A One-Shot DTW-Based Method for Early Gesture Recognition

Yared Sabinas, Eduardo F. Morales, and Hugo Jair Escalante

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro # 1, Tonantzintla, Puebla, México
{y.sabinas,emorales,hugojair}@inaoep.mx

Abstract. Early gesture recognition consists of recognizing gestures at their beginning, using incomplete information. Among other applications, these methods can be used to compensate for the delay of gesture-based interactive systems. We propose a new approach for early recognition of full-body gestures based on dynamic time warping (DTW) that uses a single example from each category. Our method is based on the comparison between time sequences obtained from known and unknown gestures. The classifier provides a response before the unknown gesture finishes. We performed experiments in the MSR-Actions3D benchmark and another data set we built. Results show that, in average, the classifier is capable of recognizing gestures with 60% of the information, losing only 7.29% of accuracy with respect to using all of the information.

Keywords: Early gesture recognition, DTW, one-shot learning, Kinect.

1 Introduction

The automated recognition of gestures has many applications in diverse fields, including video games, sign-language recognition and medical-monitoring systems, among others [5]. Very effective methods for gesture recognition are available nowadays, some of which require of specialized and expensive devices to capture gestures features. The Kinect sensor emerged recently and since then it has boosted the number of applications that make use of gesture recognition technology. This is due to the fact that this sensor is cheaper than similar devices, and provides useful data like RGB-D video and position of body joints (skeleton) in real time [11]. Most of the available methods for gesture recognition provide an answer once the gesture has finished. However, there are certain applications where the delay in gesture recognition is critical, e.g. in interactive and security systems. Despite the importance of this problem, called early gesture recognition, it has been scarcely explored [1,3,6,9].

This paper proposes a new method for early gesture recognition based on DTW using the Kinect sensor. Input sequences are compared with stored ones by using DTW, a prediction criterion is proposed to determine when the method is confident of the identity of the gesture depicted in input sequences. The proposed method can work under the one-shot learning framework [2], that is, using a

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part II, LNCS 8259, pp. 439–446, 2013.

single example of each gesture category to be recognized. This is advantageous for personalized and dynamic applications, where labeled data is scarce. Our method is easy to implement, it has no training phase and it is very efficient. We report results in a data set we built and in the MSR-Actions3D benchmark. Results show that the method can recognize gestures with 60% of the information, losing only 7.29% of accuracy with respect to using all of the information.

Early classification of gestures is a relatively young field; the first results were published in 2006 by Mori et al. [6]. They wanted to use the anticipated time to compensate the response delay of a robot that imitated their movements. This method was based on dynamic programming, and their gesture dictionary was composed by 18 different gestures that involved only the upper body. With these specifications, they reported up to 1 second anticipation. M. Kawashima et al. [3] and A. Shimada et al. [9] proposed early classification based on self organized maps (SOM) where each neuron learns one different posture of the possible gestures. In [9] the sparse code is extracted from the SOM and then the classification is done. In [3], while the incoming gesture is performed, initial parts from the gestures in the dictionary are chosen, with the intention of comparing similar duration gestures. The comparison of gestures is performed by Hausdorff distance, the gesture with the smallest distance is selected as the answer. Very recently, Ellis et al. proposed a method for early recognition that compares canonical poses (learnt from training data) to test gestures [1]. The authors report acceptable recognition rates, but it is difficult to assess the anticipation performance. In all of these works full body gestures are used, nevertheless, in none of these gestures more than two limbs are moved at the same time.

Differently from previous work, in this paper we recognize no only upper or lower body-movements but full-body movements. Also, our method is based on a DTW cumulative algorithm instead of SOM [3,9] or learned poses [1], thus no training phase is needed as in these alternative works. Furthermore, the proposed approach can work with only one example of each gesture to be recognized, no other early gesture-recognition approach can work under this setting.

2 One-Shot Early Recognition of Gestures with DTW

We want to classify full-body gestures made by one person regardless of his/her weight, height or speed of execution of the gestures, More importantly, we want to recognize a gesture before the user finishes its execution. This is a very complex problem because we have to classify the gesture with incomplete information and we do not know its duration beforehand. The problem is further complicated because of the similarity of gestures in the vocabulary, mainly at their beginning parts. Additionally we have to deal with noise incorporated by the considered sensor in the data acquisition process. Therefore, it is complicated to trigger a timely and correct response. We approach the problem with a DTW-based classifier and a novel criterion for early recognition. The proposed method comprises 3 main components: feature extraction, generating partial predictions, and triggering the final decision, which are described in detail in the rest of this section.

