DBN versus HMM for Gesture Recognition in Human-Robot Interaction

B. Burger\textsuperscript{1,2,3}, G. Infantes\textsuperscript{4}, I. Ferrané\textsuperscript{1,3}, F. Lerasle\textsuperscript{1,2}

\textsuperscript{1} Université de Toulouse; UPS, INSA, INP, ISAE; IRIT, LAAS : F-31077 Toulouse, France
\textsuperscript{2} CNRS; LAAS; 7 avenue du Colonel Roche, 31077 Toulouse Cedex, France
\textsuperscript{3} IRIT; 118 route de Narbonne, 31062 Toulouse Cedex, France
\textsuperscript{4} Onera; 2 avenue Edouard Belin, 31055 Toulouse Cedex 4, France

Author’ email: \{bburger,lerasle\}@laas.fr, ferrane@irit.fr, Guillaume.Infantes@onera.fr

Abstract: We designed an easy-to-use user interface based on speech and gesture modalities for controlling an interactive robot. This paper, after a brief description of this interface and the platform on which it is implemented, describes an embedded gesture recognition system which is part of this multimodal interface. We describe two methods, namely Hidden Markov Models and Dynamic Bayesian Networks, and discuss their relative performance for this task in our Human-Robot interaction context. The implementation of our DBN-based recognition is outlined and some quantitative results are shown.

I. INTRODUCTION

Since assistant robots are designed to directly interact with people, finding natural and easy-to-use user interfaces is of fundamental importance \cite{1}. Nevertheless, few robotic systems are currently equipped with a completely on-board multimodal user interface enabling robot control through communication channels like speech, gesture or both. The most advanced one is \cite{2} in which a constraint based multimodal system for speech and 3D pointing gestures has been developed, but gesture recognition is limited to mono-manual pointing gestures. In other works, like \cite{3} and \cite{4}, gesture recognition is often extracted from monocular images, loosing the depth information and thus losing the capability of dealing with a pointing gesture other than directional. With the intention of providing our interactive robot called Jido with such an interface, we developed both speech and gesture recognition systems as well as a module for fusing these two information results. This merging step enables to:

− complete an underspecified sentence, an abbreviation or an omission, which is usual in human communication particularly if a gesture can be done or even used instead
− strengthen each modality by improving the classification rates of multimodal commands thanks to a probabilistic merge of gesture and speech recognition results.

In this framework, this paper focuses on our one- and two-handed gesture recognition system given the video stream delivered by the on-board stereo head, with the physical constraints imposed by autonomous robotic systems in background: mobility of the platform, limited and shared computational power, limited memory capacities, etc.

First section describes as a background our platform and the interface we developed on it, leading to an explanation of our needs in gesture recognition. Next, we discuss the relative performance of Hidden Markov Models (HMM) and Dynamic Bayesian Networks (DBN) for such a task, given the output of our 3D visual tracker devoted to the upper human body extremities \cite{5}. Then, the implementation of our DBN-based recognition is outlined. We describe more precisely the data clustering process which is carried out thanks to a Kohonen network, the model training made by means of an Expectation-Maximization based algorithm and the recognition performed using particle filtering \cite{6}. Finally, some qualitative and quantitative results from a symbolic and deictic gesture database are presented. The DBN representation, which is commonly used for human activity recognition, is shown to outperform the HMM representation especially in terms of CPU time consuming and gesture segmentation.
II. BACKGROUND

A. **Our multimodal interface**

Our robot, called Jido, is especially equipped with a 6-DOF arm, a pan-tilt stereo system on a mast, a headset microphone for the user and two laser scanners (you can see the robot in Figure 1). Jido’s functionalities are managed thanks to the “LAAS” layered software architecture which divide these functionalities in specialised modules. From its sensors and actuators, the robot has been endowed with a set of basic functions that enables it to navigate in its environment and to recognize and manipulate objects. Other functionalities for human perception are working on it too: (i) face detection/recognition and view-based tracking from the module ICU, (ii) control of the pan-tilt unit mounted stereo head from the module PTU, (iii) gesture tracking and recognition from the module GEST, (iv) speech utterance interpretation from the module RECO, (v) fusion of speech and gestures results from the module FUSION. The last two modules are briefly described here below while the module GEST embeds the gesture recognition process which is described in more details in the next sections.

