Gesture Recognition Using a Marionette Model and Dynamic Bayesian Networks (DBNs)

Jörg Rett and Jorge Dias

Institute of Systems and Robotics
University of Coimbra
Polo II, 3030-290 Coimbra, Portugal
{jrett, jorge}@isr.uc.pt

Abstract. This paper presents a framework for gesture recognition by modeling a system based on Dynamic Bayesian Networks (DBNs) from a Marionette point of view. To incorporate human qualities like anticipation and empathy inside the perception system of a social robot remains, so far an open issue. It is our goal to search for ways of implementation and test the feasibility. Towards this end we started the development of the guide robot 'Nicole' equipped with a monocular camera and an inertial sensor to observe its environment. The context of interaction is a person performing gestures and 'Nicole' reacting by means of audio output and motion. In this paper we present a solution to the gesture recognition task based on Dynamic Bayesian Network (DBN). We show that using a DBN is a human-like concept of recognizing gestures that encompass the quality of anticipation through the concept of prediction and update. A novel approach is used by incorporating a marionette model in the DBN as a trade-off between simple constant acceleration models and complex articulated models.

1 Introduction

Nowadays, robotics has reached a technological level that provides a huge number of input and output modalities. The future higher level cognitive systems must benefit from the technological advances and offer an effortless and intuitive way of interacting with a robot to its human counterpart.

Our solution to this problem is a cognitive system for a robot that mimics human perception in two aspects: Anticipation, the ability to predict future situations in a world physically in motion and empathy, the ability to estimate the intentions of our populated environment. Towards this end we define a library of intuitive and useful gestures divided in three categories, including some earlier sources of research [1], [2]. We present a system that extracts features of gestures from a monocular camera image by detecting and markerless tracking of hands and face. Our approach also incorporates face recognition to take advantage of a person's "individual" gesture pattern. We contribute a novel probabilistic model using the framework of Dynamic Bayesian Networks (DBNs) to anticipate the gesture given the observed features. DBNs offer combinations of the whole

A. Campilho and M. Kamel (Eds.): ICIAR 2006, LNCS 4142, pp. 69–80, 2006.

family of probabilistic tools like Hidden Markov Models (HMMs), Kalman Filters and Particle Filters and their various modifications. Though, DBNs can be used for all kind of system modeling (e.g. navigation, speech recognition, etc.) they are specially suited for cognitive processes. The process of prediction and update represents an intrinsic implementation of the mental concept of anticipation. Furthermore these methods have already proven their usability in gesture recognition [3, 4]. To enhance the quality of inference and learning we introduce the marionette concept as a physical model of human motion to support the probabilistic model. The concept which was inspired by research on human behavior [5] represents a trade-off between simple constant acceleration models and complex articulated models.

The development of our guide robot named 'Nicole' will receive this systems as a part of it's abilities to interact. Nicole will be able to guide visitors through our Lab, talk about the research and react on gestures performed by people recognized as "godfathers". We will be able to test several human-robot interaction scenarios to answer and probably also raise some questions related with "social robots". Some examples of successful development of guide robots is the development of a family of robot guides serving at the Carnegie Museum of Natural History as docents for five years [6]. The autonomous tour-guide/tutor robot *RHINO* which was deployed in the "Deutsches Museum Bonn" in 1997 [7] and the mobile robot *Robox* which operated at the Swiss National Exhibition Expo02 [8].

Section 2 introduces a concept to analysis gestures and the definition of a gesture library. Section 3 introduces the marionette model and the computational solution for gesture interpretation using a Hidden-Markov-Model framework. Section 4 presents the architecture of the guide robot 'Nicole' and aspects of its implementation. Section 5 explains the process of feature extraction and the techniques used. Section 6 shows how the features are registered in a gesture plane by fusing camera and inertial data. Section 7 shows results on the registration of gesture trajectories. Section 8 closes with a discussion and an outlook for future work.

2 Means of Interaction – Gesture Libraries

The communication from the human to the robot will be based on hand movements conveying useful information, i.e. hand gestures. This raises two questions to be answered: 1. What makes a movement to appear as a gesture? 2. What is a useful set of gestures? To tackle the first question we start with a concept proposed for human motion analysis. As a gesture is created by motion we need to find an appropriate description for the spatio-temporal behavior. Our aim is to define 'atomic' segments of gestures which we can relate to our observation sequence.

