Combination of HMM and DTW for 3D Dynamic Gesture Recognition Using Depth Only

Hajar Hiyadi, Fakhreddine Ababsa, Christophe Montagne, El Houssine Bouyakhf and Fakhita Regragui

Abstract Gesture recognition is one of the important tasks for human Robot Interaction (HRI). This paper describes a novel system intended to recognize 3D dynamic gestures based on depth information provided by Kinect sensor. The proposed system utilizes tracking for the upper body part and combines the hidden Markov models (HMM) and dynamic time warping (DTW) to avoid gestures misclassification. By using the skeleton algorithm provided by the Kinect SDK, body is tracked and joints information are extracted. Each gesture is characterized by one of the angles which remains active when executing it. The variations of the angles throughout the gesture are used as inputs of Hidden Markov Models (HMM) in order to recognize the dynamic gestures. By feeding the output of (HMM) back to (DTW), we achieved good classification performances without any misallocation. Besides that, using depth information only makes our method robust against environmental conditions such as illumination changes and scene complexity.

Keywords 3D gesture recognition · Gesture tracking · Depth image · Hidden Markov models · Dynamic time warping

1 Introduction

1.1 Motivation

The goal of Human Robot Interaction (HRI) research is to increase the performance of human robot interaction in order to make it similar to human-human interaction, allowing robots to assist people in natural human environments. As for communication between humans, gestural communication is also widely used in human robot

H. Hiyadi (✉) · F. Ababsa · C. Montagne
Laboratoire IBISC, Université d'Évry Val d'Essonne, Évry, France
e-mail: hajar.Hiyadi@ufrst.univ-evry.fr; hiyadi.h89@gmail.com

H. Hiyadi · E.H. Bouyakhf · F. Regragui
Laboratoire LIMIARF, Université Mohammed V Agdal, Rabat, Morocco

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1_13

interaction. Several approaches have been developed over the last few years. Some approaches are based on data markers or gloves and use mechanical or optical sensors attached to these devices that transform reflexion of the members into electrical signals to determine the posture. These methods are based on various information such as the angles and the joints of the hand which contain data position and orientation. However, these approaches require that the user wear a glove or a boring device with a load of cables connected to the computer, which slows the natural human robot interaction. In the other side, computer vision is a non intrusive technology which allows gesture recognition, without any interference between the human and the robot. The vision-based sensors include 2D and 3D sensors. However, gesture recognition based on 2D images had some limitations. Firstly, the images can not be in a consistent level lighting. Second, the background elements can make the recognition task more difficult. With the emergence of Kinect [1], depth capturing in real time becomes very easy and allows us to obtain not only the location information, but also the orientation one. In this paper we aim to use only the depth information to build a 3D gesture recognition system for human robot interaction.

1.2 Related Work

A gesture recognition system includes several steps: detection of one or more members of the human body, tracking, gesture extraction and finally classification. Hand tracking can be done based on skin color. This can be accomplished by using color classification into a color space. In [2], skin color is used to extract the hand and then track the center of the corresponding region. The extracted surface into each chrominance space has an elliptical shape. Thus, taking into account this fact, the authors proposed a skin color model called elliptical contour. This work was extended in [3] to detect and localize the head and hands. In addition, the segmentation process is also an important step in tracking. It consists of removing non-relevant objects leaving behind only the regions of interest. Segmentation methods based on clustering are widely used in hand detection and especially K-means and expectation maximization. In [4] the authors combine the advantages of both approaches and propose a new robust technique named KEM (K-means Expectation Maximization). Other detection methods based on 2D/3D template matching were also developed [5–7]. However, skin color based approaches are greatly affected by illumination changes and background scene complexity. Therefore, recent studies tend to integrate new information such as depth. Indeed, depth information given by depth sensors can improve the performance of gesture recognition systems. There are several studies that combine color and depth information, either in tracking or segmentation [8–11]. Other works combine depth information, color and speech [12]. In [10], the authors use a silhouette shape based technique to segment the human body, then they combine 3D coordinates and motion to track the human in the scene. Filtering approaches are also used in tracking such as the Unscented Kalman Filter [13], the Extented Kalman Filter [14] and the Particle Filter [15]. Other methods are based

on points of interest which have more constraints on the intensity function and are more reliable than the contour based approaches [16]. They are robust to occlusions present in a large majority of images.

The most challenging problem in dynamic gesture recognition is the spatial-temporal variability, when the same gesture could be different in velocity, shape and duration. These characteristics make recognition of dynamic hand gestures very difficult compared to static gestures [17]. As in speech, hand writing and character recognition [18, 19], HMM were successfully used in gesture recognition [20–22]. Actually, HMM can model spatial-temporal time series and preserve the spatial-temporal identity of gesture. The authors in [23] developed a dynamic gesture recognition system based on the roll, yaw and pitch orientations of the left arm joints. Other mathematical models such as Input-Output Hidden Markov Model (IOHMM) [24], Hidden Conditional Random Fields (HCRF) [25] and Dynamic Time Warping [26] are also used to model and recognize sequences of gestures.

