Real-time Body Gestures Recognition using Training Set Constrained Reduction

Fabrizio Milazzo, Vito Gentile, Antonio Gentile and Salvatore Sorce

Ubiquitous Systems and Interfaces Group (USI)
Università degli Studi di Palermo - Dipartimento dell'Innovazione Industriale e Digitale (DIID)
Viale delle Scienze, Edificio 6, 90128 Palermo, Italy
{firstname.lastname}@unipa.it
http://usi.unipa.it

Abstract. Gesture recognition is an emerging cross-discipline research field, which aims at interpreting human gestures and associating them to a well-defined meaning. It has been used as a mean for supporting human to machine interaction in several applications of robotics, artificial intelligence, and machine learning. In this paper, we propose a system able to recognize human body gestures which implements a constrained training set reduction technique. This allows the system for a real-time execution. The system has been tested on a publicly available dataset of 7,000 gestures, and experimental results have highlighted that at the cost of a little decrease in the maximum achievable recognition accuracy, the required time for recognition can be dramatically reduced.

Keywords: Gesture Recognition, Real-time systems, Constrained optimization

1 Introduction

In the last decade, *Gesture recognition*, a new field of artificial intelligence has grown more and more. It aims to interpret human movements and to associate them to a specific meaning. Here, the term "movement" refers to the motion of either the whole or parts of human body [1].

Gesture recognition was born with the aim of improving human-machine interactions, by making it as simple and natural as possible. Indeed, there are many applications that may take advantage from gesture recognition, e.g.: *health monitoring* [2], *lie detection* [3], *automatic movie subtitling* [4], *online games* [5], *e-tutoring systems* [6], *emotion recognition* [7], *management systems for ambient intelligence* [8], [9] and so on.

Among the others, there are two typical issues that must be addressed in every gesture recognition application: *ensuring real-time processing* and *maximizing recognition accuracy*.

Real-time processing allows recognition of gestures in a negligible time interval; on the other hand, the recognition accuracy represents the probability that the gesture recognition algorithm will properly recognize a gesture.

adfa, p. 1, 2011.

The main contribution of this work is a novel system for body gesture recognition, which implements a technique based on training set constrained reduction. The key idea is to reduce as much as possible the size of the training set used for recognition, by taking into account the two aforementioned issues.

The proposed system benefits of the *Dynamic Time Warping* [10] recognition technique, which makes the system independent of gestures length and size.

The rest of the paper is organized as follows: Section 2 deepens the discussion and state of the art in gesture recognition and some state of the art solutions; Section 3 describes the proposed system for gestures recognition; Section 4 highlights the experimental results obtained by using the system with an online available dataset of over 7000 gestures; finally, Section 5 describes the conclusions and some possible improvements of our proposal.

2 Related Works

In the last twenty years, gesture recognition has been the subject of several researches in the field of pattern recognition and has found many applications in robotics [11] and human-computer interaction [12]. Moreover, the availability of novel technologies has significantly contributed to the growing interest towards the development of gesture recognition algorithms.

While earlier works used RGB cameras as data source [13], the more recent Kinect-like devices (i.e. low-cost devices providing an integrated channel for RGB and depth data [14]) allow for more precise information about the observed gestures.

Indeed, Shotton et al. developed a robust algorithm for human pose estimation from single depth images [15], and thanks to their intuition, nowadays there exist many software libraries able to extract skeletal joints[1] from depth images of humans.

Using the aforementioned joints as basic features, it is possible to extract dynamic and static body gestures. According to Henze et al. [16], gestures are said to be static if they can be described by their position and spatial arrangement only; this class of gestures is also known as postures or poses [17], and they only need a single time frame to be entirely observed. In this work, instead, we will focus on the so-called dynamic gestures, i.e. a sequence of changing postures along a variable time interval.

Many authors have described methods for recognizing gestures by modeling them as temporal sequences of skeletal joints. In this context, two of the most suited and adopted mathematical tools are the Hidden Markov Models (HMMs) and the Dynamic Time Warping (DTW).

