletons with different orientations (*left*: ground-truth reference frame, *right*: rotationally distorted test frame due to improper body orientation)

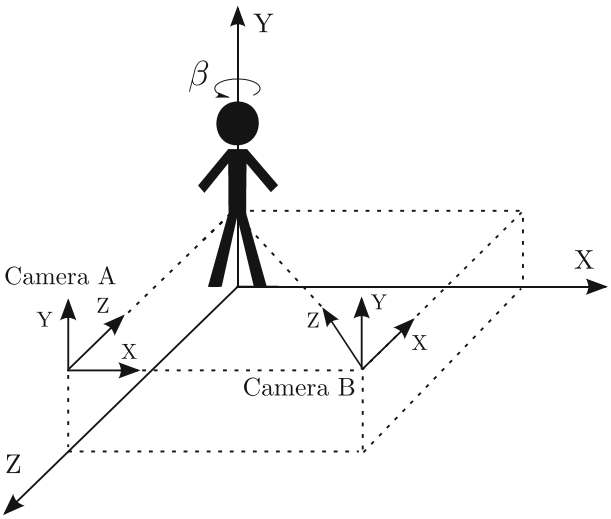

Fig. 3 *Camera A* is used to record the ground-truth reference gestures with perpendicular angles, *Camera B* is used to record a rotationally distorted test sequence. β is the desired angle to rotate the skeleton in Y axis. After this rotation, the skeleton will be rotated in other axes if needed until it will be perpendicular to all axes

rotation is then applied using these angles with the appropriate order. See an example rotation in Y axis with $\mathbf{R}_{\mathbf{y}}^{\beta}$ in Fig. 3. The third and the last pre-processing stage is the elimination of variations in the feature vectors due to different skeleton ratios (broad-shouldered, narrow-shouldered). All feature vectors are normalized with the distance between the left and the right shoulders to account for the variations due to a person's *size*. Note that the reference sequences are recorded with people who has average skeleton ratios. Next, we present a more detailed discussion on DTW.

4 Dynamic time warping for gesture recognition

DTW is a template matching algorithm to find the best match for a test pattern out of the reference patterns, where the patterns are represented as a time sequence of features. In Fig. 4 we show an example matching of two sequences.

Let $\mathbf{R} = \{r_1, r_2, \ldots, r_N\}$, $N \in \mathbb{N}$ and $\mathbf{T} = \{t_1, t_2, \ldots, t_M\}$, $M \in \mathbb{N}$ be reference and test sequences (sequence of set of joint positions in our case), respectively. The objective is to align the two sequences in time via a nonlinear mapping. Such a warping path can be illustrated as an ordered set of points as given below

$$p = (p_1, p_2, \ldots, p_L), \;\; p_l = (n_l, m_l),$$

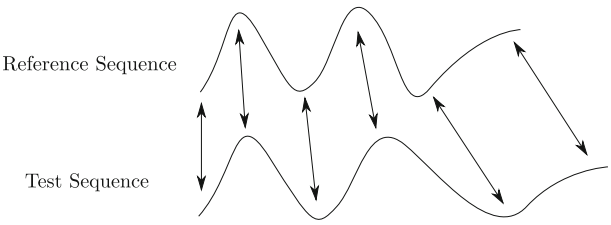

Fig. 4 DTW used to match two sequences, reference sequence and test sequence

where $p_l = (n_l, m_l)$, denotes mapping of r_{n_l} to t_{m_l}. $p_l \in [1:N] \times [1:M]$ for $l \in [1:L]$, where L is the number of mappings. The total cost D of a warping path p between **R** and **T** with respect to a *distance* function $d(r_i, t_j), i \in [1:N]$ and $j \in [1:M]$, is defined as the sum of all distances between the mapped sequence elements

$$\mathrm{D_p} = \sum_{l=1}^{L} d(r_{n_l}, t_{m_l}), \tag{2}$$

where D_p is the total cost of the path p and $d(r_i, t_j)$ measures the distance between elements r_i and t_j. For gesture recognition, distance can be chosen as the distance between the corresponding joint positions (3D points) of the reference gesture, **R**, and the test gesture **T**.