A suitable model is to divide the gesture into three phases [9]: 1. *Pre-stroke* (preparation), 2. *Stroke* and 3. *Post-stroke* (retraction). Figure 2 a) - c) shows an example for a deictic gesture (i.e. pointing gesture). Gesture recognition systems

Table 1. Gesture-Action Mapping

Gesture	Action	Category
Circle	Turn 360	1
Horizontal Line	Sway left and right	1
Hand next to the ear	Speak louder	1
Finger over the mouth	Speak lower voice	1
Oscillation to the front	Go back	1
Oscillation over shoulder	Come close	1
Oscillating left or right	Go left or right	1
Waving Bye-Bye	Call\Send Away Nicole	3
Pointing gesture	Look There	2
Oscillating pointing gesture	Go There	1

have often adopted this temporal composition [3,4]. In [10] the phases are called 'phonemes' following the terms used in phonology to describe the principal sounds in human languages.

The second question may be expressed in a more general way as: What kind of knowledge about the world do I need to provide to the robot? The set of gestures need to be rich enough to trigger a certain variety of actions and the gestures must be intuitively and effortlessly performed by the human. The 'Nicole' dictionary maps a set of gestures into actions to be executed (see table 1). We have divided the set of gestures into three categories: 1. *Control Gestures*, 2. *Pointing Gestures* and 3. *Social Gestures*. An representative example for each gesture is shown in fig. 7. Category 1 are gestures that are used to control movements and audio output like 'move to the left'. Such sets have already been used in the past to control actuated mechanisms [1]. Category 2 are deictic gestures that are meant to shift Nicole's focus of attention to a certain direction. Pointing gestures have already been used in the past to search and find objects in an image [2]. The last category covers useful social gestures like 'Waving Bye-Bye'.

3 Gesture Recognition Using Dynamic Bayes Nets

Our goal is to design a probabilistic model using the framework of Dynamic Bayesian Networks (DBNs) to anticipate the gesture given the observed features. DBNs offer combinations of the whole family of probabilistic tools like Hidden Markov Models (HMMs), Kalman Filters and Particle Filters and their various modifications. Though, DBNs can be used for all kind of system modeling (e.g. navigation, speech recognition, etc.) they are specially suited for cognitive processes. The process of prediction and update represents an intrinsic implementation of the mental concept of anticipation. In general, modeling offers the opportunity to reach a modest dimensionality of the parameter space that describes the human motion. Bayesian models in particular also maintain an intuitive approach which can also be understood by non-engineers [11]. Furthermore these methods have already proven their usability in gesture recognition [3,4].

A *Bayesian Net* represents the knowledge of an agent about his environment. The definition of states and the (in-)dependencies among each other provides the ground for probabilistic reasoning. If probabilistic reasoning over time is needed the concept of Dynamic Bayesian Nets (DBNs) can be applied. DBNs are often interpreted as a generalization of Hidden Markov Models (HMMS) and Kalman filter networks. The latter also is sometimes referred as Linear Dynamic Systems (LDSs). In designing a DBN to solve a particular task one needs to address the following three issues, preferably in the stated sequence: 1. Topology and conditional probabilities of the network. 2. Method of inference and 3. Learning the parameters.

3.1 Hidden Markov Model (HMM Framework)

A Hidden Markov Model (HMM) is a DBN with a single discrete state variable X_t and a single discrete evidence variable E_t in each slice as shown in fig. 1. With more than one discrete state variable per slice one can combine all the state variables to a single state variable whose numbers are possible tuples of values of the of the individual state variables. The basic properties of the HMM framework:

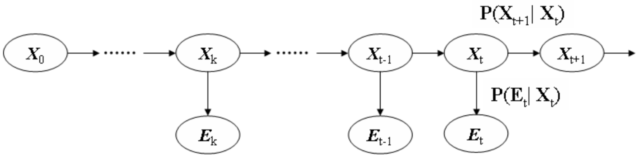

Fig. 1. Typical HMM-Topology

- The time slices are represented by t
- The same subset of variables is observable in each time slice.
- $\mathbf{X}_t$: Set of unobservable state variables at time t
- $\mathbf{E}_t$: Set of observable evidence variables at time t
- $\mathbf{e}_t$: Observation at time t
- Fixed finite interval labeled by integers State sequence starts at $t = 0$
- Evidence starts arriving at $t = 1$
- $\mathbf{X}_{a:b}$ denotes the sequence of the set of variables $\mathbf{X}$ from slice a to b
- The system is modeled by a First-order Markov Process