For instance, in [18] authors use an algorithm based on a Gaussian Mixture Hidden Markov Model, while in [19] Carmona and Climent have compared the performance of these two tools, showing that DTW is more suited for gesture recognition. Both HMM and DTW need a training stage devoted to learning a mathematical model used in a later stage to recognize new unseen sequences.

[1] A joint is defined as the point of conjunction between two adjacent bones of the human skeleton.

Despite the mathematical tool used for recognition, the more complex the learned model is, the more the computation needed for recognizing the sequences will be. To this aim, many algorithms have been recently developed for reducing the complexity of the learned models. In particular, they belong to the so-called class of *training set reduction* algorithms.

As regards HMM-based solutions, the problem is usually faced up by using dimensionality reduction algorithms as the Principal Component Analysis (PCA) [20]. On the other hand, in DTW-based solutions, the reduction algorithms aims at reducing as much as possible the cardinality of the training set to very few and representative samples named prototypes (see for instance [21] and [22]).

With the aim of providing a real-time system for gesture recognition, in this work we propose a system making use of DTW as a mathematical tool for comparing temporal joint sequences of variable length. This choice is in line with findings described in [19], i.e. DTW requires a lower number of training samples to achieve the same performance of HMM. Moreover, we developed a *training set constrained reduction* technique, which at the same time reduces the size of the training set and constrains the accuracy of the recognizer to be over a certain threshold.

3 System Description

The purpose of this Section is to describe our proposed system for body gestures recognition. We implemented a *real-time* system, with the aim of keeping as low as possible the computational burden of the recognition task, while maximizing its recognition accuracy.

To this end, we shifted the most of the computation in a learning method aimed at reducing the cardinality of the available training set and, as a consequence, the time complexity of the recognition.

First of all, we assume the availability of a training set named *LG*, made up of pairs in the form of <*Label*, *Gesture*>.

The "label" component is a text representing the name of the gesture; as an example, a movement of the arm at eye's height from right to left may be labeled as "Swipe Right To Left".

As regards the definition of "gesture", we choose to use the *joint* representation of the human skeleton, so we define a gesture *G* of length *T* as the sequence of the *N* joints coordinates over the time:

$$\mathrm{G} = \begin{pmatrix} x_{1,1}y_{1,1}z_{1,1} \cdots \mathrm{x}_{1,\mathrm{N}}y_{1,\mathrm{N}}z_{1,\mathrm{N}} \\ x_{2,1}\ y_{2,2}z_{2,2} \cdots x_{2,\mathrm{N}}\ y_{2,\mathrm{N}}z_{2,\mathrm{N}} \\ \vdots \\ x_{T,1}y_{T,1}Z_{T,1} \cdots x_{T,N}y_{T,N}Z_{T,N} \end{pmatrix} \quad (1)$$

In order to maintain the approach as generic as possible, we will make no assumptions about neither the duration nor the volume occupied by the training gestures. The proposed system is thus implemented by two modules:

1. **gesture recognition***:* a new incoming gesture is matched to the most similar one in the reduced dataset, and the associated label is provided as output;
2. **training set reduction**: in order to provide real-time performance, here the input *LG* dataset is filtered in order to retain only the most representative gestures, named here as "prototypes", to be used for the recognition task.

3.1 Gesture recognition module

The role of this module is to accept a new body gesture as a sequence of skeletal joints coordinates and to output a label representing the recognized gesture name.

With the aim of providing real-time performance, we implemented this module as a one-nearest neighbor classifier, which compares the incoming gesture to those in the training set, and returns the label of the nearest one. Mathematically speaking, this is carried out as follows:

$$G^* = \underset{G}{\operatorname{argmin}} Dist(G^{new}, G), \forall\ G \in LG \tag{2}$$

where G^{new} is the gesture to be recognized, $Dist(\cdot,\cdot)$ is a distance metric, and G^* is the nearest gesture in LG. As a consequence, the recognized label L^* will be the label component of the $<L^*,G^*>$ pair contained in LG.