The RECO module processes French utterances in two steps. First, the sentence uttered by the user is recognized thanks to a HMM-based speech engine called Julian using grammar as language model. The required linguistic resources for this task are: a set of acoustic models for French phonetic units (drawn up from the French lexical database BDLEX and adapted to our applicative context) and a set of specifically designed grammars. Second, in order to make the robot understand what the user has just said, semantic units have to be extracted from the recognized sentence. Some of these units are related to actions while others are related to objects or their own attributes like colour or size as well as location or robot configuration parameters (speed, heading, distance). Evaluations carried out on 1200 various sentences uttered by 16 different speakers, showed an Interpretation Error Rate (IER) of 6%.

The module FUSION merges gesture recognition and speech interpretation outputs thanks to a late-stage and hierarchical strategy. Speech is used as the main channel and actions needing a gesture disambiguation are identified by the module RECO. Following a rule based approach, the command generated by RECO is completed. Thus, for human-dependent commands e.g. “viens ici” (“come here”), the human position and the pointed direction are characterized thanks to the 3D visual tracker. Late-stage fusion consists of fusing the confidence scores for each N-best hypothesis produced by the speech and vision modules like in [13].

This architecture and its components allow us to perform complex multimodal scenarios for human-robot interaction. Figure 1 shows two snapshots extracted from our last experiment dedicated to multimodal commands and based on speech and gesture recognition. The goal was to make the robot follow the user to a table on which some bottles were put down, to pick up the bottle pointed by the user and finally to give it to him/her.

---

**Fig. 1:** Two snapshots from our last experiment including gesture recognition.

A more detailed description of the components of our system as well as some experimental results based on a previous scenario can be found by the interested reader in [5] and further information and related movies at the URL [www.laas.fr/~bburger](http://www.laas.fr/~bburger).
B. Gesture recognition

In previous work [5], a primary gesture recognition system was developed using HMMs to recognize 8 gestures. From the experiments performed around a multimodal scenario emerges that this gesture recognition system was really costly in terms of computational time. In light of this observation and following our will to increase the number of possible gestures to recognize, we decided to explore the field of DBNs. The first goal was to decrease this computational cost while keeping the recognition rate acceptable or even increasing it. But what we were also interested in, was the capacity of DBNs to easily add new kind of information like the coarse orientation of the head or discreet variables describing the context.

In order to recognize gestures, we need to extract them from the binocular video stream. The full tracking of all the human body kinematics is not essential for this task. In [5], we proposed to model solely the upper human body extremities thanks to deformable and coarse ellipsoids for computational tractability and person-independency reasons. Given this model, we proposed an interactively distributed multiple object tracker based on the well-known particle filtering framework to (i) ensure the 3D consistencies between the targeted human body parts, (ii) limit the phenomena of “error merge” and “labeling” problems when partial or complete self-occlusion occurs while performing natural gestures (Figure 2-left). Thanks to this tracking step, temporal sequences of hand poses relatively to the head are extracted and can be analysed to look for the occurrence of a one- or two-handed gesture. Our investigations detailed hereafter are considering a 12 gesture set:

- 7 symbolic gestures defined by their motion templates, namely: “calling out” (with one or two hands), “introducing oneself”, “come to me” (with one or two hands), “stop”, “go away”.
- 5 deictic gestures depending on the coarse pointed direction relatively to the user who performs the gesture i.e. “in front of”, “bottom left”, “bottom right”, “top left”, “top right”. The pointing direction is calculated by the connecting line between the centre of the head and the hand in 3D.