A mapping can also be viewed as a path on a two-dimensional (2D) grid, also known as the cost matrix, which is of size $N \times M$ (see Fig. 5), where grid node (r_i, t_j) denotes the distance between r_i and t_j. The node (r_1, t_1) which starts the alignment by matching the first sequence elements is conventionally placed on the left-bottom corner of the grid. Each path p on the 2D grid (i.e., the cost matrix) is associated with a total cost D given in (2). Note that among all possible paths, we are mostly interested in the path which makes the total accumulated cost minimum while satisfying the desired constraints. Hence, optimal path denoted by p^* is the path with the minimum total cost. The DTW distance between two sequences is defined by the distance associated with a total cost D given in (2) using the optimal path, i.e.:

$$\mathrm{DTW}(\mathbf{R}, \mathbf{T}) = \mathrm{D_{p^*}}(\mathbf{R}, \mathbf{T}). \tag{3}$$

Some well-known restrictions on the warping path have been proposed to eliminate unrealistic correspondences between the sequences [21, 28]. The most fundamental constraints which are applied in various topics as well as gesture recognition, are the following:

(i) Boundary conditions: $p_1 = (1, 1)$, $p_L = (N, M)$.
(ii) Step size condition: $p_{l+1} - p_l \in \{(0, 1), (1, 0), (1, 1)\}$ for $l \in [1 : L - 1]$.

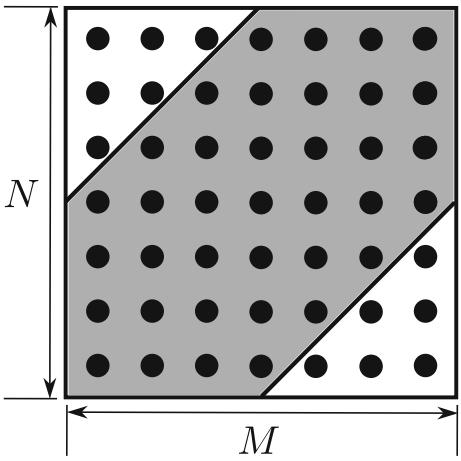

Fig. 5 Accumulated cost matrix of two sequences **R** and **T** with sizes N and M, respectively. Global constraint region, R, *Sakoe–Chiba band* [28], is shown with *gray color*

The boundary conditions require the whole reference sequence to be mapped to the whole test sequence, and can be modified if this is not strictly desired. The step size condition requires that only one element of both sequences can be skipped at each cost computation step of Bellman's principle. Hence, optimal path can progress from a restricted set of predecessor nodes as shown in Fig. 6. Since all the elements are ordered in time, the set of predecessor nodes are to the left and bottom of a current node.

First, let's define $C(n_l, m_l)$ as below

$$\mathrm{C}(\mathrm{n_l}, \mathrm{m_l}) = \mathrm{DTW}(\mathbf{R}(1 : n_l), \mathbf{T}(1 : m_l)). \tag{4}$$

Note that C(N, M) is equal to DTW($\mathbf{R}$, $\mathbf{T}$). Let's further assume that the total costs of the optimal paths to three predecessor nodes denoted by $(n_l - 1, m_l)$, $(n_l, m_l - 1)$, and $(n_l - 1, m_l - 1)$ have been computed. Since the $(l-1)$th position of the path (i.e., (n_{l-1}, m_{l-1})) is restricted to be one of these three nodes on the 2D grid, Bellman's principle leads to

$$\begin{aligned}\mathrm{C}(\mathrm{n_l}, \mathrm{m_l}) = \min\{&\mathrm{C}(\mathrm{n_l}, \mathrm{m_l} - 1),\\ &\mathrm{C}(\mathrm{n_l} - 1, \mathrm{m_l}),\\ &\mathrm{C}(\mathrm{n_l} - 1, \mathrm{m_l} - 1)\} + d(r_{n_l}, t_{m_l}).\end{aligned} \tag{5}$$

Finally, the minimum cost path aligning two sequences has cost C(N, M) = DTW($\mathbf{R}$, $\mathbf{T}$), and the test sequence is matched to the reference sequence that has the minimum cost among all reference sequences.