As regards the distance metric, we chose to use the *Dynamic Time Warping* one, which is able to compare gestures of different time length and spatial volume, by using simple insertion and deletion operations.

For reader's commodity, in the following algorithm we report the steps needed to compute DTW between two gestures G^{new} and G:

Algorithm DTW(G^{new}, G)
Input: G^{new} as a $T_1 \times N \times 3$ *matrix* *#gesture to be recognized*
Input: G as a $T_2 \times N \times 3$ *matrix* *#gesture in the train set*
Output: *x* as a *scalar* *#distance between gestures*

```
1.Declare DTW as a (T1 + 1) × (T2 + 1) matrix
2.for i=1 to T1 do
2.1. DTW[0,i]=infinity
3.for i=1 to T2 do
3.1. DTW[i,0]=infinity
4.DTW[0,0]=0
5.for i=1 to T1 do
5.1.for j=1 to T2 do
5.1.1. d=L2norm(G^new[i],G[j])        #distance between frames
5.1.2. DTW[i,j]=d+min{DTW[i-1,j],DTW[i,j-1],DTW[i-1,j-1]}
6.Return x=DTW[T1,T2]
```

Clearly, the time required by the nearest neighbor classifier is linear with respect to the number of gestures composing the *LG* set. In order to allow for real-time recognition, it is important to keep the cardinality of such set as low as possible. This issue is thus be solved by the training set reduction module.

3.2 Training set reduction module

The main purpose of this module is to reduce the size of the training set used by the recognizer. For this reason, it must be run *before* new gestures are recognized. In particular, it reduces the cardinality of the *LG* training set, as it has a direct influence onto the time complexity of the recognition module.

The idea is to extract only the relevant pairs <*Label*, *Gesture*>, which can be seen as a sort of "prototypes" for the training set, and then use such prototypes instead of the whole training set to perform recognition.

The module induces a partition of the original training set *LG* by splitting it into two subsets, namely *P (*which contains the prototype gestures) and *NP* (containing non-prototype gestures), so that $P \cup NP = LG$ and $P \cap NP = \emptyset$.

In order to evaluate how good an induced partition is, we can use the procedure described in Section 3.1. In particular, we can recognize the gestures contained in *NP* using *P* as the training set (instead of the whole *LG*).

Moreover, we define the evaluation function $M(P,NP) \rightarrow [0...1]$ as the accuracy of the recognition for the induced partition of *LG*.

Since the purpose of this module is to lower as much as possible the number of prototypes in *P* while keeping as high as possible the value of *M(P,NP)*, we apply a gradient descent to the following constrained optimization problem:

$$\begin{aligned} &\max \quad M(P,NP) \\ &\min \quad\quad |P| \\ &\theta \leq M(P,NP) \end{aligned} \tag{3}$$

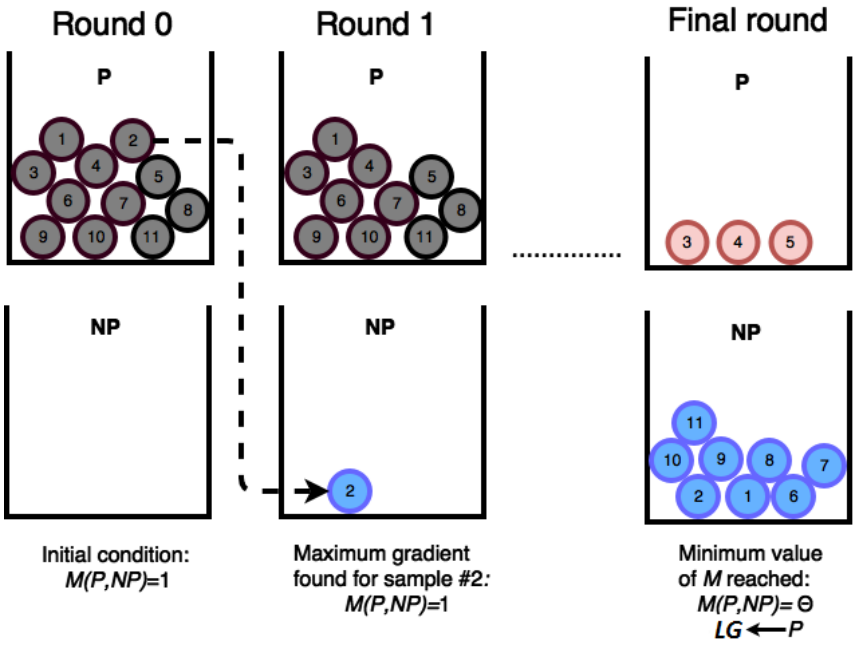
where θ is a lower bound on the accuracy of the recognition in the training stage.