III. METHODS DESCRIPTION

A. Methods description

Template matching and especially state-space based approaches are widely used for gesture recognition. Hidden Markov Models (HMMs) are well-known for their use in speech recognition [7] and their variants have become most popular [8]. More general temporal dependency models e.g. Dynamic Bayesian Network (DBN) [9] have been adopted for the modeling and the recognition of human activities [10]. But, this generic representation has not yet been applied to the gesture recognition field. Figure 2 (centre and right) depicts the two explored representations for gesture recognition. DBN provides a unified probability model contrarily to HMMs which requires one model per gesture. For each HMM, \( H_t \) and \( O_t \) denote the hidden state and the observation vector at time \( t \). In the single DBN model, the node \( G \) represents the gesture values while the other nodes are the observation variables.
The DBN framework generalises the HMM by representing the states in terms of state variables which can have complex interdependencies. This offers dual advantages. First, a broad variety of modeling schemes can be conceptualised in a single framework with an intuitive graphical notation. Second, dependencies between the variables can be modeled, their graphical structure providing a simple way to specify conditional independencies and hence to provide a compact parametrisation of the model. These advantages provide the DBN framework with a higher flexibility than HMMs, leading to a sharper modelisation and consequently to a better recognition rate if the model is representative enough. But a linked drawback is the complexity: a good model is not easy to find because of the high number of possibilities and the lack of automatic modelisation algorithm.

In order to classify a gesture instance, the HMM method leads to a simple solution: given a bank of HMM, the model with the highest likelihood is selected and the gesture is classified correspondingly. With the DBN method, the gesture can be guessed, its weights being updated thanks to particle filtering and the one with the highest score is finally selected.

B. Implementation

We assume that all the gestures start and finish in the same rest positions (the hands lying along the thighs). The features used as model inputs are derived from the outputs of our multi-tracker, and in order to make our gesture recognition independent of the user position in relation to the robot, we define our observation vector from the hand positions relative to a spherical coordinate \((\rho, \vartheta, \phi)\) centred on the head. The \(y\) and \(z\) axes of this coordinate represent the “human plane” i.e. the plane formed by the head and both hands at his rest position. Then, we can write the 7-dimensional feature vector corresponding to the frame at time \(t\) as:

\[
o_t = (\rho_R, \vartheta_R, \phi_R, \rho_L, \vartheta_L, \phi_L, D_{H_R-H_L}),
\]

where the feature \(\rho_{R,L}, \vartheta_{R,L}, \phi_{R,L}\) are the spherical coordinates of the right \(R\) and left \(L\) tracked hand and \(D_{H_R-H_L}\) is the distance between the two hands. In this study, we use discrete HMMs and DBNs, this means that the feature vectors have to be clustered. For an efficient discretization, their observations space size (number of clusters per observed variable) and geometry (size of each of these clusters) are determined through a self-organizing map, or Kohonen network [11]. In order to find the optimal topologies of HMM and DBN as well as free parameters (cluster number of Kohonen network, number of particles, etc.), the following protocol is used. Following a three-fold cross-validation strategy, our gesture database is splitted into three subsets: two of them are used for training and the other one for testing each representation. This process is repeated three times with each of the three subsets used exactly once as the validation data. Then, the processed results are averaged to produce the overall evaluation. The training of the HMM bank is achieved thanks to a classical Expectation-Maximization (EM) algorithm which leads to the best results with 7 states per HMM. The HMM representation assumes that the performed gestures are starting and ending in the same natural/rest position as it requires a finite number of observations. Conversely, for the DBN training, we use an adaptation of this algorithm maintaining an approximated “belief state” i.e. a probabilistic hypothesis over the possible assignment of the variable values, by means of a particle filter (see [9] for more details). Thus, we can marginalize over the \(G\) (i.e. gesture) variable, giving us at each particle filtering step a weight for every gesture modeled. When a gesture consecutively “wins” several times with a sufficiently discriminant score, the algorithm can terminate itself without waiting until the end of the sequence and thus can save CPU resource. What emerges from the evaluations of different DBN topologies is that the one depicted in Figure 2-right leads to higher performances.

Different free parameters are needed to perform our gesture recognition. Some of them directly influence the recognition performance and are not easy to determine empirically in a suitable manner, that is why we used a genetic algorithm [12], which is a well-known technique for optimizing the free parameters of a classifier. Finally, both trained HMM and DBN models representations are evaluated and results are reported in the next section.