2.1 Data Representation

While the user is performing a gesture, a virtual skeleton is generated using Kinect [11] and OpenNI/NITE libraries[1]. The skeleton consists of 15 - 3D coordinates corresponding to body joints. The left plot in Figure 1 shows these points. Data is recorded at a speed of 30 frames per second (fps). For each gesture the user performs, we create a $15 \times 3 \times$ *number-of-frames* matrix to save the raw data, where *number-of-frames* varies depending on the gesture.

Raw-data collected with the Kinect sensor is represented using a simplified version of the method presented in [8]. This representation reduces dimensionality and makes the data invariant to rotation, translation, and scale. Instead of using principal component analysis as in [8] to get the torso frame, we propose the following: (1) obtain a normalized vector $\overrightarrow{r}$ from the segment $(\overrightarrow{N, T})$, where N and T are the neck and torso joints respectively. (2) obtain a normalized vector$\overrightarrow{u}$ from the segment $(\overrightarrow{N, RS})$, where RS is the right-shoulder joint, adjusting$\overrightarrow{r}$ in order to $\overrightarrow{u} \cdot \overrightarrow{r} = 1$ and still be a normalized vector, (3) calculate $\overrightarrow{t} = \overrightarrow{u} \times \overrightarrow{r}$. Then, we describe the **first-degree joints**, as in [8], (c.f. Figure 1, left) with two angles (θ and φ) which are calculated relative to the *torso frame* and the **second-degree joints** represented by two angles calculated relative to the limb to which they are connected. The result of the transformation is a 16-dim. vector per-frame instead of the initial 3×15 matrix.

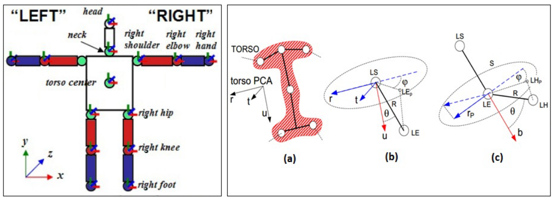

Fig. 1. Left: the 15 points of the skeleton we use: first-joints (in red), second-joints (in blue), green circles are used to calculate the *torso frame*. Right: (taken from [8]) shows the *torso frame*(a), and the angles representing first (b) and second (c) degree joints.

2.2 Early Classification

Let $\mathcal{D} = \{G_1, \ldots, G_R\}$ be the dictionary of gestures after the data transformation representation process and $G_r = \{f_r(1), \ldots, f_r(T_r)\}$ be one of the gestures in the vocabulary for $r \in \{1, \ldots, R\}$. Each G_r is composed of a sequence of T_r-frames, where each frame is represented by 16 angles as described in the previous section: $f_r(t_r) = \{\theta_1, \ldots, \theta_8, \varphi_1, \ldots, \varphi_8, \}$. One should note that we assume we have a single gesture of each particular class, that is, a one-shot learning scenario [2], thus we have R different classes of gestures.

[1] http://www.openni.org/

The *new gesture* we want to recognize is denoted by $G_T = \{f_T(1), \ldots, f_T(T_T)\}$, thus G_T has T_T frames. The classifier receives sequentially the frames of the *new gesture* at a 30fps rate. In order to avoid having to make predictions every time a frame is received, the classifier waits until w-frames are accumulated and then it makes a partial prediction by comparing *known* gestures with the *new* one. If this is not possible, the method waits again to receive another w-frames and it performs another comparison. This iterative process is repeated several times until either the gesture is recognized or the end of the *new gesture* is reached.

For comparing sequences, we estimate the distance between the partial information of the *new gesture* and the partial information of all the *known gestures* in $\mathcal{D}$. For this, we considered each of the 16 angles in a gesture up to time t_{it} as follows: $G_r(t_{it}) = \{A_{r,1}(t_{it}), \ldots, A_{r,16}(t_{it})\}$, where $A_{r,i}(t_{it}) = \{\theta_{r,i}(1), \ldots, \theta_{r,i}(t_{it}), \varphi_{r,i}(1), \ldots, \varphi_{r,i}(t_{it})\}$ for $1 \leq i \leq 16$, are the 16 time sequences of the gesture G_r and $\theta_{r,i}(t_{it})$, $\varphi_{r,i}(t_{it})$ are, respectively, the first and second degree angles of gesture G_r until time t_{it}. We used dynamic time warping (DTW) to compute the distance between sequences because it is one of the most used methods to compare sequences that may vary in time or speed. To avoid recalculating the similarity between the partial information of *known* and *new gestures* that was already calculated in previous iterations, we modified DTW to be accumulative (DTWacc): in each iteration DTWacc receives a new part of two time sequences to be compared, calculates the similarity between these parts and adds it to the results of the comparisons of previous iterations, this is shown in Figure 2. The comparison of two time sequences with DTWacc yields a distance value. To calculate the distance between the partial information of a *known* (G_r) and the *new gesture* (G_T) within DTW we proceed as follows:

$$D(G_r, G_T, t_{it}) = \sum_{i=1}^{16} dist(G_{r,i}, G_{T,i}, t_{it})$$

where $dist(G_{r,i}, G_{T,i}, t_{it}) = DTWacc\,(A_{r,i}(t_{it}), A_{T,i}(t_{it}))$, and $A_{r,i}(t_{it})$, $A_{T,i}(t_{it})$ are the sequences of angles of the i^{th}-joint up to time t_{it} for the known (G_r) and test gestures (G_T). We also incorporated a motion threshold γ_m to eliminate those limbs that the user hardly moves, and therefore are useless for recognition; thus, only those time sequences that move more than γm, are taken

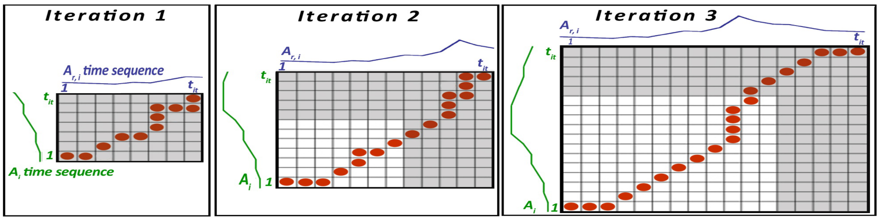

Fig. 2. Three iterations of the DTWacc method. In gray we show the part of each sequence that DTWacc compares; in white are shown the results of previous iterations; orange circles show how DTWacc aligns the two time sequences.

into account. For each distance $D(G_r, G_T, t_{it})$ we calculate the normalized score $S(G_r, G_T, T_{it}) = \frac{(D(G_r,G_T,t_{it}))^{-1}}{\sum_r (D(G_r,G_T,t_{it}))^{-1}}$. $S(G_r, G_T, T_{it})$ can be considered the probability that the gesture r is the one depicted in the test gesture G_T up to time t_{it}. The gesture with the highest probability will be chosen as a partial prediction for iteration it. We propose two ways to take a final decision on the identity of the gesture (i.e., triggering a flag indicating that a gesture has been recognized): **By separation** where one of the *known gestures* is noticeably more similar to the *new gesture*. **By forced classification** where the *new gesture* is about to end, according to an estimate on the duration of the gesture.

For **decision by separation** we consider two aspects: (1) the number of standard deviations n_σ that fit in the difference between the best gesture probability and the average of the next L best gesture probabilities, and (2) verify that a certain percentage of the estimated duration of the *new gesture* has been already executed. We defined the constant L to discard the $R - L + 1$ *known gestures* with the lower probabilities. With the remaining gestures we calculate the standard deviation dev and the average avg to calculate n_σ. If n_σ exceeds a certain threshold μ, then the classifier throws a final decision.

On **forced decision** the classifier provides an answer because it is estimated that more than $maxPer$ (a defined limit percentage very close to 100%) of the *new gesture* has been already performed and there was no decision by separation. We do not know how much the *new gesture* will last, so we need to do an estimate to prevent the new gesture of finishing without a prediction from the classifier or prevent hasty decisions. We consider that the total length of the *new gesture* is the minimum duration obtained from the two *known gestures* with the greatest probability on the most recent iteration, therefore, this duration is recalculated in each iteration.