In this paper, we propose a robust classification realized by combining HMM and DTW methods for 3D dynamic gesture recognition. The basic framework of the technique is shown in Fig. 1. The Skeleton algorithm given by the Kinect SDK is used for body tracking. Only depth information is recorded. The 3D joints information are extracted and used to calculate new and more relevant features which are the angles between joints. Discrete HMM with Left-Right Banded topology are used to model and classify gestures. Finally, the output of HMM is given as input for DTW algorithm in order to measure the distance between the gesture sequence and a reference sequence. The final decision is given by comparing the distance calculated

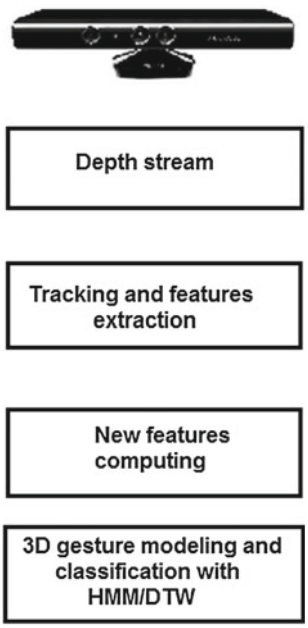

Fig. 1 Flowchart of the proposed 3D dynamic gesture recognition technique

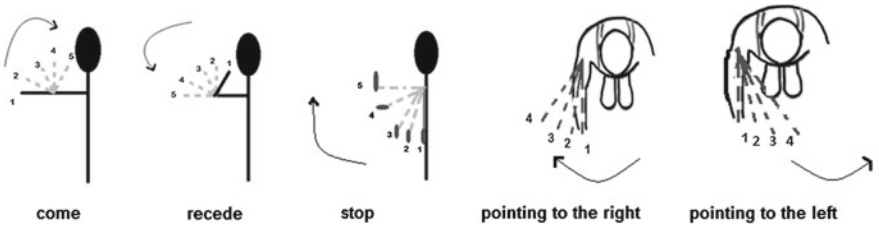

Fig. 2 Five distinct gesture kind

by DTW to a fixed threshold. The evaluation experiments show the effectiveness of the proposed technique. The performance of our technique is further demonstrated with the validation step which yielded good recognition even without training phase.

The rest of the paper is organized as follows: Sect. 2 describes our 3D dynamic gesture approach and the features we used. Section 3 gives some experimental results. Finally, Sect. 4 ends the paper with a conclusion and future work.

2 Proposed Approach

In the context of human robot interaction, the aim of our work is to recognize five 3D dynamic gestures based on depth information. We are interested in deictic gestures. The five gestures we want to recognize are: {*come*, *recede*, *stop*, *pointing to the right* and *pointing to the left*}. Figure 2 shows the execution of each gesture to be recognized. Our gesture recognition approach consists of two main parts: (1) Human tracking and data extraction, and (2) gesture classification.

2.1 Human Tracking and Data Extraction

In order to proceed to the gesture recognition, we need first to achieve a robust tracking for Human body and arms. Most recent tracking methods use color information. However, color is not a stable cue, and is generally influenced by several factors such as brightness changing and occlusions. Hence, color-based tracking approaches fail often and don't success to provide 3D human postures at several times. In our work we choose to use a depth sensor (Kinect) in order to extract 3d reliable data. Figure 3 shows the reference coordinate frames associated to the acquisition system.

The coordinates x, y and z denote, respectively, the x and y positions and the depth value. Human tracking is performed using the Skeletal Tracking method given by the kinect SDK.[1] This method projects a skeleton on the human body image so each joint

[1] http://msdn.microsoft.com/en-us/library/jj131025.aspx.

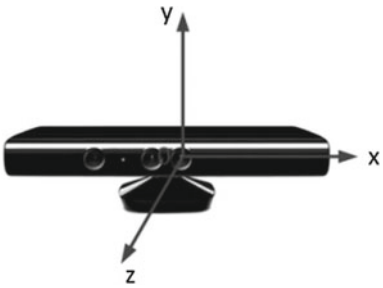

Fig. 3 Kinect system coordinate

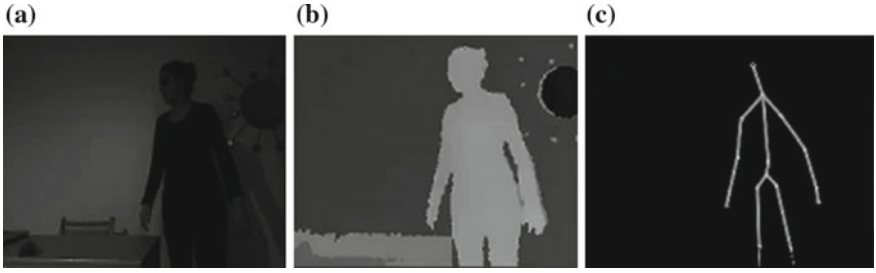

Fig. 4 **a** RGB image, **b** depth image, **c** skeleton tracking

of the body is related to a joint of the projected skeleton. In this manner, it creates a collection of 20 joints to each detected person. Figure 4 shows the information used in our approach: depth image (b) and skeleton tracking (c).

The idea is to estimate in real time the variations of the active angles while executing the gestures. The considered angles are: α elbow, β shoulder and γ armpit angle, as shown in Fig. 5. Each angle is then computed from the $3D$ coordinates of the three joints that are commonly accounted to it:

- α elbow angle is computed from the $3D$ coordinates of elbow, wrist and shoulder joints.
- β shoulder angle is computed from the $3D$ coordinates of shoulder, elbow and shoulder center joints.
- γ armpit angle is computed from the $3D$ coordinates of shoulder, elbow and hip joints.

