In Proceedings of 8th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Daejeon, Korea, 2008.

Recognizing Complex, Parameterized Gestures from Monocular Image Sequences

Tobias Axenbeck [1], Maren Bennewitz [1], Sven Behnke [2], and Wolfram Burgard [1]

[1] *Institute for Computer Science, University of Freiburg, Germany*

[2] *Institute for Computer Science, University of Bonn, Germany*

***Abstract*— Robotic assistants designed to coexist and communicate with humans in the real world should be able to interact with them in an intuitive way. This requires that the robots are able to recognize typical gestures performed by humans such as head shaking/nodding, hand waving, or pointing. In this paper, we present a system that is able to spot and recognize complex, parameterized gestures from monocular image sequences. To represent people, we locate their faces and hands using trained classifiers and track them over time. We use few, expressive features extracted out of this compact representation as input to hidden Markov models (HMMs). First, we segment gestures into distinct phases and train HMMs for each phase separately. Then, we construct composed HMMs, which consist of the individual phase-HMMs. Once a specific phase is recognized, we estimate the parameter of the current gesture, e.g., the target of a pointing gesture. As we demonstrate in the experiments, our method is able to robustly locate and track hands, despite of the fact that they can take a large number of substantially different shapes. Based on this, our system is able to reliably spot and recognize a variety of complex, parameterized gestures.**

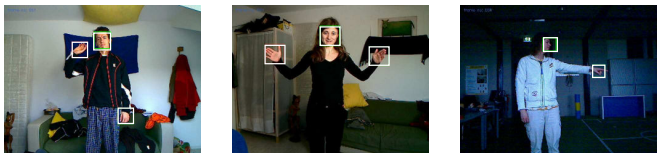
I. Introduction

Whenever robots are designed to operate in human-populated environments, they must be able to interact with them in an intuitive way. Our humanoid robot (see Fig. 1) is able to generate a variety of human-like arm and head gestures that support its speech [1]. At former public demonstrations we asked people who interacted with the robot to fill out questionnaires about their impression of the interaction capabilities of the robot. We discovered that several people were confused by the asymmetry between action generation and perception since the robot's visual perception of people was limited to head position and size. To reduce this asymmetry and to enrich its multimodal interaction capabilities, it is necessary that the robot also recognizes gestures performed by humans. This requires robust and accurate tracking of human body parts as well as the ability to spot and recognize typical gestures in order to infer non-verbal signals of attention and intention.

In this paper, we present a system that is able to spot and recognize complex gestures from monocular image sequences. We consider typical gestures performed with head and arms, such as head shaking/nodding or hand waving as well as parameterized gestures. Fig. 2 shows some examples.

We represent humans using their heads and hands. For locating faces and hands in the images, we use the object detection framework proposed by Viola and Jones [2] to train reliable and fast classifiers. We use an adaptive skin color model (which is initially based on the detected face) and constrain the search to skin-colored regions to speed-up and to increase the robustness of the hand detection process.

Fig. 1. Our humanoid robot interacts with people using multiple modalities such as speech, facial expressions, eye-gaze, and gestures.



(a) (b) (c)

Fig. 2. Snapshots of typical gestures: (a) waving, (b) indicating the size of an object (parameterized), and (c) pointing to an object (parameterized). The bounding boxes highlight detected faces and hands.

We segment complex arm gestures into their three natural phases and train hidden Markov models (HMMs) for the individual phases. We then construct HMMs composed of the individual phase-HMMs for one- and two-handed gestures as well as for head gestures.

Our approach proceeds in three stages. First, we locate faces and hands in the images and update a probabilistic belief which tracks them over time. Second, we extract features from this compact representation of humans. Finally, these features are used as input to the HMMs. Our system recognizes a variety of complex gestures and allows for the estimation of parameters for general gestures once a specific phase is recognized. In contrast to that, existing techniques for parameter estimation of gestures either concentrate on pointing gestures only (e.g., [3], [4], [5]) or rely on the assumption that the whole gesture can be observed [6].

The contribution of our work is a robust and fast gesture recognition method that relies on monocular image sequences (no stereo). In contrast to previous approaches relying on monocular data (e.g., [7], [8], [9]), our system works under realistic settings such as varying and difficult lighting

conditions, multiple people, and cluttered background. On a notebook computer, we achieve a frame rate of 20 fps and are able to spot gestures as well as to recognize them, i.e., our system distinguishes between previously learned gestures and irrelevant or unconscious movements.