Give these assumption we can state the complete joint distribution:

$$\mathbf{P}(X_0, X_1, ..., X_t, E_1, ..., E_t) = \mathbf{P}(X_0) \prod_{i=1}^{t} \mathbf{P}(X_i|X_{i-1})\mathbf{P}(E_i|X_i) \tag{1}$$

Equation 1 is the central to our probabilistic reasoning because any probabilistic query can be answered from the full joint distribution. Section 3.3 will present this *inference* more deeply. To express this more elegantly and to implement the basic algorithms we will now adopt the use of matrix notations, taking advantage of the restricted structure of HMMs.

Let the state variable X_t have values denoted by integers $1, ...S$, where S is the number of possible states. The transition model (see fig. 1) $\mathrm{P}(X_{t+1}|X_t)$ becomes a $S \times S$ matrix T, where

$$\mathbf{T}_{ij} = P(X_t = j|X_{t-1} = i)$$

That is T_{ij} is the probability of a transition from state i to j.

The sensor model describes how the evidence variable (sensor) are affected by the actual state. It can be expressed as a diagonal matrix $\mathbf{O}_t$ given by the diagonal entries $P(e_t|X_t = i)$ where e_t reflects the *known* value of the evidence variable E_t.

The primary question concerning the design of the topology is the number of states. As we mentioned earlier a suitable model is that of gesture strokes. A simple tree state HMM with state transitions to itself was suggested by [3] to recognize a pointing gesture.

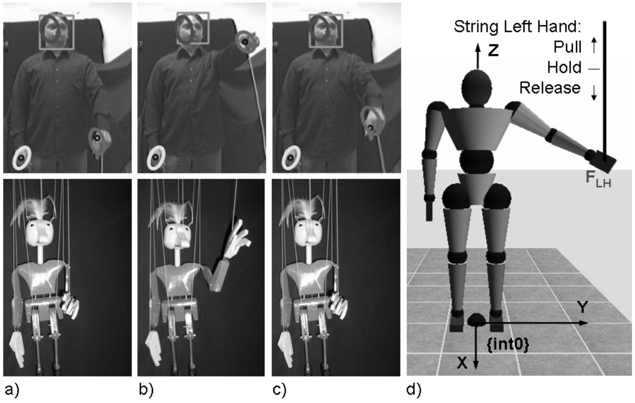

Fig. 2. Gesture phases: a) Pre-Stroke. b) Stroke. c) Post-Stroke. d) Marionette gesture produced by operating a string.

3.2 Framework for the Marionette System

Trying to model all neural and muscular events that makes a human perform a particular gesture still remains computationally expensive. Marionette systems resemble the concept of human motion by controlling the articulation of rigid parts through pulling and releasing strings. Some examples for the attention puppetry receives from the research community are given in [12]. We use a Hidden Markov Model with hidden states variables $\mathbf{X}_t$ to model the marionette. The dimension of a particular state variable is equal to the number of strings attached to the (virtual-)marionette. In the case of full body motion a model with 10 strings as shown in fig. 2 can be used. In the case of gesture recognition we can simplify the model to have three strings, one for each hand and head. Furthermore we have specified the values to be discrete and having the values Pull, Hold or Release.

3.3 Inference and Decoding

Inference in temporal models covers filtering, prediction, smoothing and finding the most likely explanation. The basic task for any probabilistic inference is to compute the posterior probability distribution for a set of query variables given some values for a set of evidence variables. Any conditional probability can be computed by summing terms from the full joint distribution (see equation 1). The full joint distribution specifies the probability of every atomic event.