When performing a gesture we record the values given by each of these three angles and we store the results in vectors as follow:

$$V_\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_T] \tag{1}$$

$$V_\beta = [\beta_1, \beta_2, \ldots, \beta_T] \tag{2}$$

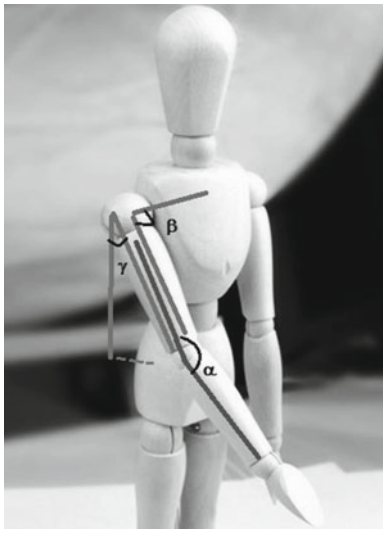

Fig. 5 α, β and γ angles

$$V_\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_T] \tag{3}$$

where T is the length of the gesture sequence, it is variable from a gesture to another and from a person to another. The input vector of our $3D$ dynamic gesture recognition system will be then written as:

$$V_\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_T, \beta_1, \beta_2, \ldots, \beta_T, \gamma_1, \gamma_2, \ldots, \gamma_T] \tag{4}$$

The gesture description based on angles variation allows distinguishing between different human gestures. Thus, for every canonical gesture, there is one main angle which changes throughout the gesture and the remaining two angles vary slightly. We consider the five gestures defined previously. The angle which is varing for *come* and *recede* is the angle α. Likewise, the angle γ for *stop* gesture, and angle β for both pointing gestures. The main angle's variations in each gesture are showing in the Table 1.

Table 1 The main angle's variations in each gesture

	α	β	γ
Come	180°–30°	–	–
Recede	30°–180°	–	–
Pointing to right	–	90°–150°	–
Pointing to left	–	90°–40°	–
Stop	–	–	30°–80°

In this work, we propose to use the sequences of angles variations as an input of our gesture recognition system as explained in the next section.

2.2 Gesture Classification Method

Our recognition method is based on a combination of Hidden Markov Models (HMM) and Dynamic Time Warping (DTW) method. HMM are widely used in temporal pattern, speech, and handwriting recognition, they generally yield good results. The problem in the dynamic gestures is their spatial and temporal variability which make their recognition very difficult, compared to the static gestures. In fact, the same gesture can vary in speed, shape, length. However, HMM have the ability to maintain the identity of spatio-temporal gesture even if its speed and/or duration change. Since we work with time series data, we use Dynamic Time Warping algorithm to measure similarity between two sequences that may vary in time and speed. DTW warps the sequences and gives a distance like quantity between them.

In the first stage, we classify the gesture using HMM [27]. Based on the best probability of belonging to one of the five classes, the gesture kind is recognized. In the second stage, we measure the similarity between the variations of the main angle sequence that characterizes the gesture class which HMM gave as output and another variations sequence of the same angle taken as a reference using DTW. Next, the distance is compared to a precalculated threshold. If the distance is less than the threshold we keep the result provided by HMM method else the gesture will be considered as an unknown gesture then rejected. Therefore a bad performed gesture will be rejected instead of being misclassified. Figure 6 shows the steps of our recognition system. First, HMM method will classify a given gesture (Gtest) into one of the five classes. Then, HMM method will give the result which is the type of gesture (for example: *Come*). As mentioned before, the angle which characterizes the gesture *Come* is elbow angle designed by α. Thus, we take the first part of the gesture sequence (Gtest) which corresponds to α angle variations, and we take a reference sequence of α angle variations in *Come* gesture from the database. Next, we calculate the distance between these two sequences using DTW method. The resulting distance is compared to a threshold that was fixed for the gesture *Come*.

Hidden Markov Models. An HMM can be expressed as $\lambda = (A, B, \pi)$ and described by:

(a) A set of N states $S = \{s_1, s_2, \ldots, s_n\}$.
(b) An initial probability distribution for each state $\Pi = \{\pi_j\}$, $j = \{1, 2, \ldots, N\}$, with $\pi_j = \text{Prob}(S_j \text{ at } t = 1)$.
(c) A N-by-N transition matrix $A = \{a_{ij}\}$, where a_{ij} is the transition probability of s_i to s_j; $1 \leq i, j \leq N$ and the sum of the entries in each row of the matrix A must be equal to 1 because it corresponds to the sum of the probabilities of making a transition from a given state to each of the other states.