The initial condition is $P=LG$ $NP=\Phi$, $M(P,NP)=1$. Then, the module starts a loop composed of a variable number of rounds, iterated until the constraints are satisfied at equality. During each round, all the samples in *P* are removed (one at a time), put in *NP*, and labeled with the gradient of *M*, computed as follows:

$$\nabla M = M(P^-,NP^+) - M(P,NP) \tag{4}$$

where P^- and NP^+ indicate the sets obtained by moving one sample gesture from *P* into *NP*. At the end of each round, gestures in *P* are sorted according to their gradients, and the one with the maximum value is definitively put in the *NP* set. The loop is iterated until *M(P,NP)* remains above the θ threshold. In the end, the prototypes in the resulting dataset *P*, derived from *LG*, will be used for the recognition task. **Fig. 1** clarifies, with a visual example, the training set reduction flow.

Fig. 1. Example flow of the training set reduction



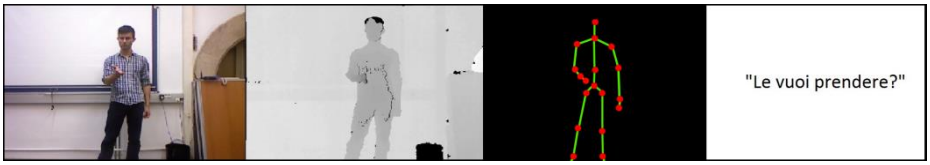
4 Experimental Assessment

The recognition algorithm have been tested in a real deployment, by using the "Chalearn multimodal gesture recognition dataset" [23].

The dataset is made up of over 7000 samples containing each one: a RGB-D video, the gesture joints sequence and a textual label representing the name of the gesture.

The RGB-D videos were acquired by a Microsoft Kinect device, at the rate of 30 FPS, the skeleton data are described by 20 joints per frame, while the textual labels were manually added and represent 20 Italian cultural/anthropological signs, performed by 27 different users. **Fig. 2** depicts one sample data taken from the dataset.

Fig. 2. RGB, depth, skeletal and textual data of one sample from the dataset.



First of all, we built the *LG* dataset by extracting only the <*Label*, *Gesture*> pairs from the samples contained in the Chalearn dataset.

We then implemented the modules described in Section 3.1 and 3.2 using the Python programming language, and deployed in a Raspberry Pi 3 device (4-core CPU at 1.2 GHz, running a 32-bit Raspbian distribution).