The forward algorithm can be used for filtering, that is computing the posterior distribution over the current state, given all evidence to date. The process of recursive estimation is to compute the result for $t+1$ from the new evidence $\mathbf{e}_{t+1}$, given the result of filtering up to time t:

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t}) \tag{2}$$

Using the matrix notation and expressing the filtered estimate $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ as a message $\mathbf{f}_{1:t}$ propagated forward along the sequence we can write the forward equation as:

$$\mathbf{f}_{1:t+1} = \alpha \mathbf{O}_{t+1} \mathbf{T}^{\top} \mathbf{f}_{1:t} \tag{3}$$

Finding the most likely explanation, sometimes also called decoding, is the task of, given a sequence of observations, finding the sequence of states that has most likely generated those observations. The solution can be formulated in a recursive manner shown in equation 4

$$\begin{aligned} &\max_{\mathbf{x_1}\cdots\mathbf{x_t}} \mathbf{P}(\mathbf{x_1}, \cdots, \mathbf{x_t}, \mathbf{X_{t+1}}|\mathbf{e_{1:t+1}}) \\ &= \alpha \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \max_{\mathbf{x}_t} \left(\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t \max_{\mathbf{x}_1\cdots\mathbf{x}_{t-1}} P(\mathbf{x}_1, \cdots, \mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{e}_{1:t}) \right) \end{aligned} \tag{4}$$

Equation 4 is identical to equation 2 and 3 except that the message $\mathbf{f}_{1:t}$ is replaced by the message:

$$\mathbf{m}_{1:t} = \max_{\mathbf{x}_1\cdots\mathbf{x}_{t-1}} P(\mathbf{x}_1, \cdots, \mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{e}_{1:t})$$

and the summation over $\mathbf{x}_t$ is replaced by a maximization over $\mathbf{x}_t$.

Thinking about each sequence as a path through a graph whose nodes are possible states at each time step. The equation expresses the recursive relationship between the most likely path to each state $\mathbf{x}_{t+1}$ and the most likely path to each state $\mathbf{x}_t$. The algorithm is know as the *Viterbi algorithm.*

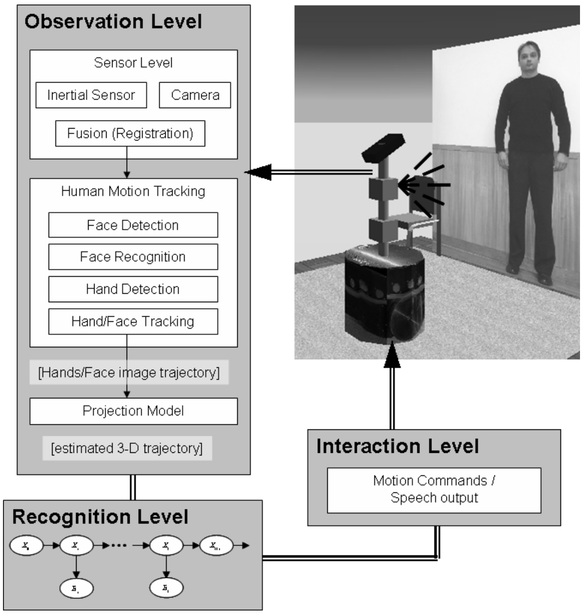

Fig. 3. Architecture of the GP-System

4 The Gesture Perception (GP)-System

In [12] the complete system architecture of the guide robot "Nicole" was presented as well as the architecture of the GP-System. To give a brief overview we will divide the system in three levels of gesture perception (see fig. 3). The processing starts inside the *Observation Level* with the visual-inertial sensor dealing with image capture and inertial data registration. The image data is used by the *Human (Motion) Tracking* module to perform face detection, face recognition, skin-color detection and object tracking. The *Projection Model* module registers the 2D Image trajectory of hands and face in a 3D gesture plane. The features extracted in the *Observation Level* will be used by the Recognition Level. The module will recognize a gesture from the known vocabulary through the transformation of the observed features to marionette states. The following *Interaction Level* will initiate actions like speech output or motion commands according to the recognized gesture.

5 Feature Extraction – The Human Tracking Module

The *Human (Motion) Tracking Module* has been described in [12]. In brief it takes the images from the *Visual-Inertial Sensor* and creates three image trajectories from the head and both hands. As shown in fig. 3 the module contains of four major parts. The process starts with the detection if any human is present in the scene. We use a face detection module based on haar-like features as described in [13], [14] and [15]. If a face is detected the systems checks if the person belongs to the group of people (godfathers) from which Nicole will accept commands. This second part is dealing with face recognition and based on eigen-objects and PCA as described in [16] and [17]. If the persons is identified as a "godfather" then in the third part the skin color detection and the tracking of the hands will be activated. For the skin detection and segmentation we use the CAMshift algorithm presented in [18]. To deal with hands and head occlusion we predict the positions and velocities based on a Kalman-filter [19]. Figure 4 shows the Human Motion Tracking Interface and the resulting trajectories performing a pointing gesture.