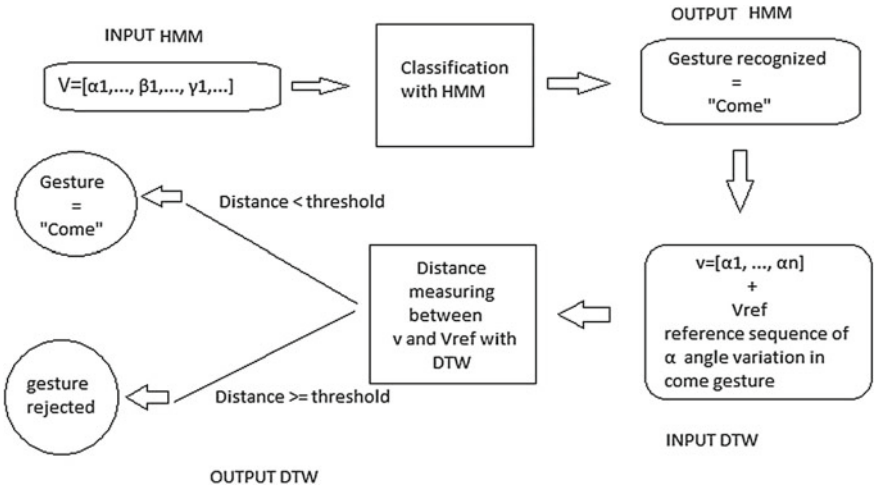

Fig. 6 Our recognition system combining HMM and DTW

(d) A set of observations $O = \{o_1, o_2, \ldots, o_t\}$, $t = \{1, 2, \ldots, T\}$ where T is the length of the longest gesture path.
(e) A set of k discrete symbols $V = \{v_1, v_2, \ldots, v_k\}$.
(f) The N-by-M observation matrix $B = \{b_{im}\}$, where b_{im} is the probability of generating the symbol v_k from state s_j and the sum of the entries in each row of the matrix B must be 1 for the same previous reason.

There are three main problems for HMM: evaluation, decoding, and training, which are solved by using Forward algorithm, Viterbi algorithm, and Baum-Welch algorithm, respectively [28]. Also, HMM has three topologies: Fully Connected (Ergodic model) where each state can be reached from any other state, Left-Right (LR) model where each state can go back to itself or to the following states and Left-Right Banded (LRB) model in which each state can go back to itself or the following state only (Fig. 7). We choose left-right banded model Fig. 7a as the HMM topology, because the left-right banded model is good for modeling-order-constrained time-

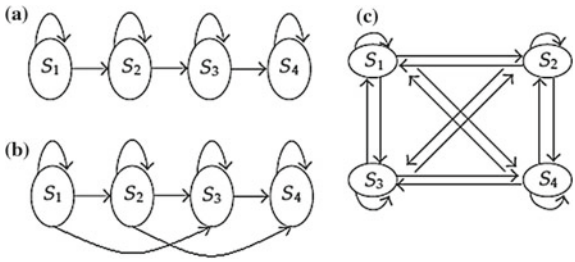

Fig. 7 HMM topologies. **a** Left-right banded topology, **b** left-right topology, **c** ergodic topology

series whose properties sequentially change over time. We realized five HMM, one HMM for each gesture type.

Initializing Parameters for LRB Model. We created five HMM, one for each gesture. First of all, every parameter of each HMM should be initialized. We start with the number of states. In our case this number is not the same for all the five HMM, it depends on the complexity and duration of the gesture. We use 12 states as maximum number and 8 as minimum one in which the HMM initial vector parameters Π will be designed by;

$$\Pi = (1\,0\,0\,0\,0\,0\,0\,0) \tag{5}$$

To ensure that the HMM begins from the first state, the first element of the vector must be 1. The second parameter to be defined is the Matrix A which can be written as:

$$A = \begin{matrix} a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{ii} & 1-a_{ii} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{ii} & 1-a_{ii} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{ii} & 1-a_{ii} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{ii} \end{matrix} \tag{6}$$

where a_{ii} is initialized by a random value. The Matrix B is determined by:

$$B = \{b_{im}\} \tag{7}$$

where b_{im} is initialized by a random value.

Training and Evaluation. Our database is composed of 100 videos for each kind gesture (50 for training and 50 for testing). In the training phase the Baum-Welch algorithm [28] is used to do a full training for the initialized HMM parameters $\lambda = (\Pi, A, B)$. Our system is trained on 50 sequences of discrete vector for each kind of gesture by using LRB topology with the number of states ranging from 3 to 12. After the training process, we obtain new HMM parameters (Π', A', B') for each type of gesture. According to the forward algorithm with Viterbi path, the other 50 video sequences for each type of gesture are tested using the new parameters. The forward algorithm computes the probability of the discrete vector sequences for all the five HMM models with different states. Thereby, the gesture path is recognized corresponding to the maximal likelihood of 5 gesture HMM models over the best path that is determined by Viterbi algorithm. The following steps demonstrate how the Viterbi algorithm works on LRB topology [29]:

- Initialization:
  for $1 \leq i \leq N$,
  $\delta_1(i) = \Pi_i \cdot b_i(o_1)$
  $\phi_1(i) = 0$
- Recursion:
  for $2 \leq t \leq T, 1 \leq j \leq N$,

$\delta_t(i) = max[\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(o_t)$
$\phi_t(i) = argmax[\delta_{t-1}(i) \cdot a_{ij}]$

- Termination:
  $p^* = max[\delta_T(i)]$
  $q_T^* = argmax[\delta_T(i)]$
- Reconstruction:
  for $T - 1 <= t <= 1$
  $q_t^* = \phi_{t+1}(q_{t+1}^*)$

The resulting trajectory (optimal states sequence) is $q_1^*, q_2^*, \ldots, q_T^*$ where a_{ij} is the transition probability from state s_i to state s_j, $b_j(o_t)$ is the probability of emitting o at time t in state s_j, $\delta_t(j)$ represents the maximum value of s_j at time t, $\phi_t(j)$ is the index of s_j at time t and p^* is the state optimized likelihood function.