3 Experimental Results

For our experiments we used two data sets. The first one is our Dance data set that consist of four dancing gestures: *up an down arm (A), pointing to the sky (B), moving arms and feet (C),* and *cow boy dance (D)* (see Figure 3, left), the gestures were performed by one person ten times each. This data set was captured with a Kinect at a 30fps rate and a resolution of 640x480. The second data set is MSR-Action3D [4], it comprises 20 gestures associated to interactive games (e.g., *side-boxing, tennis serve,* etc). Each gesture was performed by ten subjects for at most three times. The data were captured with Kinect at a 15fps and a resolution of 640x480. For this data set, the skeleton is represented with 20 points, but we only used the 15 available with the OpenNI skeleton. The MSR-Action3D data set has not been previously used for early gesture recognition, but we used here due to the lack of a benchmark for this task. Besides this is one of the most used data sets for action recognition using Kinect data. The parameters of our method: μ, γ_m, L and $maxPer$ were fixed empirically in preliminary experimentation.

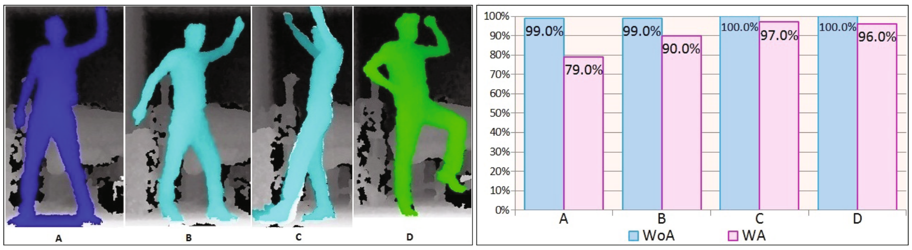

Fig. 3. Depth map generated with Kinect device of the four gestures of the data set Dance (left). Precision achieved per gesture WA and WoA for the Dance data set(right).

For the experiments with the dance dataset our approach obtained 99% recognition rate without anticipation (WoA, i.e., using 100% of the information for gestures) and 90% recognition rate with anticipation (WA, i.e., applying our early recognition technique), see Figure 3. On average, the proposed method was able to recognize a gesture using only ≈ 60% of the total duration of gestures and the response time for early recognition was below the 33.5ms.

For the MSR-Action3D data set, in a first experiment, we compared between randomly choosing one example of each gesture and using the half of the gestures to choose the best example of each category to form the training set and the rest of the examples for testing. The results are shown in Table 1 (a), where the column MSR-R shows the results with a random selection and the column MSR-S shows the results with the best selection of half of gestures. It can be seen that very similar results are obtained when using a randomly selected example for each category (MSR-R) and when the best example from the training set is obtained (MSR-S). This result evidences the robustness of our method to the selection of good training examples. For the random selection, we gained 2% of accuracy with anticipation and only needed 55% on average of the total duration of the *new gestures*. Although the accuracy is lower than that in the Dance data set, one must consider that the number of gestures in MSR-Action3D is 5 times larger than in the Dance data set and that gestures were performed by several subjects. The best recognition result for this collection is 88% [10], however, we emphasize that our method works under one-shot learning and it is intended to run with the gestures of a single subject, as in [2]. Another method based on DTW obtained 54% of accuracy in this collection [7], which is slightly better than our proposal, but that method is neither one-shot nor early recognition. Finally, the anticipation method in [1] achieved rates of up to 65.7% in the MSR-Action3D data set, but anticipation performance is not reported.

For the rest of the experiments we considered the MSR-Action3D data set and used the half of the gestures to choose the best example of each category to form the training set and the rest of the examples for testing.

In order to further evaluate the performance of our method when using gestures from a single subject we performed experiments dividing the examples of the MSR-Action3D data set by subjects c.f. Table 2. It can be seen that DTW

Table 1. Results of our method in the MSR-Action3D data set

MSR-R	MSR-S
45	50
47	48

(a)

	All subjects			Sep. subjects			Ref. [4]		
	AS1	AS2	AS3	AS1	AS2	AS3	AS1	AS2	AS3
WoA	42	47	52	97	93	96	72	71	79
WA	46	44	50	95	89	93	-	-	-

(b)

is very effective for recognizing gestures when a single subject is considered. Also, we can see that the proposed method is very effective at anticipating the recognition of gestures, as accuracy only decreases by 6.5%. Also, the average per-subject performance under WoA and WA (93.2% and 86.7%, respectively) is comparable with the best performance reported so far for the MSR-Action3D data set. As further comparison with other approaches, we divided the gestures by complexity, as reported in [4], where they form three groups: AS1, AS2 and AS3. AS1 and AS2 are intended to group gestures with similar movement (difficult to classify), while AS3 is intended to group very dissimilar actions together. Table 1 (b) shows the results of these sub-groups, considering all the subjects c.f. Table 1.b (columns 1-3); considering subjects and groups, c.f. Table 1.b (columns 4-6), and the results reported in [4], c.f. Table 1.b (columns 7-9).