This paper is organized as follows. The next section reviews related work. Section III describes training and application of the hand classifiers. Section IV details our technique to track people. Section V explains the features selection process and how HMMs are trained and used for gesture recognition. Finally, Section VI presents the experimental results.

II. Related Work

Several approaches to visual gesture or activity recognition have already been presented. Yamato *et al.* [10] apply discrete HMMs to recognize tennis strokes from monocular images. As features, they use simply the number of pixels corresponding to the human pose and apply a vector-quantization. Rigoll *et al.* [8] proposed to recognize gestures from low-resolution grey-scale images using continuous HMMs. To compute a 7-dimensional feature vector, they describe the region corresponding to the moving body parts using statistics such as image moments. Montero and Sucar [9] use a ceiling-mounted camera and apply a back-projection using a given color histogram to locate a hand on a desk. Given features based on the 2D trajectory of the hand, the authors apply a HMM to recognize typical office movements such as writing, using the mouse, etc. Li *et al.* [7] presented an approach to recognizing manipulative actions via a hierarchical HMM. In addition to the hand trajectory, objects in the vicinity of the hand serve as observations in order to be able to distinguish between the different activities. It should be noted that in contrast to all these approaches, our gesture recognition system works under realistic settings such as varying and difficult lighting conditions and cluttered background.

The work presented by Nickel and Stiefelhagen [4] concentrates on pointing gesture recognition using a stereo camera system. They use heuristics to locate heads and hands by combining color and range information. The authors model the pointing gestures using three individual phase-HMMs. As soon as the hold phase is recognized, the target is estimated using the 3D positions of head and hand. In contrast to this system, ours is not restricted to one single gesture. Instead, we are able to recognize a variety of gestures. Furthermore, Nickel and Stiefelhagen apply a time-consuming analysis to estimate the end of a gesture phase which can more efficiently done using the Viterbi algorithm. Just *et al.* [11] consider the problem of recognizing one- and two-handed gestures given 3D trajectories. The authors do not tackle the problem of recognizing parametric gestures. An interesting extension to HMMs was introduced by Wilson and Bobick [6]. To extract information carried by parameterized gestures, the corresponding parameter is explicitly integrated into a parametric HMM (PHMM). Using PHMMs, the parameter can be estimated with a high accuracy given an entire sequence. However, it is unclear how PHMMs perform in case only part of the gesture can be observed. Our approach is able to estimate the parameter of the gesture as soon as the corresponding hold phase is recognized.

Martin *et al.* [3] use a combination of Garbor filters and neural function approximators to estimate the target of pointing gestures from monocular images. They only analyze the static part of a gesture. Irie *et al.* [5] proposed to control devices in an intelligent room equipped with two cameras by hand and finger gestures using heuristics to determine the hand motion.

The problem of whole body gesture recognition from depth images or using a motion capture system has been addressed by several researchers (e.g., [12], [13]). The proposed methods need a high-dimensional feature vector consisting of joint angles as input to HMMs. Thus, a high number of training sequences is needed. The same holds for approaches which mainly focus on the reproduction of whole body, respectively, arm movements using learned HMMs (e.g.,[14], [15], [16]).

Finally, we would like to review related work regarding hand detection, which is an inherently difficult task. Kolsch and Turk [17] also applied the object detection framework introduced by Viola and Jones [2] to detect different hand postures. They concentrate on few distinctive hand shapes which are frequency-analyzed for good class separation ability. Similarly, Chen *et al.* [18] trained classifiers for four different hand postures. Ong and Bowden [19] proposed training of a two-layer classifier tree for hand shape detection where a database of hand images is clustered into sets of similar hands according to a distance metric based on shape context. In contrast to these methods, our system is able to detect and track hands with a large number of substantially different shapes and to furthermore determine whether a hand is a left or right one. Our system works reliably even under difficult background and lighting conditions.

