A Depth-based approach for 3D dynamic gesture recognition
Hajar Hiyadi, Fakhr-Eddine Ababsa, Christophe Montagne, El Houssine Bouyakhf, Fakhita Regragui

To cite this version:

HAL Id: hal-01177438
https://hal.archives-ouvertes.fr/hal-01177438
Submitted on 20 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Depth-based approach for 3D dynamic gesture recognition

Hajar Hiyadi1,2, Fakhreddine Ababsa1 Christophe Montagne1, El Houssine Bouyakhf2 and Fakhita Regragui2
1Evry Val d’Essonne University, Evry, France
2Mohammed V University, Rabat, Morocco
{Hajar.Hiyadi, Fakhr-Eddine.Ababsa, christophe.Montagne}@ufrst.univ-evry.fr, {bouyakhf, regragui}@fsr.ac.ma

Keywords: 3D gesture recognition, Gesture tracking, Depth image, Hidden Markov Models.

Abstract: In this paper we propose a recognition technique of 3D dynamic gesture for human robot interaction (HRI) based on depth information provided by Kinect sensor. The body is tracked using the skeleton algorithm provided by the Kinect SDK. The main idea of this work is to compute the angles of the upper body joints which are active when executing gesture. The variation of these angles are used as inputs of Hidden Markov Models (HMM) in order to recognize the dynamic gestures. Results demonstrate the robustness of our method against environmental conditions such as illumination changes and scene complexity due to using depth information only.

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 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 flexion of the members into electrical signals to determine the posture. These methods are based on various informations 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 (Zhang, 2012), 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 (Rautaray and Agrawal, 2011), 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 (Xu et al., 2011) 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 (Ghobadi et al., 2007) 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 (Barczak and Dadgostar, 2005; Chen et al., 2008; Xu et al., 2010). 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 (Bleiweiss and Werman, 2009; Xia et al., 2011; Qin et al., 2014; Xu et al., 2014). Other works combine depth information, color and speech (Matuszek et al., 2014). In (Xia et al., 2011), 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 (Boesen et al., 2011), the Extended Kalman Filter (F., 2009) and the Prticle Filter (F. and Mallem, 2006). 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 (Koller et al., 2010). 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 (Wang et al., 2012). As in speech, hand writing and character recognition (Saon and Chien, 2012; Li et al., 2011), HMM were successfully used in gesture recognition (Elmezain et al., 2008; Eickeler et al., 1998; Binh and Ejima, 2002). Actually, HMM can model spatial-temporal time series and preserve the spatial-temporal identity of gesture. The authors in (Gu et al., 2012) 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) (Bengio and Frasconi, 1996), Hidden Conditional Random Fields (HCRF) (Wang et al., 2006) and Dynamic Time Warping (Corradini, 2001) are also used to model and recognize sequences of gestures.

In this paper, we propose a 3D dynamic gesture recognition technique based on depth camera. The basic framework of the technique is shown in Figure 1. The Skeleton algorithm given by the Kinect SDK is used for body tracking. The 3D joints informations are extracted and used to calculate new and more relevant features which are the angles between joints. Finally, discrete HMM with Left-Right Banded topology are used to model and classify gestures. The evaluation experiments show the effectiveness of the proposed technique. The performance of our technique is further demonstrated with the validation step which give good recognition even without training phase.

The rest of the paper is organized as follows: Section 2 describes our 3D dynamic gesture approach and the features we used. Section 3 gives some experimental results. Finally, section 4 ends the paper with a conclusion and futur 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 recognition.
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. This method projects a skeleton on the human body image so each joint 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 mean idea of our approach is to estimate in real time the variations of the active angles while executing the gestures. The considered angles are: $\alpha$ elbow, $\beta$ shoulder and $\gamma$ armpit angle, as shown in Figure 5. Each angle is then computed from the 3D coordinates of the three joints that are commonly accounted to it:

- $\alpha$ elbow angle is computed from the 3D coordinates of elbow, wrist and shoulder joints.
- $\beta$ shoulder angle is computed from the 3D coordinates of shoulder, elbow and shoulder center joints.
- $\gamma$ 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:

$$V_\alpha = [\alpha_1, \alpha_2, ..., \alpha_T]$$  \hspace{1cm} (1)

$$V_\beta = [\beta_1, \beta_2, ..., \beta_T]$$  \hspace{1cm} (2)

$$V_\gamma = [\gamma_1, \gamma_2, ..., \gamma_T]$$  \hspace{1cm} (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:

$$V_a = [\alpha_1, \alpha_2, ..., \alpha_T, \beta_1, \beta_2, ..., \beta_T, \gamma_1, \gamma_2, ..., \gamma_T]$$  \hspace{1cm} (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 varying for *come* and *recede* is the angle $\alpha$. Likewise, the angle $\gamma$ for *stop* gesture, and angle $\beta$ 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 (1, 2, 3, 4, 5 refer respectively to *come*, *recede*, *pointing to right*, pointing to left, *stop*).

<table>
<thead>
<tr>
<th></th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$180^\circ \rightarrow 30^\circ$</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>$30^\circ \rightarrow 180^\circ$</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>$90^\circ \rightarrow 150^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>$90^\circ \rightarrow 40^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>$30^\circ \rightarrow 80^\circ$</td>
</tr>
</tbody>
</table>

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 Hidden Markov Models (HMM). HMM are widely used in temporal models, 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.