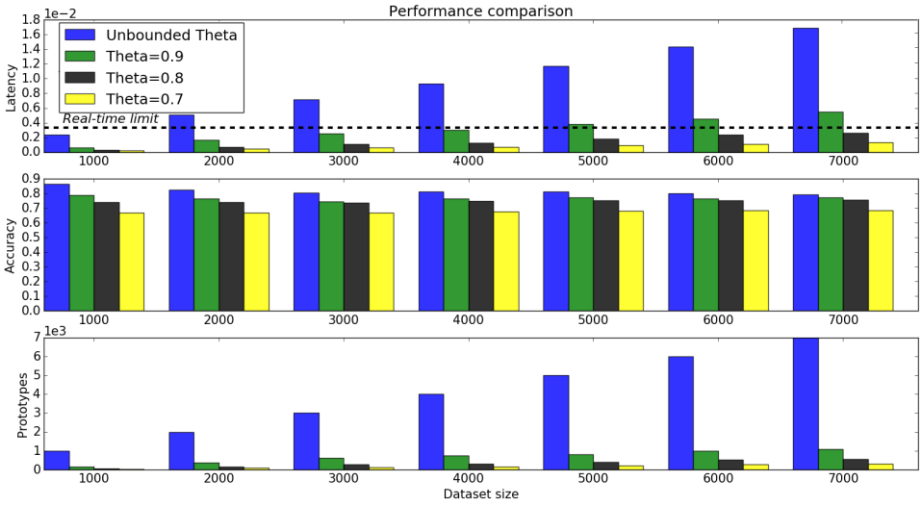
The raw dataset was sub-sampled by randomly choosing from 1000 to 7000 samples (with steps of 1000). Then, the resulting datasets have been divided into train and test by using the leave-1-out technique [24].

The baseline for our comparison is the recognition applied without using training set reduction. The other versions make use of the training set reduction module for three different values of the training accuracy thresholds $\theta \in \{0.7, 0.8, 0.9\}$.

Fig. 3 depicts the results obtained for: i) the latency required for recognizing one gesture, ii) the accuracy of the recognition, and iii) the number of prototypes retained from the original *LG* dataset.

The first row reports the latency required for recognizing one new incoming gesture. We set the maximum limit for real-time computation to 3.33 ms (i.e. the maximum available time for recognizing a gesture in a continuous stream of data at 30 FPS).

Fig. 3. Performance of the recognition module for different thresholds



Unsurprisingly, the recognition module performs very fast for all the cases where training set reduction was applied, while the baseline is very far from real-time performance. We note also that when the training set size goes over 5000 samples, the case for $\theta = 0.9$ is not real-time compliant.

The second row reports the accuracy of the recognition. The baseline case achieves very good performance with an average recognition accuracy of 0.81, and this is due to the use of the all available samples in the dataset. Anyway, in all the remaining cases, accuracy is only a little bit lower than baseline, ranging between 0.65 and 0.8.

The third row reports the number of retained prototypes, given a certain training accuracy threshold. Interestingly, the number of prototypes after training set reduction is a very small percentage of the whole dataset. It is also worth noting that such number increases very slowly with respect to the dimensions of the dataset, and this positively affects the timing performance of the recognition module.

Such results highlight that running training set reduction is fundamental because at the cost of a little decrease in the maximum achievable accuracy, then the recognition module becomes 20x, 10x and 5x times faster than the baseline case for accuracy thresholds of 0.7, 0.8 and 0.9 respectively. Moreover, setting a threshold $\theta = 0.8$ allows the system to achieve the best trade-off between recognition time (always below the real-time limit) and recognition accuracy (slightly less than the baseline case).

5 Conclusions

In this paper, we presented a novel approach to recognize body gestures in real-time by applying a training set constrained reduction technique.

Starting from a dataset of *<Label, Gesture>* pairs, the training set reduction module selects the most representative gestures (prototypes) that will be used by the recognition module, implemented as a nearest-neighbor classifier based on Dynamic Time Warping.

We evaluated the performance of the recognition module by running it on a Raspberry Pi 3, using training sets of size ranging from 1000 to 7000 gestures. Moreover, the results highlighted the importance of the training set reduction module, which allows for real-time execution at the cost of a little decrease in the maximum achievable accuracy.

In a future work, we are planning to check performance (time complexity and accuracy) for more sophisticated classifiers, as well as testing the recognizer in-the-wild (i.e. including it in an actual deployment, and testing its performance against end users).