III. Hand Detection

A. Training Classifiers

Hands can take the highly various shapes as they are articulated with more than 20 degrees of freedom and they appear arbitrarily rotated, in-plane as well as out-of-plane. For training robust hand classifiers, we apply the object detection approach proposed by Viola and Jones [2] which is based on AdaBoost. We use Haar-like features [20] to construct the classifiers, i.e., each feature is computed based on the sum of pixel values in rectangular regions in grey-scale images. The idea is to use information about the relative darkness between different regions.

B. Training Data

We train two kinds of classifiers: a *generic* hand classifier that detects hands and rejects non-hands and a *specific* hand classifier that is able to discriminate right hands from left ones, given there is a hand. The size of the positive samples is chosen so as to include contextual information that might be useful for the classification (see Fig. 3).

Since in the detection process, we only consider skin-colored regions (see next subsection), only these regions are considered during acquisition of negative samples. The

Fig. 3. Positive example patches for training hand classifiers.

negative training examples of the *specific* classifier consist only of images patches containing the contralateral hand as negative samples. In this way, AdaBoost is forced to select features which best discriminate left and right hands.

C. Constraining the Search and Applying the Classifiers

To support the hand detection process, we perform a preprocessing step and incorporate information derived from the face detection result. We use the classifiers provided by Intel's OpenCV library [21] to detect frontal and profile faces. By means of color analysis of the respective image patch containing the face, the skin-color of the person can be estimated. This information can then be used to identify candidate hand regions in the image. The advantage of this approach is two-fold. First, we can constrain the search for hands to regions containing the same color as the face and thus speed-up the search. Second, objects with similar structure but with different color are excluded and thus the false detection rate is reduced. Once the hands are detected, we start updating the skin color model with information from the hand regions.

Our hand detection system proceeds in two stages. First, the generic hand detector is applied. In case it succeeds, the right hand classifier is applied twice, once in the original image and once in the flipped one. Four cases are possible:

1) No success in both images: Return generic hand.
2) Success in original image: Return right hand.
3) Success in flipped image: Return left hand.
4) Success in both images: Return left/right depending on the output scores of the classifiers.

IV. Representation and Tracking of Humans

We represent humans using their heads and hands. We maintain a probabilistic belief about the existence of people and the positions of their faces and hands over time. This way, our system improves robustness since it can deal with false detections and is not restricted to a single person.

We proceed as follows. We first run the face detector in the images. Before we can update the Kalman filters tracking the faces, we have to solve the data association problem, i.e., we must determine which observation corresponds to which already tracked face and which observation belongs to a new face. We use a distance-based cost function and apply the Hungarian method [22] to determine the optimal mapping from current observations to faces given this cost function.

To deal with false classifications, we maintain for each face the probability that it really exists, i.e., that a person is there. We follow the approach presented by Bennewitz *et al.* [23] to update this probability by applying a recursive update scheme. This update scheme determines the probability of the existence of a person given a sequence of positive and/or negative observations (face detections) assigned to it

$$P(f \mid z_{1:t}) = \left[1 + \frac{1 - P(f \mid z_t)}{P(f \mid z_t)} \cdot \frac{P(f)}{1 - P(f)} \cdot \frac{1 - P(f \mid z_{1:t-1})}{P(f \mid z_{1:t-1})}\right]^{-1} . \quad (1)$$

Here, f denotes the existence of a face, z_t is the observation (face detected/not detected) at time t, and $z_{1:t}$ refers to the observation sequence up to time t. We experimentally determined values for $P(f \mid z = det)$ that a face exists if it is detected in the image and $P(f \mid z = \neg det)$ that a face exists if it is not detected. If the probability of a face drops below a certain threshold, the person is deleted from the belief.

Additionally, we track the 3D head pose of people. We use an appearance-based approach [24] which locates distinctive facial features. The positions of the features within the face bounding box serve as input to a neural network which computes the three Euler angles of rotation around the neck.

For the hands, we maintain two kinds of probabilities. First, we also compute the existence probability and, second, we compute for each hand the probability that it is a left or right hand given the results of the specific classifier. The information about the laterality of hands is important to keep track of them in case hands are crossing each other.