IV. COMPARATIVE RESULTS AND INTEGRATION
In order to evaluate and realistically compare HMMs and DBNs, 11 people performed each of our 12 gestures from 5 up to 10 times, leading to a set of 772 sequences. Each of these sequences consists of the outputs of the gesture tracking performed on our robot Jido (an example is shown in Figure 2-left while our robot can be seen in Figure 1).

In our experiments, we were particularly interested in comparing: (1) the recognition rate of HMM versus DBN, and (2) the CPU computation time on a 3.2 GHz Pentium PC. Figure 3 illustrates these results when increasing the set of gestures to be recognized. The recognition rate logically decreases with the size of this gesture set because similarities between some gestures rise. But it is maintained over a satisfactory level with 79.7% of the whole database sequences correctly classified using a single DBN while the HMM bank leads to a score of 76.0%. Most importantly, especially in our robotic context, DBN imposes less computational burden (factor 3) than HMM. Furthermore, it is worth noting that for DBN it is possible to greatly decrease the computation time without too strongly degrading the recognition rate. For instance, we achieved to obtain an average CPU time consuming of only 24 ms per recognition for a recognition rate of 72.7%.

Table 1 details the results of our DBN-based gesture recognition system for the overall gesture set. On the diagonal is the amount of gesture correctly recognized. The most important errors coherently arise from some gestures similarities. This kind of misclassification goes up with the variability of the gestures in the database which stems from the number of people who performed those gestures. We remind that sensibility measures the proportion of gestures correctly identified while the specificity measures the proportion of gestures correctly rejected, i.e. they respectively measure the capability of the classifier to detect true positives and to reject false positives.

**TABLE 1. Recognition results for our gesture recognition by DBN (in %).**

<table>
<thead>
<tr>
<th>Gestures given</th>
<th>Gestures recognized</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>&quot;stop&quot;</td>
<td>(1)</td>
</tr>
<tr>
<td>&quot;come to me (one hand)&quot;</td>
<td>(2)</td>
</tr>
<tr>
<td>&quot;come to me (two hands)&quot;</td>
<td>(3)</td>
</tr>
<tr>
<td>&quot;pointing bottom right&quot;</td>
<td>(4)</td>
</tr>
<tr>
<td>&quot;pointing top right&quot;</td>
<td>(5)</td>
</tr>
<tr>
<td>&quot;pointing bottom left&quot;</td>
<td>(6)</td>
</tr>
<tr>
<td>&quot;pointing top left&quot;</td>
<td>(7)</td>
</tr>
<tr>
<td>&quot;pointing in front of&quot;</td>
<td>(8)</td>
</tr>
<tr>
<td>&quot;introducing oneself&quot;</td>
<td>(9)</td>
</tr>
<tr>
<td>&quot;go away&quot;</td>
<td>(10)</td>
</tr>
<tr>
<td>&quot;calling out (one hand)&quot;</td>
<td>(11)</td>
</tr>
<tr>
<td>&quot;calling out (two hands)&quot;</td>
<td>(12)</td>
</tr>
</tbody>
</table>

V. CONCLUSION
This paper presents a gesture recognition system dedicated to Human-robot interaction. A comparative study of Hidden Markov Models (HMMs) vs Dynamic Bayesian Networks (DBN) performances for such a task is described and show some advantages of the DBN framework (higher flexibility, intuitiveness in model designing, etc) as well as some drawbacks (mainly, its potential complexity). Evaluations highlight that our DBN-based recognition outperforms the HMM representation in terms of recognition error rate as well as computational cost. The integration of this gesture recognition tool into our multimodal interface and on our robot Jido is briefly described.

Besides planed improvements of the different modules composing our multimodal interface (tracking, speech understanding, multimodal fusion), further investigations about gesture recognition will concern automatic segmentation of gesture without rest position. In fact, a segmentation based on a fixed position is not very convenient and consequently unrealistic for a so called “natural” human-robot communication. A second extension we consider is to incorporate head orientation into our feature vector: a human often looks at the direction he is pointing at. Finally, a third extension could be about a more advanced framework for modelling gestures. A lead we follow is about building a gesture grammar with grammar-based speech recognition as an inspiration.

REFERENCES