Calculating the Threshold for DTW Distance. We have calculated empirically five threshold values one for each class of gesture. First we consider for each gesture its own reference sequence. For *Come* class, the reference sequence contains the variations of α angle throughout a *Come* gesture. For *Recede* class, the reference sequence contains the variations of α angle throughout a *Recede* gesture. For *Pointing to the right* class, the reference sequence contains the variations of β angle throughout a *Pointing to the right* gesture. For *Pointing to the left* class, the reference sequence contains the variations of β angle throughout a *Pointing to the left* gesture. And for *Stop* class, the reference sequence contains the variations of γ angle throughout a *Stop* gesture. The threshold of a gesture class corresponds to the maximum distance between its appropriate reference sequence and 50 sequences of test. The distance is given by DTW algorithm and the sequences of test are extracted from the training database.

3 Experimental Results

3.1 Experimental Protocol

Before the experiment, the experimental protocol was given to the subjects which describes the beginning and the end of the five gestures. The gesture duration is not fixed. The person can do a gesture whether slowly or speedy. We used the Kinect sensor that must remain stable. The distance between the kinect and the person should be between 80 cm and 3 m in order to detect the person properly. Figure 8 shows some cases when the Kinect can not totally detect the body. The environment is more or less crowded with no obstacles between the subject and the Kinect. While performing a gesture, the person should be standing and remains in front of the kinect.

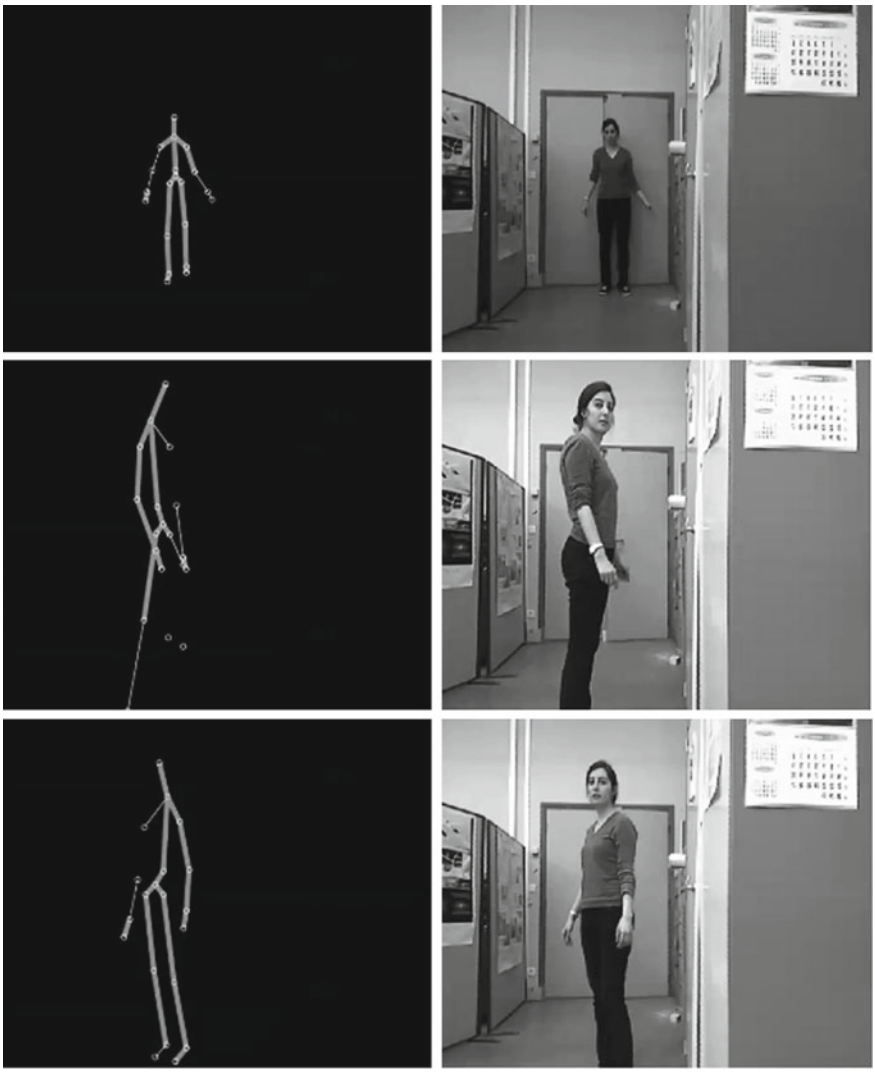

Fig. 8 The cases of detection failure by the Kinect. The first image: the distance is greater than 3 m. The second and third images: the person is not in front of the Kinect

3.2 Recognition Results

Angles variations are plotted in Figs. 9, 10, 11, 12 and 13. As it is shown, each gesture is characterized by the most changing angle comparing to the two others. We choose the state number of HMM for each gesture according to the experiment results and find that the recognition rate is maximum when the state number is 11 states for the