Again, we first have to solve the data association problem. The costs of an assignment for an observed and a tracked hand depend on the laterality costs c_{lat}, which considers the probability that a hand is a left or right one, in combination with the distance between the position of the detected hand and the predicted position of the already tracked hand using a weighted sum. c_{lat} is defined as

$$c_{lat} = [P(h^r|h, z_t = h^r) \cdot P_{bel}(h^r|h, z_{1:t-1}) + P(h^l|h, z_t = h^l) \cdot P_{bel}(h^l|h, z_{1:t-1})]^{-1} . \quad (2)$$

Here, h^r and h^l denote a right/left hand and the observations z_t are either left hand or right hand. After determining the assignment, the existence probability $P(h \mid z_{1:t})$ and the probability that a hand is a right or left one $P(h^{r/l} \mid h, z_{1:t})$ are updated using the recursive formula described above.

The final step is to assign hands to people. To do so, we first partition the set of tracked hands into left and right hands according to their most likely probability. Then, we assign the sets of left and right hands to the tracked people individually. To avoid that hands change their assignment to another person in case people come close to each other, we consider the history of assignments in the cost function. For each hand in the belief, we maintain a histogram in which each bin i stores how often the hand has been assigned to person i. In this way, we maintain an assignment of a particular hand to a particular person over time. The costs c_{ij} of the assignment of the tracked hand i to person j are then computed as follows

$$c_{ij} = \frac{1}{\mathit{hist}(h^i, q_j)} + c_{dist}(h^i, q_j). \quad (3)$$

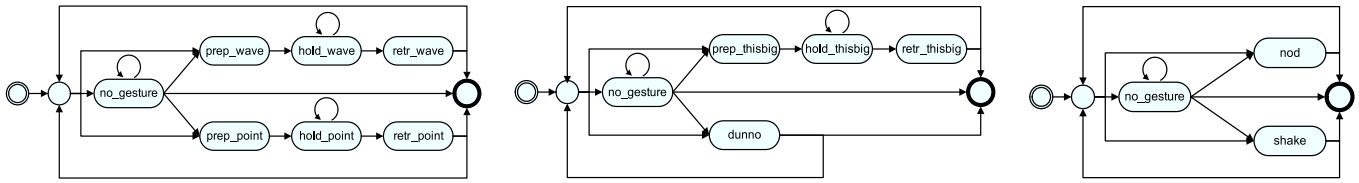


Fig. 4. Composed HMMs consisting of the individual phase-HMMs. The first two for one- and two-handed gestures, and the right one for head gestures. The start and end states of the HMMs as well as the branching points are non-emitting.

Here, $hist(h^i, q_j)$ denotes the normalized bin value for person j of hand i. $c_{dist}(h^i, q_j)$ is proportional to the distance to person j or is maximum if the hand is too far away.

V. Recognizing Complex, Parameterized Gestures

We currently consider six different types of gestures:

1) *Waving*: One-handed gesture.
2) *Pointing*: Parametric, one-handed gesture.
3) *Thisbig*: This parametric, two-handed gesture is carried out to indicate the size of an object.
4) *Dunno*: This two-handed gesture is used to express ignorance (informal short for *don't know*).
5) *Head shaking*.
6) *Head nodding*.

A. Gesture Modeling

To model the complex arm gestures *Waving*, *Pointing*, and *Thisbig*, we use three phases: the preparation phase which is an initial movement before the main gesture, the hold phase which characterizes the gesture, and the retraction phase in which the hand moves back to a resting position. Our main motivation behind this segmentation is that once the hold phase is recognized, the parameters of *Pointing* and *Thisbig* can be estimated. The less complex gestures *Dunno* and *Head shaking/nodding* are modeled monolithically. We train individual HMMs for each phase of a gesture separately, i.e., we train 12 HMMs for the gestures/gesture phases.

We use continuous left-right HMMs with 3-5 (non-skip) states in addition to the non-emitting start and end states. The actual number of states depends on the average length of the gesture phases. As output distribution we use a mixture of two Gaussians. We apply Viterbi training and the Baum-Welch algorithm to estimate for a HMM λ the transition probabilities a_{ij}^{λ} between states i and j and the observation probabilities $b_j^{\lambda}(o)$ for a state j given an observation o.

To be able to identify movements not corresponding to any learned gesture, we train an additional model. Here, we follow the approach presented by Yang *et al.* [13] and build a HMM by copying all states from all trained models and arrange them in a fully connected HMM with smoothed output probabilities. The transition probabilities in this *no_gesture* model are defined as $a_{ij} = (1 - a_{jj}^{\lambda}) \cdot \frac{1}{|\#states-1|}$ for $i \neq j$ and $a_{ij} = a_{jj}^{\lambda}$ else. Here, a_{jj}^{λ} is the self-transition probability of state j which originally belongs to the HMM λ.