Although (5) outputs the minimum cost between two sequences, it does not output the optimal path. To find the optimal path, which can be used to map test sequence elements to reference sequence elements, one needs to backtrack the optimal path starting with the final node. Note that if the boundary condition is satisfied, i.e., the

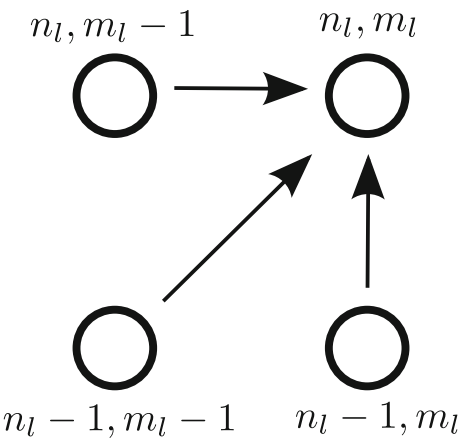

Fig. 6 Predecessor nodes used in Bellman's principle where $n_l \in [1 : N]$, $m_l \in [1 : M]$ and $l \in [2 : L]$. Note that $(n_{l-1}, m_{l-1}) \in \{(n_l - 1, m_l), (n_l, m_l - 1), (n_l - 1, m_l - 1)\}$

whole test sequence is mapped to the whole reference sequence, than $(n_L, m_L) = (N, M)$ and $(n_1, m_1) = (1, 1)$.

4.1 Boosting the reliability of DTW

Global constraints define a set of nodes on the 2D grid to be searched for finding the optimal path. Imposing global constraints not only reduces the DTW computational complexity, but also increases the reliability of DTW's dissimilarity measure by omitting unrealistic paths. We used a well-known global constraint region, *Sakoe–Chiba band* [28] given in Fig. 5. The Sakoe–Chiba band effectively limits the warping amount, i.e., slowing down or speeding up of a sequence in time. For example a gesture can be performed with different speeds in time depending on the performer but it is logical to expect that there is a limit to how slow or how fast a gesture is performed.

Another problem that degrades DTW's reliability in gesture recognition is due to unknown beginning and ending times of gesture samples. A gesture in a test sequence can often begin later or end sooner than the gesture in the reference sequence stored for that gesture class. Boundary conditions assume that all gestures start at the beginning of the sequence and finish at the ending of the sequence. Hence, imposing boundary conditions in such cases decreases the reliability of DTW costs. To boost the reliability, we relaxed the boundary conditions by changing the total cost given in (2) as below

$$\mathrm{D_p} = \sum_{l=1}^{L} \alpha_l d(r_{n_l}, t_{m_l}), \tag{6}$$

where α_l is a weight that is equal to 1 everywhere except the regions close to the starting node (i.e., left-bottom node denoted by (r_1, t_1)) and the ending node (i.e., right-top node denoted by (r_N, t_M)). To infer the proximity of the current node to starting and ending nodes the length of the path, $||p_l|| = \sqrt{n_l^2 + m_l^2}$, is utilized. The distance terms coming from the beginning and ending of the sequence is weighted down by computing α_l from the below formula

$$\alpha_l = \begin{cases} \dfrac{||p_l||}{\tau} & \text{if } ||p_l|| < \tau \\ \dfrac{L - ||p_l||}{\tau} & \text{if } L - ||p_l|| < \tau \\ 1 & \text{otherwise,} \end{cases} \tag{7}$$

where L is the length of the longest path and τ is a threshold value.

4.2 Weighted DTW

The conventional DTW computes the dissimilarity between two time sequences by aligning the two sequences based on a sample based distance as in (5). If the sequence samples are multi-dimensional (18 dimensional for the gesture recognition problem),

using an Euclidean distance gives equal importance to all dimensions. We propose to use a weighted distance in the cost computation based on how relevant a body joint is to a specific gesture class. The relevancy is defined as the contribution of a joint to the motion pattern of that gesture class. To infer a joint's contribution to a gesture class we compute its total displacement (i.e., contribution) during the performance of that gesture by a trained user by

$$C_j^g = \sum_{n=2}^{N} Dist^j(\mathbf{f}_{n-1}^g, \mathbf{f}_n^g), \tag{8}$$

where g is the gesture index, j is the joint index and n is the skeleton frame number. $Dist^j()$ computes the displacement of jth joint's two consecutive coordinates in feature vectors $\mathbf{f}_{n-1}^g$, and $\mathbf{f}_n^g$. By summing up these consecutive displacements one can find the total displacement of a joint in a selected reference gesture.