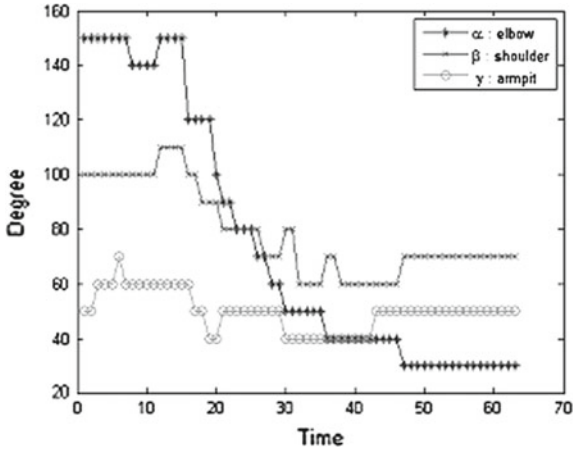

Fig. 9 Angles variations for *come* gesture

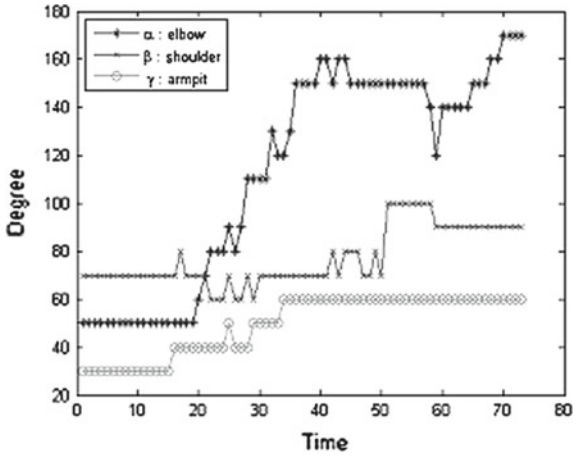

Fig. 10 Angles variations for *recede* gesture

gestures *come*, *recede* and *pointing to the right*, 12 for the gesture *pointing to left*, and 8 for the last gesture *stop* as shown in Fig. 14. Therefore, we use this setting in the following experiments. A given gesture sequence is recognized in 0.1508 s. The recognition results are listed in Table 2. We can see that the proposed method can greatly improve the recognition process, especially for opposed gestures like *come* and *recede*, *pointing to the right* and *pointing to left*. We can also see that there is no confusing between some gestures such as *come* and *recede*. In this case, it is due to the fact that the angle α changes during these two gestures decreases in *come* and increases in *recede*.

The same reasoning can be given in the case of the tow opposed gestures, *pointing to the right* and *pointing to left*. As a matter of fact, even if the same angle varies in two different gestures, our method can distinguish them.