B. Gesture Recognition via Composed HMMs

The gesture phases appear in a specific order which has to be considered during recognition. Fig. 4 illustrates the HMM topology for one- and two-handed gestures as well as for head gestures. As indicated by the arrow, the hold phase can occur several times. The transition probability from the end state of a phase-HMM to the start state of the same HMM is set equal to the transition probability of going to the next phase-HMM in the network.

To identify the most likely gesture given such a composed HMM, we apply the Viterbi algorithm [25]. The Viterbi algorithm computes the state sequence with maximum likelihood given an observation sequence $O_{1:T} = o_1, \ldots, o_T$. For the HMM λ, the likelihood of the best state sequence of length t ending in state j is recursively defined as

$$\delta_t(j) \quad = \quad \max_{1 \leq i \leq N^{\lambda}} \delta_{t-1}(i) a_{ij}^{\lambda} b_j^{\lambda}(o_t), \delta_1(j) = \pi_j^{\lambda} b_j^{\lambda}(o_1). \quad (4)$$

Here, a^{λ} and b^{λ} are the parameters of λ, N^{λ} is the number of states, and π_j^{λ} specifies the initial state distribution. The algorithm terminates with the computation of the most likely path x_T^* (which is found via backtracking) and its likelihood P^*

$$P^* \quad = \quad \max_{1 \leq i \leq N^{\lambda}} \delta_T(i). \quad (5)$$

In theory, it would be possible to model all gestures in one single HMM. However, to reduce the amount of necessary training data and to improve recognition accuracy, we use separate HMMs and extract individual input features. To distinguish between one- and two-handed gestures, we consider the two-handed HMM only when the HMMs for the right and left hand report the same most-likely gesture. This heuristics is applicable since all our two-handed gestures are symmetric.

C. Input Features

As input to the HMMs, we use few, expressive features extracted out of the trajectories of head and hands. First, we transform the position of the hands into coordinates relative to the head position and normalize the coordinates with respect to the size of the face bounding box. For one-handed gestures, we use polar coordinates in the image with the head as origin and the velocity. Accordingly, the feature vector $\mathbf{f}_{one}$ is defined as $\mathbf{f}_{one} = (r, \phi, v)$. Here, r is the distance of the hand to the head, ϕ is the angle, and v is the velocity.

Since the two-handed gestures we consider are symmetric, we measure the difference in x/y-direction of their left and right hand coordinates $(x_t^{l/r}, y_t^{l/r})$ at time t in the features $d_x = |x_t^l| - |x_t^r|$ and $d_y = y_t^l - y_t^r$. Furthermore, we record the sum of the y-coordinates of the hands in the feature $y^{lr} = y_t^l + y_t^r$ and consider the change of the hand

coordinates in x-direction

$$\Delta x^l x^r = |x_t^l| - |x_{t-1}^l| + |x_t^r| - |x_{t-1}^r|. \quad (6)$$

As a final feature, we consider the velocities of the hands $v^{lr} = v_t^l + v_t^r$. Thus, the feature vector $\mathbf{f}_{two}$ is defined as

$$\mathbf{f}_{two} = (d_x, d_y, y^{lr}, \Delta x^l x^r, v^{lr}). \quad (7)$$

The head gestures nodding and shaking are described by a feature vector $\mathbf{f}_{head}$ which consists of the three Euler angles of rotation roll, pitch, and yaw as well as their velocities

$$\mathbf{f}_{head} = (\theta^r, \theta^p, \theta^y, v^{\theta_r}, v^{\theta_p}, v^{\theta_y}). \quad (8)$$

D. Estimating Parameters of Gestures

Currently, we consider two parameterized gestures: *Thisbig* and *Pointing*. The corresponding parameters are estimated during the hold phase of the respective gesture. For *Thisbig*, the estimation is done straightforwardly using the tangent function and a learned mapping to estimate the distance of the person to the camera given the size of the bounding box of the face.