After the total displacements are calculated, we filter out the noise (e.g, shaking, trembling) and threshold them from the bottom and the top. This prevents our parametric weight model to output too high or low weights as given below

$$C_j^g = \begin{cases} C_a & \text{if } 0 \le C_j^g < T_1 \\ \dfrac{C_j^g - T_1}{T_2 - T_1}(C_b - C_a) + C_a & \text{if } T_1 \le C_j^g < T_2 \\ C_b & \text{otherwise,} \end{cases} \tag{9}$$

where C_a and C_b are threshold values, and T_1 and T_2 are experimentally determined boundary values for threshold assignment.

Using the total displacement to asses the contribution of a joint in performing a gesture, the weights of gesture class g are calculated via

$$w_j^g = \frac{1 - e^{-\beta C_j^g}}{\sum_k \left(1 - e^{-\beta C_k^g}\right)}, \tag{10}$$

where w_j^g is joint j's weight value for gesture class g. Note that in this formulation a joint's weight value can change depending on the gesture class. For example, for the right-hand-push-up gesture, one would expect the right hand, right elbow and right wrist joints to have large weights, but to have smaller weights for the left-hand-push-up gesture.

To incorporate these weights into the cost, the distance function $d(r_n, t_m)$ becomes a weighted average of joints distances between two consecutive frames and is defined to be

$$d(r_n, t_m) = \sum_j Dist^j(r_n, t_m) w_j^g, \tag{11}$$

which gives the distance between nth skeleton frame of reference gesture $\mathbf{R}$ and mth skeleton frame of test gesture $\mathbf{T}$, where $\mathbf{R}$ is a sequence known to be in gesture class g and $\mathbf{T}$ is an unknown test sequence.

The weights are obtained from the model given in (10), which has a single parameter β. Our objective is to choose a β value that minimizes the within-class variation while between-class variation is maximized. Between-class variation maximization and within-class variation minimization can be achieved by making irrelevant joints contribute less to the cost (e.g., reducing the weights of right hand in left-hand-push-up gesture) and not reducing (or possibly increasing) the weights of joints that can help to discriminate different gestures. We try to achieve this goal by maximizing a discriminant ratio similar to Fisher's Discriminant Ratio [16]. To this end, we define $D_{g,h}(\beta)$, as the average weighted DTW cost between all samples of gesture class g and gesture class h using weights calculated with β. Then between-class dissimilarity is the average of all $D_{g,h}(\beta)$'s ($h \neq g$) as the following:

$$D_B(\beta) = \sum_g \sum_{\substack{h \\ h \neq g}} D_{g,h}(\beta), \tag{12}$$

which measures the sum of average distances between gesture classes. This helps us infer the average distance between a gesture and the rest of the gestures for a given β.

Within-class dissimilarity is the sum of within-class distance $D_{g,g}(\beta)$ for all gesture classes,

$$D_W(\beta) = \sum_g D_{g,g}(\beta), \tag{13}$$

which sums the average distance $D_{g,g}(\beta)$ between the samples of gesture classes for all g.

The discriminant ratio of a given β, $R(\beta)$, is then obtained by

$$R(\beta) = \frac{D_B(\beta)}{D_W(\beta)}. \tag{14}$$

The optimum β, β^*, is chosen as the one that maximizes R:

$$\beta^* = \arg\max_{\beta} R(\beta). \tag{15}$$

5 Results

Our experiments were performed on our gesture database which was recorded with 38 participants. It took approximately one week to finish all the recordings. All participants performed 12 different gestures with six samples per gesture class. Bad records due to a bad gesture performance (e.g., incomplete gesture) or Kinect's human-pose recognition failure, correspond to approximately 30 % percentage of all recorded gestures. They were manually deleted by using an OpenGL based gesture visualizer. The physical factors (e.g., distance from the Kinect sensor to the user, illumination in the room) are kept constant during the recording of all records. Each gesture sample includes 20 joint position data per frame in addition to time stamps of each skeleton frame. The gesture databases used in the experiments, source code for visualization of gestures, source code used to produce the results in this paper and

more results are publicly available.[1] We are hoping that the databases can be used in testing other gesture recognition algorithms as well.