Table 2. Results of the classification by subject

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**Avg.**
WoA	95.0	94.4	86.1	87.5	94.7	97.1	97.4	97.5	86.8	95.0	93.2
WA	85.0	69.4	80.6	87.5	94.7	88.2	100.0	90.0	81.6	90.0	86.7

From Table 2 (columns *All subjects*), we can see that the average performances (over groups) when considering all of the subjects are of 47% WA and 48% WoA; thus losing 1% in accuracy but using only the 47% of the duration of gestures. However, when we evaluate the performance over groups by separating users we obtained average performances (over groups) of 92.3% and 95.3% for WA and WoA, respectively (column Sep. subjects in Table 2); for these results only $\approx$ 50% of the gestures were needed for recognition. When compared with the 74% average accuracy obtained in [4], our method has higher precision using half of the information. Therefore, the proposed method is very effective for the classification of gestures when a single-user is considered, even when a single example is used for training the model.

The time required for the classification depends on the number of gestures. For 20 known gestures it takes 50.8ms WA and 450.5ms WoA on average to classify the new gesture. Using only 10 known gestures, it takes 38.9ms WA and 186.8ms WoA. Besides, our method can be parallelized so these response times can be improved.

4 Conclusions and Future Work

We proposed a DTW-based method for one-shot early-recognition of gestures. The proposed method is able to recognize gestures before the user finishes of executing it. The highest drop in accuracy when making early recognition was of 7.29% in terms of accuracy, but our savings in recognition response were between $40\%-50\%$. The features of DTW allowed us to design a method for one-shot learning, eliminating the training phase and thereof, the number of labeled gestures needed to generate a model. Also, the method proved to be robust to the selection of examples for the dictionary, and we show that it produces better results when a single-subject performs the gestures. The response time of our method depends directly on the number of *known gestures* used, in our experiments the time needed for classification WoA is up to 8 times larger in average than that required to make classification WA. For future work, we want to include a segmentation method to the classifier to detect the beginning and the end of the gestures in order to achieve online classification. Also we want to parallelize our method to reduce even more the response time. Finally, we want to automate the learning of the parameters μ, γ_m, L and $maxPer$ to avoid setting them empirically.

References

1. Ellis, C., Masood, S.Z., Tappen, M.F., LaViola Jr, J., Sukthankar, R.: Exploring the trade-off between accuracy and observational latency in action recognition. International Journal of Computer Vision 101(3), 420–436 (2013)
2. Guyon, I., Athitsos, V., Jangyodsuk, P., Escalante, H.J., Hamner, B.: Results and analysis of the chaLearn gesture challenge 2012. In: Jiang, X., Bellon, O.R.P., Goldgof, D., Oishi, T. (eds.) WDIA 2012. LNCS, vol. 7854, pp. 186–204. Springer, Heidelberg (2013)
3. Kawashima, M., Shimada, A., Nagahara, H., Taniguchi, R.I.: Adaptive template method for early recognition of gestures. In: 17th WFCV, pp. 1–6. IEEE (2011)
4. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: CVPRW, pp. 9–14. IEEE (2010)
5. Mitra, S.: Gesture recognition: A survey. Trans. on Syst. Man and Cyb. - C 37, 311–324 (2007)
6. Mori, A., Uchida, S., Kurazume, R., Taniguchi, R.-I., Hasegawa, T., Sakoe, H.: Early recognition and prediction of gestures. In: ICPR, pp. 560–563 (2006)
7. Muller, M., Roder, T.: Motion templates for automatic classification and retrieval of motion capture data. In: Proc. SIGGRAPH-SAC, pp. 137–146 (2006)
8. Raptis, M., Kirovski, D., Hoppe, H.: Real-time classification of dance gestures from skeleton animation. In: SoCA, pp. 147–156. ACM (2011)
9. Shimada, A., Kawashima, M., Taniguchi, R.-I.: Early recognition based on co-occurrence of gesture patterns. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part II. LNCS, vol. 6444, pp. 431–438. Springer, Heidelberg (2010)
10. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR, pp. 1290–1297. IEEE (2012)
11. Zhengyou, Z.: Microsoft kinect and its effect. IEEE MultiMedia 19, 4–10 (2012)