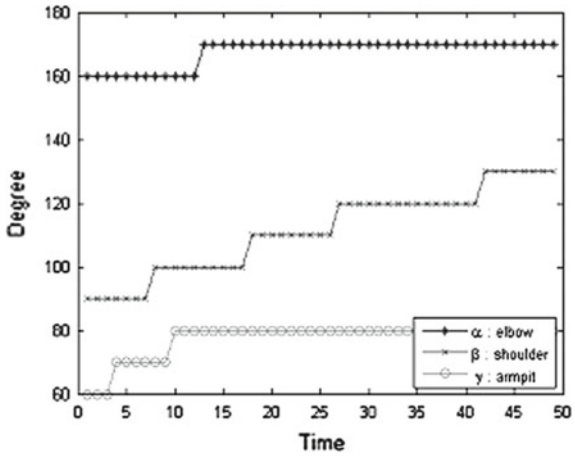

Fig. 11 Angles variations for *pointing to the right* gesture

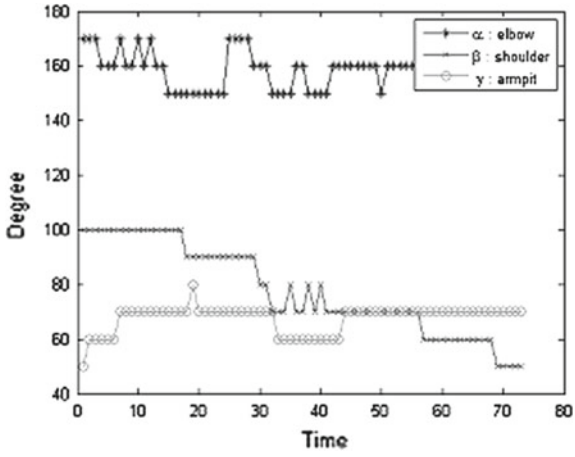

Fig. 12 Angles variations for *pointing to the left* gesture

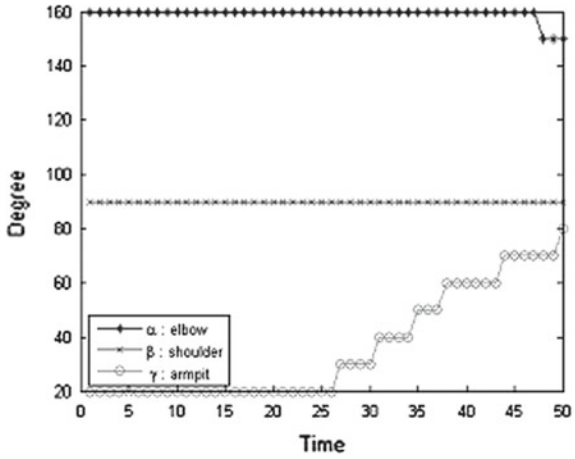

Fig. 13 Angles variations for *stop* gesture

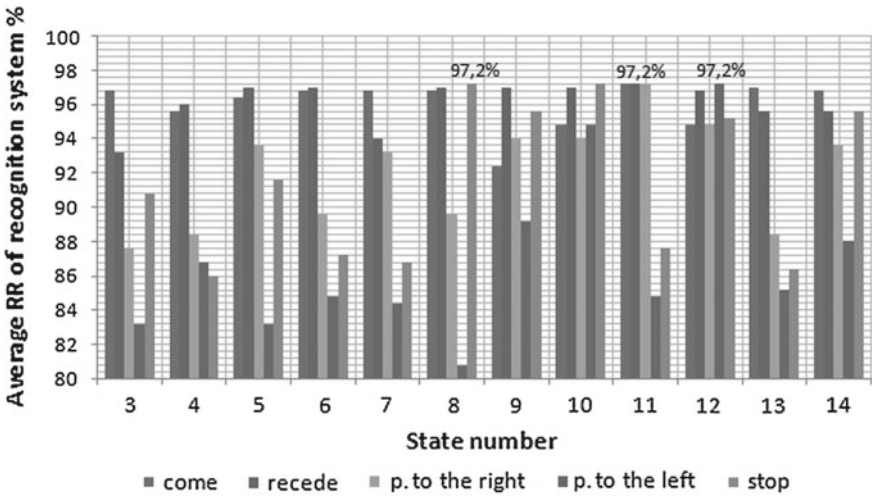

Fig. 14 Recognition accuracy when changing the number of state of HMM from 3 to 14 states

Table 2 Confusing matrix and recognition accuracy

	Come	Recede	P. to the right	P. to the left	Stop	Unknown gesture	Accuracy (%)
Come	50	0	0	0	0	0	100
Recede	0	50	0	0	0	0	100
P. to the right	0	0	49	0	0	1	98
P. to the left	0	0	0	48	0	2	96
Stop	0	0	0	0	46	4	92
Average accuracy 97.2 %							

Table 3 presents a comparison of our approach with that of the authors in [30]. They use raw, roll and pitch orientations of *elbow* and *shoulder* joints of the left arm. Their database contains five gestures trained by one person and tested by two. The gesture duration is fixed beforehand. In offline mode, the accuracy of recognizing gestures executed by persons who did training was found to be 85 % with their method and 97.2 % with our method. And without training, the recognition accuracy attained 73 % with their method and 82 % with our method. The gestures we have defined for the human robot interaction are natural. They are almost the same that we use daily and between people. Whereas, most methods in the state of the art are based on constrained gestures that use signs which are not natural. The proposed gesture recognition approach is based only on depth information that is what makes it very robust against the environment complexity and illumination variation.

Table 3 Comparison between the performance of our approach and Ye and Ha [30]'s approach

Methods	Ye and Ha [30]	Our approach
Gesture nature	Dynamic	Dynamic
Used Info.	Raw, roll and pitch orientations	Angles between joints
Gestures number	5	5
Joints number	2	5
Used data	Segmented	Brute
Classification	HMM	HMM
Database	75	500
People for test	2	21
Gesture duration	Fixed	Variable
Accuracy	73 %	97.2 %

4 Conclusion and Future Work

In this paper we have presented an efficient method for 3D natural and dynamic gesture recognition intended for human robot interaction. The proposed gesture recognition system is able to recognize five deictic gestures described by depth information only. The upper body part is tracked using Kinect camera and angles are computing from the 3d coordinates of five different joints. Our five gesture are represented by a sequence which combines the variations of three angles. This sequence is the input of our classification system that combines HMM and DTW method. First, HMM affects the given gesture to one of five classes corresponding to the maximum probability. Based on this result, the DTW measures the similarity between the variations sequence of the main angle that characterizes the gesture class which HMM method gave as output and its reference sequence. The output distance is compared to the threshold corresponding to the same class; if the distance is less than the threshold then we keep the HMM result else we reject the gesture. Experimental results presented in this paper, confirm the effectiveness and the efficiency of the proposed approach. In one hand, the recognition rate can reach up to 100 % for some kind of gestures. The combination of HMM and DTW avoid misclassification and reject bad performed gestures. In the second hand, the system can recognize gestures even if the distance or the location of people change knowing that some conditions should be respected as given in the experimental protocol. Finally, The environment and the brightness do not affect the data collection and analyzing because we rely on depth only.

Nevertheless, and despite the vast amount of relevant research efforts, the problem of efficient and robust vision based recognition of natural gestures in unprepared environments still remains open and challenging, and is expected to remain of central importance in human-robot interaction in the forthcoming years. In this context we intend to continue our research efforts towards enhancing the current system. At first, the training and test data-sets will be expanded to include richer gesture types in order

to recognize different gestures in the same sequence. Then, we intend to introduce other information such as speech to improve the proposed recognition system by detecting the beginning and the end of the gesture.