We tested the performance of our feature pre-processing technique and proposed weighting method on our three discrete gesture databases to show the improvements separately: (i) *Rotationally distorted gesture database*: In this database we recorded a set of noisy gestures in terms of the rotational orientation of the body with respect to the Kinect sensor in X, Y and Z axes (See Fig. 3). The gestures are performed by trained users. This database is designed in order to see the effect of pre-processing on the recognition performance. It has 12 different gesture classes and 21 gesture samples per gesture class. (ii) *Relaxed gesture database*: In this database there is no intentionally generated rotational distortion, instead, these gesture samples are performed more relaxed in terms of the movement of body parts other than the active joints involved in gesture performance. For example in one sample of this database, performer scratches his head with his left hand while he performs the right-hand-push-up gesture. This database has 8 gesture classes and *1116* gesture samples in total. (iii) *Rotationally distorted and relaxed gesture database*: In this database performers recorded gestures *relaxed* in terms of both rotation and body movement. This database has 12 gesture classes and *198* gesture samples in total. We use this database to show the overall performance of the system. All the three databases are created using Microsoft Kinect Sensor.

In addition to these databases, there is a set of reference samples per gesture class, performed properly by trained users without any rotational distortion and without any undesired movements. These reference samples are used in learning the total distance measures of each joint in each class, which is required by our weight model in (10). Two sample reference gestures are shown in Fig. 7.

In the first experiment, we test our pre-processing method using the rotationally distorted gesture database. We first calculated the discriminant ratios (See (14)) of 21 samples for each 12 gesture class without using any of the pre-processing methods. Then, we used the same gesture samples to calculate the discriminant ratios again, but this time using our proposed pre-processing methods. Note that uniform weights were used in order to see the performance of the pre-processing method alone. The improved achieved by pre-processing can be seen in Fig. 8.

In the second experiment we compared our weighted DTW algorithm against the conventional DTW method and a weighted DTW method proposed by [26] using the relaxed gesture database. The confusion matrices for the three algorithms for six chosen gesture classes are given in Tables 1, 2, and 3. Note that the recognition rates for these classes are consistent with recognition rates of other classes(i.e. classes that are not presented in the confusion matrix), but only these classes are shown for the sake of brevity.

After creating the confusion matrices, we computed the overall recognition accuracies according to the following formula:

$$A = 100 \cdot \frac{\text{Trace}(C)}{\sum_{i=1}^{m} \sum_{j=1}^{n} C(i,\, j)}, \tag{16}$$

where A denotes the accuracy, and C denotes the confusion matrix.

[1] http://mll.sehir.edu.tr/mtap2013

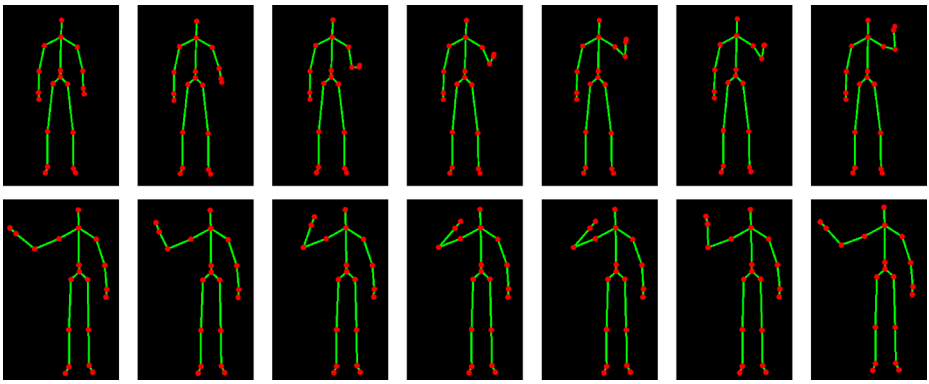

Fig. 7 Two sample reference gestures in the gesture database: *Right Hand Push Up* and *Left Hand Wave*

Our proposed method outperforms the weighted DTW method in [26] by a large margin as given in Table 4. The reason is that their weights are global weights, i.e., a joint's weight is independent of the gesture class. However, in our proposed method a joint can have a different weight depending on the gesture class we are trying to align with. This degree of freedom in computing the associated DTW cost increases the reliability of DTW cost significantly.

In the third and the last stage, we tested the overall performance of our system using the rotationally distorted and relaxed gesture database. The purpose of this operation is to determine the overall improvement of the pre-processing and the weighting on the recognition performance using a larger database. These experiments clearly demonstrate the performance boost provided by our proposed techniques. The results are given in Table 5.