References

1. S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311-324, 2007.
2. T. Starner, J. Auxier, D. Ashbrook and M. Gandy, "The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring," in *The fourth international symposium on Wearable computers*, 2000.
3. C. Davatzikos, K. Ruparel, Y. Fan, D. Shen, M. Acharyya, J. Loughead, R. Gur and D. D. Langleben, "Classifying spatial patterns of brain activity with machine learning methods: application to lie detection," *Neuroimage*, vol. 28, no. 3.
4. S.-B. Park, E. Yoo, H. Kim and G.-S. Jo, "Automatic emotion annotation of movie dialogue using WordNet," 2011.
5. H. Kang, C. W. Lee and K. Jung, "Recognition-based gesture spotting in video games," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1701-1714, 2004.
6. R. W. Picard and R. Picard, Affective computing, vol. 252, MIT press Cambridge, 1997.

7. V. Gentile, F. Milazzo, S. Sorce, A. Gentile, A. Augello and G. Pilato, "Body Gestures and Spoken Sentences: a Novel Approach for Revealing User's Emotions," in *11th International Conference on Semantic Computing (ICSC 2017)*, 2017.
8. A. a. R. G. L. a. M. F. a. O. M. De Paola, "Adaptable data models for scalable ambient intelligence scenarios," International Conference on Information Networking (ICOIN), 2011.
9. E. Daidone and F. Milazzo, "Short-Term Sensory Data Prediction in Ambient Intelligence Scenarios," in *Advances onto the Internet of Things*, Springer, 2014, pp. 89-103.
10. D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, vol. 10, 1994, pp. 359-370.
11. A. K. Malima, E. Özgür and M. Çetin, "A fast algorithm for vision-based hand gesture recognition for robot control," in *IEEE 14th Signal Processing and Communications Applications*, Antalya, Turkey, 2006.
12. V. Gentile, A. Malizia, S. Sorce and A. Gentile, "Designing Touchless Gestural Interactions for Public Displays In-the-Wild," in *Human-Computer Interaction: Interaction Technologies*, M. Kurosu, Ed., Springer International Publishing, 2015, pp. 24-34.
13. Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review," *Gesture-Based Communication in Human-Computer Interaction*, vol. 1739, pp. 103-115, 2001.
14. V. Gentile, S. Sorce and A. Gentile, "Continuous Hand Openness Detection Using a Kinect-Like Device," in *Eighth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, Birmingham, UK, 2014.
15. J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook and R. Moore, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition - CVPR '11*, 2011.
16. N. Henze, A. Löcken, S. Boll, T. Hesselmann and M. Pielot, "Free-hand gestures for music playback: deriving gestures with a user-centred process," in *9th International Conference on Mobile and Ubiquitous Multimedia*, 2010.
17. S. Sorce, V. Gentile and A. Gentile, "Real-Time Hand Pose Recognition Based on a Neural Network Using Microsoft Kinect," in *Eighth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, 2013.
18. Y. Song, Y. Gu, P. Wang, Y. Liu and A. Li, "A Kinect based gesture recognition algorithm using GMM and HMM," in *6th International Conference on Biomedical Engineering and Informatics*, 2013.
19. J. M. Carmona and J. Climent, "A Performance Evaluation of HMM and DTW for Gesture Recognition," in *17th Iberoamerican Congress (CIARP 2012)*, Buenos Aires, Argentina, 2012.
20. H. P. Shum, E. S. Ho, Y. Jiang and S. Takagi, "Real-time posture reconstruction for Microsoft Kinect," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1357--1369, 2013.
21. C. Kasemtaweechok and W. Suwannik, "Training set reduction using Geometric Median," in *15th International Symposium on Communications and Information Technologies (ISCIT)*, 2015.
22. J. Sánchez, "High training set size reduction by space partitioning and prototype abstraction," *Pattern Recognition*, vol. 37, no. 7, p. 1561–1564, 2004.
23. S. Escalera, J. Gonzàlez, X. Barò, M. Reyes, O. Lopes, I. Guyon, V. Athitsos and H. Escalante, "Multi-modal gesture recognition challenge 2013: Dataset and results," in *Proceedings of the 15th ACM on International conference on multimodal interaction*, 2013.
24. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Ijcai*, vol. 14, no. 2, pp. 1137-1145, 1995.