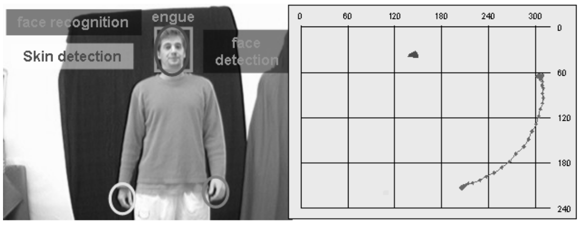

Fig. 4. Human Motion Tracking: a) Interface, b) Image trajectories

6 Feature Registration – The Gesture Plane Projection

Taking the features directly from the image will produce trajectories that are influenced by both the camera's parameters and its orientation in space. The necessity to tilt the camera for adjusting to the height of a person and not being restricted to leveled ground is the reason to register the extracted features in a frame of reference aligned with the gravity. We start by defining the reference frame $\{\mathcal{WI}\}$ at the point of intersection of the vertical body plane π_{vert}, the (mid)sagittal plane π_{sag} and the ground plane π_{grd} shown in Fig. 5 c). Here, we assume that the ground plane π_{grd} and the horizontal plane π_{horiz} are identical. In general we define π_{horiz} as always vertical to the gravity vector $\mathbf{g}$ the ground plane might have a normal vector different from $\mathbf{g}$.

Any generic 3-D feature point $\mathbf{F} = [X\ Y\ Z]^\top$ and its corresponding projection $\mathbf{p} = [u\ v]^\top$ on an image-plane can be mathematically related through the projection matrix $\mathbf{A}$ using projective geometry and the concept of homogeneous coordinates $\mathbf{p} = \mathbf{AF}$. Matrix $\mathbf{A}$ can be expressed by parameters of the projective finite camera model, as stated in [20].

$$\mathbf{A} = \mathbf{C}\left[{}^{\{\mathcal{C}\}}\mathbf{R}_{\{\mathcal{WI}\}}\ {}^{\{\mathcal{C}\}}\overrightarrow{\mathbf{t}}_{\{\mathcal{WI}\}}\right] \tag{5}$$

Where $\mathbf{C}$ is the camera's calibration matrix, more frequently known as the intrinsic parameters matrix, while the camera's extrinsic parameters are represented by the rotation orthogonal matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ that relates the chosen $\{\mathcal{WI}\}$ to the camera frame.

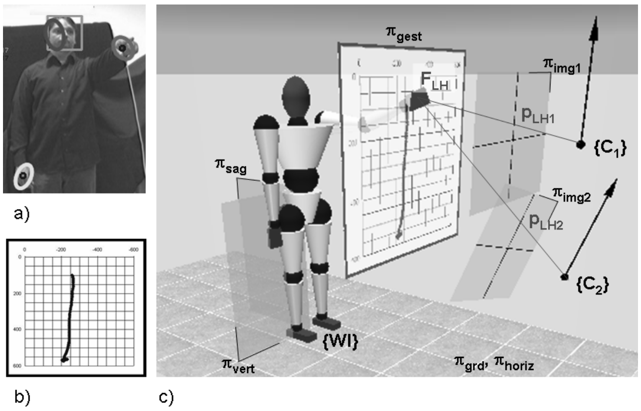

Fig. 5. a) Pointing gesture b) Gesture trajectory c) Different camera perspectives and the Gesture plane

Forming a 3×3 matrix by separating the fourth column of matrix $\mathbf{A}_{3\times4}$ it is easy to show that using the row vectors $\mathbf{a}_1, \mathbf{a}_2$ and $\mathbf{a}_3$ present us with a solution called the projection or projecting line, represented by equation (6) [21].

$$\overrightarrow{\mathbf{n}} = (\mathbf{a}_1 - u\mathbf{a}_3) \times (\mathbf{a}_2 - v\mathbf{a}_3) \tag{6}$$

This relation indicates that all 3-D points on the projecting line correspond to the same projection point on the image-plane. To establish an unique correspondence between the 3D point and its projection on the image-plane we restrict the locus of the 3-D points to lie on a plane $\pi_{gest}\tilde{\mathbf{P}} = 0$. We call π_{gest} which is parallel to π_{vert} the gesture plane. One can think of this as if the person would draw all the gestures with his hands on a virtual blackboard (see fig. 5).