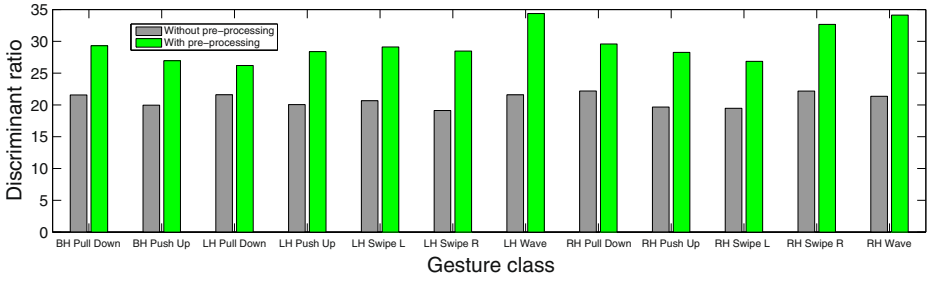

Fig. 8 Discriminant ratios for with and without pre-processed gesture samples using the rotationally distorted gesture database. Note that the discriminant ratios are increased, on average, 42 % with the proposed pre-processing method. There are 21 gesture samples in each gesture class. The gesture classes are, namely, *Both Hands Pull Down*, *Both Hands Push Up*, *Left Hand Pull Down*, *Left Hand Push Up*, *Left Hand Swipe Left*, *Left Hand Swipe Right*, *Left Hand Wave*, *Right Hand Pull Down*, *Right Hand Push Up*, *Right Hand Swipe Left*, *Right Hand Swipe Right*, *Right Hand Wave*, respectively

Table 1 Confusion matrix for the conventional DTW

	RH push up	LH push up	RH pull down	LH pull down	RH swipe L	RH swipe R
RH push up	93.9	0	0	2.3	3.8	0
LH push up	2.4	94.6	0.6	0	2.4	0
RH pull down	0	0	98.6	1.4	0	0
LH pull down	2	0	0.7	97.3	0	0
RH swipe L	0	0.8	0	4	95.2	0
LH swipe R	5.6	0	2.1	22.6	0.7	69

Table 2 Confusion matrix for the weighted DTW in [26]

	RH push up	LH push up	RH pull down	LH pull down	RH swipe L	RH swipe R
RH push up	96.2	1.5	0	0.8	1.5	0
LH push up	3	97	0	0	0	0
RH pull down	0	1.4	98.6	0	0	0
LH pull down	2	0	0	98	0	0
RH swipe L	0	2.4	0	2.4	95.2	0
LH swipe R	7.8	0	0	25.3	0.7	66.2

Table 3 Confusion matrix for our proposed weighted DTW

	RH push up	LH push up	RH pull down	LH pull down	RH swipe L	RH swipe R
RH push up	100	0	0	0	0	0
LH push up	0	100	0	0	0	0
RH pull down	0	0	100	0	0	0
LH pull down	0	0	0	100	0	0
RH swipe L	0.8	0	0	0	99.2	0
LH swipe R	0	0	0	0	2.8	97.2

Table 4 Accuracies of the three methods

Method	Accuracy
Classical DTW	84.41 %
State-of-the art	86.56 %
Proposed method	97.13 %

Note that not only six gesture classes given in Tables 1–3 are used, but all eight gesture classes are taken into consideration

Table 5 Overall performance comparison using the rotationally distorted and relaxed gesture database

Method	Accuracy (%)
Traditional DTW	62.41
Pre-processing + traditional DTW	76.26
Weighted DTW	84.13
Pre-processing + weighted DTW	96.64

6 Conclusion

We have developed a weighted DTW method to boost the discrimination capability of DTW's cost, and shown that the performance increases significantly. The weights are based on a parametric model that depends on the level of a joint's contribution to a gesture class. The model parameter is optimized by maximizing a discriminant ratio, which helps to minimize within-class variation and maximize between-class variation. We have also developed a pre-processing method to cope with real life situations, where different body shapes and user orientations with respect to the depth sensor may occur. Our weighted DTW, enables *noise* in skeleton joints as long as they do not make a gesture of one class similar to a gesture of another class. This is because weights are selected by maximizing between-class variation. We hope that the proposed method will enable more natural remote control of different devices using pre-defined commands for a given context/situation. As long as the *noise* in a joint does not overlap with another gesture class, the user is free to naturally use his/her other joints.