References

1. Zhang, Z.: Microsoft kinect sensor its effect. Multi Media (2012)
2. Rautaray, S., Agrawal, A.: A real time hand tracking system for interactive applications. Int. J. Comput. Appl. (2011)
3. Dan, X., Chen, Y.-L., Wu, X., Xu, Y.: Integrated approach of skincolor detection depth information for hand and face localization. In: IEEE International Conference on Robotics Biomimetics-ROBIO (2011)
4. Ghobadi, S.E., Leopprich, O.E., Hartmann, K., Loffeld, O.: Hand segmentation using 2d/3d images. In: Proceeding of image Vision Computiong (2007)
5. Barczak, A., Dadgostar, F.: Real-time hand tracking using a set of cooperative classifiers based on haar-like features. Res. Lett. Inf. Math. Sci. (2005)
6. Xu, J., Wu, Y., Katsaggelos, A.: Part-based initialization for h tracking. In: The 17th IEEE International Conference on Image Processing (ICIP) (2010)
7. Chen, Q., Georganas, N., Petriu, E.: H gesture recognition using haar-like features a stochastic context-free grammar. IEEE Trans. Instrum. Meas. (2008)
8. Bleiweiss, A., Werman, M.l.: Fusion time-of-flight depth and color for real-time segmentation and tracking. In: DAGM Symposium for Pattern Recognition (2009)
9. Xu, D., Wu, X., Chen, Y., Xu, Y.: Online dynamic gesture recognition for human robot interaction. IEEE J. Intell. Robot. Syst. (2014)
10. Lu, X., Chen, C.-C., Aggarwal, J.K.: Human detection using depth information by kinect. In: Computer Society Conference on Computer Vision Pattern Recognition-CVPR (2011)
11. Qin, S., Zhu, X., Yang, Y., Jiang, Y.: Real-time hand gesture recognition from depth images using convex shape decomposition method. J. Signal Process. Syst. (2014)
12. Matuszek, C., Bo, L., Zettlemoyer, L., Fox, D.: Learning from unscripted deictic gesture language for human-robot interactions. I. J. Robotic (2014)
13. Boesen, A., Larsen, L., Hauberg, S., Pedersen, K.S.: Unscented Kalman filtering for articulated human tracking. In: 17th Scinavian Conference SCIA (2011)
14. Ababsa, F.: Robust extended Kalman filtering for camera pose tracking using 2D to 3D lines correspondences. In: International Conference on Advanced Intelligent Mechatronics (2009)
15. Ababsa, F., Mallem, M.: Robust line tracking using a particle filter for camera pose estimation. In: Proceedings of the ACM Symposium on Virtual Reality Software Technology (2006)
16. Koller, D., Thrun, S., PlagemannVarun, C., Ganapathi, V.: Real time identification localization of body parts from depth images. In: IEEE International Conference on Robotics Automation (ICRA) (2010)
17. Wang, X., Xia, M., Cai, H., Gao, Y., Cattani, C.: Hidden Markov models based dynamic hand gesture recognition. Math. Probl. Eng. (2012)
18. Saon, G., Chien, J.T.: Bayesian sensing hidden Markov models. IEEE Trans. Audio Speech Lang. Process. (2012)
19. Li, M., Cattani, C., Chen, S.Y.: Viewing sea level by a one-dimensional random function with long memory. Math. Probl. Eng. (2011)
20. Elmezain, M., Al-Hamadi, A., Michaelis, B.: Real-time capable system for hand gesture recognition using hidden Markov models in stereo color image sequences. J. WSCG (2008)
21. Binh, N.D., Ejima, T.: Real-time hand gesture recognition using pseudo 3-d hidden Markov model. In: Proceedings of the 5th IEEE International Conference on Cognitive Informatics (ICCI'06) (2002)

22. Eickeler, S., Kosmala, A., Rigoll, G.: Hidden Markov model based continuous online gesture recognition. In: Proceedings of 14th International Conference on Pattern Recognition (1998)
23. Gu, Y., Do, H., Ou, Y., Sheng, W.: Human gesture recognition through a kinect sensor. In: International Conference on Robotics Biomimetics (2012)
24. Bengio, Y., Frasconi, P.: Input-output HMMs for sequence processing. IEEE Trans. Neural Netw. (1996)
25. Wang, S., Quattoni, A., Morency, L., Demirdjian, D., Darrell, T.: Hidden conditional rom fields for gesture recognition. In: IEEE Computer Society Conference on Computer Vision Pattern Recognition (CVPR) (2006)
26. Corradini, A.: Dynamic time warping for off-line recognition of a small gesture vocabulary. In: ICCV Workshop on Recognition Analysis Tracking of Faces Gestures in Real-Time Systems (2001)
27. Hiyadi, H., Ababsa, F., Montagne, Ch., Bouyakhf, E.H., Regragui, F.: A depth-based approach for 3D dynamic gesture recognition. ICINCO 103-110 (2015)
28. Lawrence, R.: A tutorial on hidden Markov models selected applications in speech recognition. Proc. IEEE (1989)
29. Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B.: A hidden Markov model-based isolated meaningful hand gesture recognition. J. WSCG (2009)
30. Ye, G., Ha, D., Yongsheng, O., Weihua, S.: Human gesture recognition through a kinect sensor. Robotics Biomimetics (ROBIO) (2012)