#### 2.2.1 Hidden Markov Models

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

- a) A set of $N$ states $S = \{s_1, s_2, ..., s_N\}$.
- b) An initial probability distribution for each state $\Pi = \{\pi_j\}, j = \{1, 2, ..., N\}$, 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.
- d) A set of observations $O = \{o_1, o_2, ..., o_t\}$, $t = \{1, 2, ..., T\}$ where $T$ is the length of the longest gesture path.
- e) A set of $k$ discrete symbols $V = \{v_1, v_2, ..., v_k\}$.
- 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 (Lawrence, 1989). 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 to the following state only (Figure 6). We choose left-right banded model Figure 6(a) as the HMM topology, because the left-right banded model is good for modeling-order-constrained time-series whose properties sequentially change over time. We realized five HMM, one HMM for each gesture type.

![Figure 6: HMM topologies.](image)

#### 2.2.2 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
The following steps demonstrate how the Viterbi algorithm works on LRB topology (Elmezain et al., 2009): the best path that is determined by Viterbi algorithm. The resulting trajectory (optimal states sequence) is \( q_1^*, q_2^*, ..., q_T^* \) 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.

3 EXPERIMENTAL RESULTS

3.1 Data Set

Our database is composed of 100 videos for each kind of gesture. The environment and the brightness do not affect the performance. The gesture duration is fixed. The person can do a gesture whether slowly or speedy. We used the Kinect sensor that must remain stable. The person must be in front of the kinect and the distance must be higher than 80 cm to well detect the body. The environment is sort of crowded but no barrier should be between the person and the camera to avoid losing tracking. During the gesture the person should stay up.

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. The environment is sort of crowded but no barrier should be between the person and the camera to avoid losing tracking. During the gesture the person should stay up.

The environment and the brightness do not affect the data collection because we rely on depth only. A given gesture is recognized corresponding to the maximum likelihood of five HMM models. Therefore, if a new executed gesture does not correspond to the five gestures, it will be awarded to one of five classes corresponding to the maximum probability and then recognized as one of them. To overcome this problem we built a new database of 20 videos containing the insignificant gestures when subject moves his hand without any goal. The probabilities of belonging to

\[
\begin{align*}
\lambda & = (\Pi, A, B) \\
\Pi & = (10000000) \\
A & = \begin{pmatrix}
a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_{ii} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{ii} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a_{ii} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a_{ii} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a_{ii} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{ii}
\end{pmatrix} \\
B & = \{b_{im}\}
\end{align*}
\]

where \( b_{im} \) is initialized by a random value. The Matrix \( B \) is determined by:

\[
\begin{align*}
\delta_t(i) & = \Pi_i \cdot b_j(o_t) \\
\phi_t(i) & = 0
\end{align*}
\]

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

2.2.3 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 (Lawrence, 1989) 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 \( (\Pi', A', B') \) 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 (Elmezain et al., 2009):

- **Initialization:**
  for \( 1 \leq i \leq N \),
  \[ \delta_1(i) = \Pi_i \cdot b_j(o_1) \]
  \[ \phi_1(i) = 0 \]

- **Recursion:**
  for \( 2 \leq t \leq T \), \( 1 \leq j \leq N \),
  \[ \delta_t(i) = \max \{ \delta_{t-1}(j) \cdot a_{ij} \} \cdot b_j(o_t) \]
  \[ \phi_t(i) = \arg\max \{ \delta_{t-1}(j) \cdot a_{ij} \} \]
the five classes are very small. From here we determined a threshold for each class gesture. Thus, the gesture is rejected if the maximum probability is less than the threshold fixed for the corresponding gesture class.

### 3.3 Recognition Results

Angles variations are plotted in Figure (7, 8, 9, 10 and 11). 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 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 Figure 12. 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 $\alpha$ changes during these two gestures decreases in *come* and increases in *recede*. The same reasoning can be given in the case of the two 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.

Table 3 presents a comparison of our approach with that of the authors in (G. et al., 2012). 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 2: Confusing matrix and recognition accuracy.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>1</td>
<td>98%</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>48</td>
<td>2</td>
<td>96%</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>46</td>
<td>92%</td>
</tr>
</tbody>
</table>

Average accuracy 97.2%

4 CONCLUSIONS AND FUTURE WORKS

We described an efficient method for 3D natural and dynamic gesture recognition for human robot/interaction. We have identified five deictic gestures, which can be recognize using only depth information. The idea is to extract the 3D coordinates of the joints of the upper part of the human body, and then compute the angles corresponding to these joints. These angles variation along the gestures are used as inputs of Hidden Markov Models (HMM). We propose one model for each gesture. The experimental results show that our approach gives better recognition compared to the method in (G. et al., 2012). Indeed, the recognition rate can reach up to 100% for some kind of gestures. In addition, we give below some characteristics of the proposed recognition system. First, the training phase, it simply saves the gesture when run. Second, the system can recognize gestures even if the distance or the location of people change. Third, although the speed of gestures can vary from one person to another, the system is able to recognize the gesture. Finally, the change in the duration of a gesture from one person to another does not affect the recognition. In the future work, we will expand our gesture database in order to recognize different gestures in the same sequence, we will also combine the depth information with speech to make automatic the detection of the beginning and the end of the gesture and make the complex gesture recognition more robust.
REFERENCES


Saon, G. and Chien, J. T. (2012). Bayesian sensing hid-