Acknowledgements We would like to thank to all Sehir University student who participated in our gesture database recordings and patiently performed all gestures to help our experiments.

References

1. Adams NH, Bartsch MA, Shifrin J, Wakefield GH (2004) Time series alignment for music information retrieval. In: ISMIR
2. Amin TB, Mahmood I (2008) Speech recognition using dynamic time warping. In: International conference on advances in space technologies. doi:10.1109/ICAST.2008.4747690
3. Baum L (1972) An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities 3:1–8
4. Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41:164–171. doi:10.1214/aoms/1177697196
5. Bellman R, Kalaba R (1959) On adaptive control processes. IRE Trans Autom Control 4(2):1–9
6. Brodley CE, Utgoff PE (1995) Multivariate decision trees. Mach Learn 19(1):45–77
7. Celebi S, Aydin AS, Temiz TT, Arici T (2013) Gesture recognition using skeleton data with weighted dynamic time warping. In: Computer vision theory and applications, Visapp
8. Chang YJ, Chen SF, Huang JD (2011) A kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities. Res Dev Disabil 32(6):2566–2570. doi:10.1016/j.ridd.2011.07.002. http://www.sciencedirect.com/science/article/pii/S0891422211002587
9. Corradini A (2001) Dynamic time warping for off-line recognition of a small gesture vocabulary. In: Proceedings IEEE ICCV workshop on recognition, analysis, and tracking of faces and gestures in real-time systems, 2001. IEEE, pp 82–89
10. Efrat A, Fan Q (2007) Venkatasubramanian S Curve matching, time warping, and light fields: new algorithms for computing similarity between curves. J Math Imaging Vis 27(3):203–216
11. Freeman WT, Roth M (1994) Orientation histograms for hand gesture recognition. In: International workshop on automatic face and gesture recognition, pp 296–301
12. Gehrig D, Kuehne H, Woerner A, Schultz T (2009) Hmm-based human motion recognition with optical flow data. In: IEEE international conference on humanoid robots (Humanoids 2009), Paris, France
13. Hong P, Huang TS, Turk M (2000) Gesture modeling and recognition using finite state machines. In: Proceedings of the fourth IEEE international conference on automatic face and gesture recognition 2000, FG '00. IEEE Computer Society, Washington, DC, USA, p 410. http://dl.acm.org/citation.cfm?id=795661.796191
14. Jain HP, Subramanian A, Das S, Mittal A (2011) Real-time upper-body human pose estimation using a depth camera. In: Proceedings of the 5th international conference on computer