6.1 The Visual-Inertial Sensor

Just like humans [22, 23] use the vestibular system to benefit from the fusion of vision and gravity our system will integrate camera and inertial sensory data to solve the problem of perspective distortion. Recent work of Lobo and Dias present the successful integration and calibration of visual and inertial data [24] and the detection of vertical features [25]. When the system is not accelerating, gravity provides a vertical reference for the camera system frame of reference given by

$$\hat{\mathbf{n}} = \frac{\mathbf{a}}{\|\mathbf{a}\|'} \tag{7}$$

where $\mathbf{a}$ is the sensed acceleration, in this case the reactive (upward) force to gravity.

The publicly available Matlab toolbox for calibration of camera and inertial sensor data [26] provides us with the possibility to undistort our image trajectory and project it on the vertical plane π_{vert}. Figure 6 shows the result of a camera

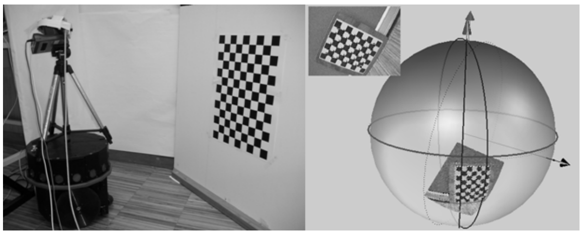

Fig. 6. Results of the calibration process: a) Calibration Set-up, b) Calibration Result (Unit Sphere projection)

calibration process. The calibration target is a checkerboard placed static in the scene while the camera-inertial system takes data from different perspectives. After successful camera calibration we are able to correct the distortion of the trajectory using the gravity normal.

7 Results and Discussion

Figure 7 shows the tracking results from the *Human Motion Tracking Interface* for nine gestures from the three categories. For each gesture it shows a representative frame from the image sequence, the image trajectory and its registration in the gesture plane the 3D trajectory was taken simultaneously with a magnetic tracker (miniBird). This article presented a framework towards a human-robot interaction based on gesture recognition. We presented a Hidden-Markov-Model based framework for gesture recognition that incorporates a novel concept, namely the marionette model. We showed experimental results for feature extraction from our Human Tracking Module and feature registration using visual-inertial senor data.

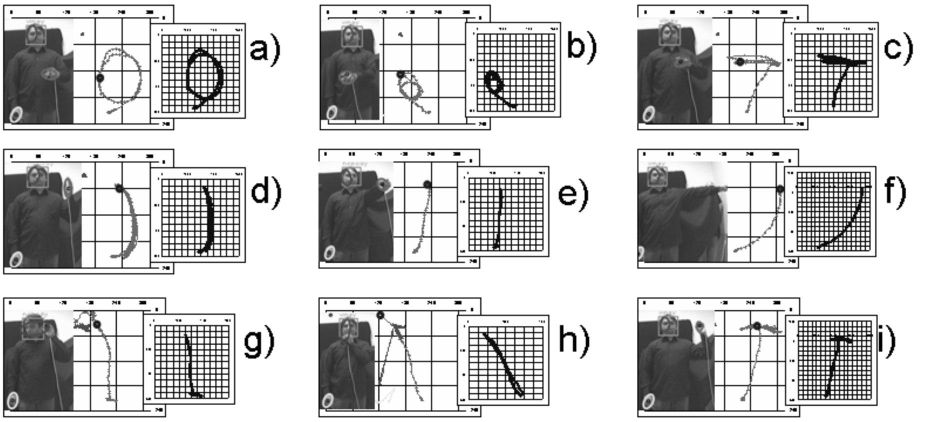

Fig. 7. (a) to (d) Command Gestures. (e) to (f) Pointing Gestures. (g) to (i) Social Gestures.

8 Conclusions

The future work will be concerned with extension of the Dynamic Bayesian Network by incorporating a Kalman filter. During a learning phase the system will estimate the parameters of the DBN from an observed image sequence. The probability of a recognized person will be included as another evidence for the recognition process. Furthermore the want to analyze the "expression" of a gesture (e.g. "a sudden movement without strength and particular direction"). We will use studies on human motion known as *Labananalysis* which defines entities called *Effort parameters* to measure the expressiveness and a notational system called *Labanotation*. Again a bayesian approach will be used for implementation. Finally, we want to develop the capability to learn following a *Learning by Imitation* approach.