For the estimation of pointing targets, we use the three rotation angles of the head pose. We assume that people are looking to the object of interest they want to draw the attention to and that the head pose coincides with the gaze direction. Furthermore, we assume the 3D positions of potential pointing targets to be known. First, we estimate the 3D position of the head using the above mentioned mapping from bounding box size to distance. Starting from that position, we construct a straight line in direction of the roll, pitch, and yaw angle of the head pose. Finally, we determine the object which has the closest distance to that line.

VI. Experimental Results

We performed a series of experiments in order to evaluate our approach. To collect training data, we asked five different people to perform gestures standing frontal in a distance of 1-2.5m to the camera. We chose two different locations, different lighting conditions, and different backgrounds (see Fig. 2). We recorded and processed the videos with a rate of 20fps and used a resolution of 640×480 pixel. We had a database consisting of 75 samples per gesture which we manually labeled, i.e., we marked the start and the end of each gesture as well as the beginning and end of the hold phase. For training and testing our hand detection system, we labeled the hands in a set of images. We used 5000 image patches containing hands for training the classifiers.

A. Hand Detection

First, we evaluated the performance of our hand detection system. In this experiment, we evaluated solely the ability of detecting hands using our system described in Sec. III (without the tracker and the belief). The results are summarized in Tab. I. The distance to the true (hand-labeled) position is measured in pixels. A detection rate of over 80% is sufficient as input to our belief to robustly track the hands. Also the ability to distinguish between left and right hands with a detection rate of almost 90% and a false alarm rate of 5% is highly satisfactory.

TABLE I

Performance of our hand detectors.

	Detection rate	False alarm rate	Avg. dist.
Generic detector	81.25	0.10	2.89
Specific detector	89.50	5.56	-

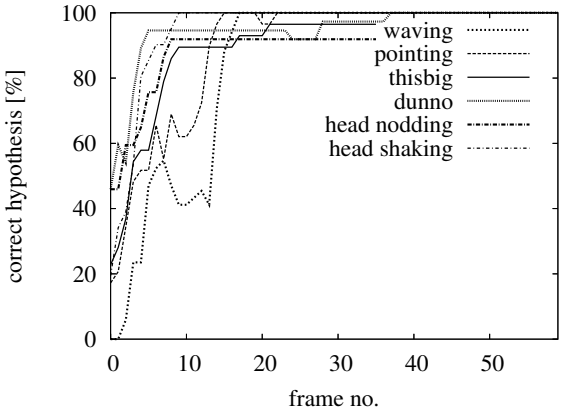


Fig. 5. Number of frames after which the most likely hypothesis is the correct gesture.

B. Gesture Recognition

The following experiment is designed to evaluate the performance of our gesture recognition system. We computed the Viterbi path in the composed HMMs at each time step and counted how often the most likely hypothesis corresponded to the true gesture. Fig. 5 shows the results for all six gestures. The gestures could be reliably recognized after processing only few frames. Nodding seems to be most difficult to recognize because sometimes people barely move their head. Rarely, it happens that *Thisbig* is classified as *Dunno*.

To better evaluate the ability of our HMMs to distinguish arm gestures, we performed experiments in which we computed for a given observation sequence the Viterbi path and its likelihood for all individual gesture HMMs consisting of the corresponding phase-HMMs (i.e., we did not use the fully composed HMMs here). We then computed the joint probability $P(g_l, g_r, g_{two})$ of the gesture g_l of the left, the gesture g_r of the right hand, and the two-handed gesture g_{two}.

Fig. 6 plots the evolution of the probabilities of the gestures over time for two sequences in which a person waved with the left hand and (top image) and performed a *Dunno* gesture (bottom image). In the beginning, the person was not performing any meaningful gesture and, thus, the *no_gesture* model had the highest probability. Afterwards, the probability of the correct gesture increased.