vision/computer graphics collaboration techniques, MIRAGE'11. Springer, Berlin, Heidelberg, pp 227–238. http://dl.acm.org/citation.cfm?id=2050320.2050340
15. Jeong YS, Jeong MK, Omitaomu OA (2011) Weighted dynamic time warping for time series classification. Pattern Recog 44(9):2231–2240
16. Kim SJ, Magnani A, Boyd SP (2005) Robust fisher discriminant analysis. In: Neural information processing systems
17. Kuzmanic A, Zanchi V (2007) Hand shape classification using dtw and lcss as similarity measures for vision-based gesture recognition system. In: EUROCON, 2007. The international conference on computer as a tool. IEEE, pp 264–269
18. Lee HK, Kim J (1999) An hmm-based threshold model approach for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21(10):961–973. doi:10.1109/34.799904
19. Liang R, Ouhyoung M (1998) A real-time continuous gesture recognition system for sign language. In: Proceedings third IEEE international conference on automatic face and gesture recognition, 1998. IEEE, pp 558–567
20. Lichtenauer JF, Hendriks EA, Reinders M (2008) Sign language recognition by combining statistical dtw and independent classification. IEEE Trans Pattern Anal Mach Intell 30(11):2040–2046
21. Müller M (2007) Information retrieval for music and motion, vol 6. Springer, Berlin
22. Myers CS, Habiner LF (1981) A comparative study of several dynamic time-warping algorithms for connected-word. Bell Syst Tech J
23. Quam D (1990) Gesture recognition with a dataglove. In: Proceedings of the IEEE 1990 national aerospace and electronics conference 1990, NAECON 1990, vol 2, pp 755–760. doi:10.1109/NAECON.1990.112862
24. Rath T, Manmatha R (2003) Word image matching using dynamic time warping. In: Proceedings IEEE computer society conference on computer vision and pattern recognition 2003, vol 2, pp. II–521–II–527. doi:10.1109/CVPR.2003.1211511
25. Rekha J, Bhattacharya J, Majumder S (2011) Shape, texture and local movement hand gesture features for indian sign language recognition. In: 3rd international conference on trendz in information sciences and computing, (TISC) 2011, pp 30–35. doi:10.1109/TISC.2011.6169079
26. Reyes M, Dominguez G, Escalera S (2011) Feature weighting in dynamic time warping for gesture recognition in depth data. In: IEEE international conference on computer vision workshops (ICCV Workshops) 2011, pp 1182–1188. doi:10.1109/ICCVW.2011.6130384
27. Ryden F, Chizeck HJ, Kosari SN, King H, Hannaford B (2011) Using kinect and a haptic interface for implementation of real-time virtual fixtures. In: Robotics sciences and systems, workshop on RGB-D: advanced reasoning with depth cameras, Los Angeles
28. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26(1):43–49. doi:10.1109/TASSP.1978.1163055
29. Schlömer T, Poppinga B, Henze N, Boll S (2008) Gesture recognition with a wii controller. In: Proceedings of the 2nd international conference on tangible and embedded interaction, TEI '08. ACM, New York, NY, USA, pp 11–14. doi:10.1145/1347390.1347395
30. Senin P (2008) Dynamic time warping algorithm review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA (2008)
31. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: CVPR, vol 2, p 7
32. Starner T, Pentland A (1996) Real-time american sign language recognition from video using hidden Markov models. In: International symposium on computer vision
33. Stowers J, Hayes M, Bainbridge-Smith A (2011) Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. In: IEEE international conference on mechatronics, (ICM) 2011, pp 358–362. doi:10.1109/ICMECH.2011.5971311
34. Tappert C, Suen C, Wakahara T (1990) The state of the art in online handwriting recognition. IEEE Trans Pattern Anal Mach Intell 12(8):787–808
35. Wang SB, Quattoni A, Morency LP, Demirdjian D, Darrell T (2006) Hidden conditional random fields for gesture recognition. In: IEEE computer society conference on computer vision and pattern recognition 2006, vol 2, pp 1521–1527. doi:10.1109/CVPR.2006.132
36. Wenjun T, Chengdong W, Shuying Z, Li J (2010) Dynamic hand gesture recognition using motion trajectories and key frames. In: 2nd international conference on advanced computer control, (ICACC) 2010, vol 3, pp 163–167. doi:10.1109/ICACC.2010.5486760
37. Wikipedia (2012) Dynamic time warping. http://en.wikipedia.org/wiki/Dynamic_time_warping. (online) Accessed 1 Aug 2008

38. Wilson AD (2010) Using a depth camera as a touch sensor. In: ACM international conference on interactive tabletops and surfaces, ITS '10. ACM, New York, NY, USA, pp 69–72. doi:10.1145/1936652.1936665.
39. Wilson AD, Bobick AF (1999) Parametric hidden Markov models for gesture recognition. IEEE Trans Pattern Anal Mach Intell 21:884–900. doi:10.1109/34.790429

Tarik Arici received his PhD degree in 2009 from the School of Electrical and Computer Engineering at Georgia Institute of Technology. During his PhD he has worked in VESTEL's Pixellence project and has designed algorithms that were patented in Europe and US. He has worked at NVIDIA in Silicon Valley between 2008 and 2010, where he designed and implemented patented algorithms that run on the GPU and is currently in NVIDIA's video drivers. He has international publications in the fields of image and video processing and sensor networks. After joining Istanbul Sehir University he has received EU Marie Curie CIG (Career Reintegration Grant).

Sait Celebi received his B.S. degree in Computer Engineering in Sakarya University. Now he is working towards his M.S. degree in Electrical and Computer Engineering in Istanbul Sehir University, where he is a research assistant in Machine Learning Lab.

Ali S. Aydin is an undergraduate research assistant in Istanbul Sehir University Machine Learning Lab. He is currently working towards a B.S. degree in Electrical Engineering. He has authored one conference and one journal publication.

Talha T. Temiz is in undergraduate studies at Istanbul Sehir University. During his BS years he worked at Machine Learning Lab in Istanbul Sehir University as Research Assistant. He has two international publications in field of Machine Learning.