Acknowledgements

The authors would like to thank Enguerran Boissier for his contributions to the Human Tracking Module. This work is partially supported by FCT-Fundação para a Ciência e a Tecnologia Grant #12956/2003 to J. Rett.

References

1. Cohen, C.J., Conway, L., Koditschek, D.: Dynamical system representation, generation, and recognition of basic oscillatory motion gestures. In: International Conference on Automatic Face- and Gesture-Recognition. (1996)
2. Kahn, R.E., Swain, M.J., Prokopowicz, P.N., Firby, R.J.: Gesture recognition using the perseus architecture. In: IEEE International Conference on Computer Vision and Pattern Recognition. (1996)
3. Starner, T.: Visual recognition of american sign language using hidden markov models. Master's thesis, MIT (1995)
4. Pavlovic, V.I.: Dynamic Bayesian Networks for Information Fusion with Applications to Human-Computer Interfaces. PhD thesis, Graduate College of the University of Illinois (1999)
5. Meltzoff, A.N., Moore, M.K.: Resolving the debate about early imitation. The Blackwell reader in developmental psychology, Oxford (1999) 151–155
6. Nourbakhsh, I., Kunz, C., Willeke, T.: The mobot museum robot installations: A five year experiment. In: IROS 2003. (2003)
7. Burgard, W., Cremers, A.B., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: Experiences with an interactive museum tour-guide robot. Artificial Intelligence **114** (1999) 3–55
8. Siegwart, R., et al.: Robox at expo.02: A large-scale installation of personal robots. Robotics and Autonomous Systems **42 No. 3-4** (2003) 203–222
9. Rossini, N.: The analysis of gesture: Establishing a set of parameters. In: Gesture-based Communication in Human-Computer Interaction, LNAI 2915, Springer Verlag. (2003) 124–131
10. Kettebekov, S., Yeasin, M., Sharma, R.: Prosody based co-analysis for continuous recognition of coverbal gestures. In: International Conference on Multimodal Interfaces (ICMI'02), Pittsburgh, USA (2002) 161–166

11. Loeb, G.E.: Learning from the spinal cord. Journal of Physiology **533.1** (2001) 111–117
12. Rett, J., Dias, J.: Visual based human motion analysis: Mapping gestures using a puppet model. In: Proceedings of EPIA 05, Lecture Notes in AI, Springer Verlag, Berlin. (2005)
13. Viola, P., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: IEEE International Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) 511
14. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: IEEE International Conference on Image Processing. Volume 1. (2002) 900–903
15. Barreto, J., Menezes, P., Dias, J.: Human-robot interaction based on haar-like features and eigenfaces. In: IEEE International Conference on Robotics and Automation. (2004)
16. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: IEEE International Conference on Computer Vision and Pattern Recognition. (1991) 586–591
17. Menezes, P., Barreto, J., Dias, J.: Face tracking based on haar-like features and eigenfaces. In: IFAC/EURON Symposium on Intelligent Autonomous Vehicles. (2004)
18. Bradski, G.R.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal (1998) 15
19. Kalman, R.E.: A new approach to linear filtering and prediction problems. Trans. ASME—J.Basic Eng. **82** (1960) 35–45
20. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
21. Ferreira, J., Dias, J.: A 3d scanner – three-dimensional recovery from reflected images. In: Proc. Controlo 2000 Conf. on Automatic Control, University of Minho, Portugal (2000) 690–695 Student Forum.
22. Carpenter, H.: Movement of the eyes. Volume 2nd ed. London Pion Limited, London (1988)
23. Soechting, J.F., Flanders, M.: Moving in three-dimensional space: Frames of reference, vectors, and coordinate systems. Annual Review of Neuroscience **15** (1992) 167–191
24. Lobo, J., Dias, J.: Inertial sensed ego-motion for 3d vision. Journal of Robotic Systems **21** (2004) 3–12
25. Lobo, J., Dias, J.: Vision and inertial sensor cooperation using gravity as a vertical reference. IEEE Trans. on PAMI **25** (2003) 1597–1608
26. Lobo, J.: Inervis toolbox.
(`http://www.deec.uc.pt/~jlobo/InerVis_WebIndex/InerVis_Toolbox.html`)