C. Parameter Estimation

Finally, we asked people to point to predefined targets. We positioned eight different targets within a range of 1.5m to the camera at different heights. The hold phase of all 66 pointing

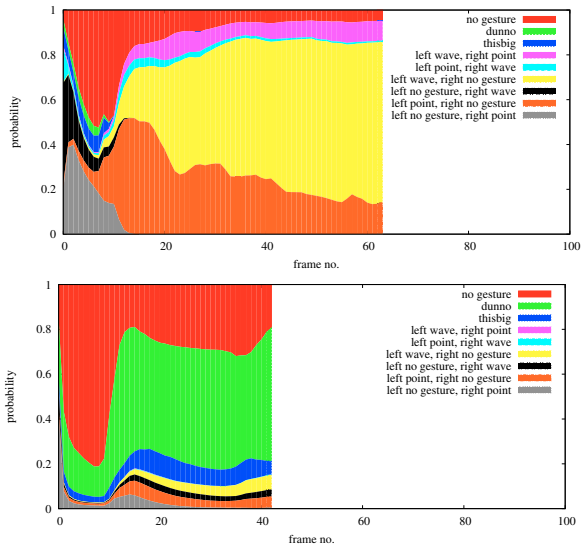


Fig. 6. Evolution of the probabilities of the gestures over time for waving with the left hand (top image) and *Dunno* (bottom image).

gestures was identified and the correct target was estimated in 80% of all cases.

Second, we asked people to indicate the size of objects. We told them to indicate the sizes 25cm, 50cm, 100cm, and 150cm and estimated the parameter in the hold phase. We performed 32 experiments and counted the nearest neighbor class of each estimate. Our system was able to determine the correct class in 94% of all cases.

VII. Conclusions

We presented an approach to robustly recognize typical gestures performed with the head and the arms such as nodding or pointing from monocular vision. We use trained classifiers in combination with an adaptive skin color model to reliably detect faces and hands. We segment complex gestures into three phases and train HMMs for each phase separately given few, expressive features. We then construct HMMs composed of the individual phase-HMMs. Whenever a certain phase is recognized, we can estimate the parameter of a gesture, e.g., the target of a pointing gesture.

Our approach has been implemented and evaluated on a humanoid robot. As the experiments demonstrate, our system works under realistic settings and is able to reliably spot and recognize gestures. Gesture recognition is not restricted to people whose gestures were collected during the training phase. However, it is assumed that the people perform the individual gestures sufficiently similar to those observed during training, which is the case for the class of gestures we consider.

Acknowledgment

This work was supported by the DFG, grant BE 2556/2-2 and by the BMBF project DESIRE.

References

[1] M. Bennewitz, F. Faber, D. Joho, and S. Behnke, "Fritz – A humanoid communication robot," in *Proc. of the IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[3] C. Martin, F.-F. Steege, and H.-M. Gross, "Estimation of pointing poses for visual instructing mobile robots under real-world conditions," in *Proc. of the 3rd European Conf. on Mobile Robots (ECMR)*, 2007.

[4] K. Nickel and R. Stiefelhagen, "Detection and tracking of 3D-pointing gestures for human-robot-interaction," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2004.

[5] K. Irie, N. Wakamura, and K. Umeda, "Construction of an intelligent room based on gesture recognition," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[6] A. Wilson and A. Bobick, "Parametric hidden Markov models for gesture recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[7] Z. Li, N. Hofemann, J. Fritsch, and G. Sagerer, "Hierarchical modeling and recognition of manipulative gesture," in *Proc. of the Workshop on Modeling People and Human Interaction at the IEEE Int. Conf. on Computer Vision*, 2005.

[8] G. Rigoll, A. Kosmala, and S. Eickeler, "High performance real-time gesture recognition using hidden Markov models," in *Proc. of the Int. Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, 1998.

[9] J. A. Montero and L. E. Sucar, "Feature selection for visual gesture recognition using hidden Markov models," in *Proc. of the Fifth Mexican Int. Conf. in Computer Science*, 2004.

[10] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1992.

[11] A. Just, O. Bernier, and S. Marcel, "HMM and IOHMM for the recognition of mono- and bi-manual 3D hand gestures," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2004.

[12] S. Lee, "Automatic gesture recognition for intelligent human-robot interaction," in *Proc. of the 7th Int. Conf. on Automatic Face and Gesture Recognition (FG)*, 2006.

[13] H.-D. Yang, A.-Y. Park, and S.-W. Lee, "Robust spotting of key gestures from whole body motion sequence," in *Proc. of the 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*, 2006.

[14] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2006.

[15] S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[16] T. Inamura, Y. Nakamura, I. Toshima, and H. Tanie, "Embodied symbol emergence based on mimesis theory," *The Int. Journal of Robotics Research (IJRR)*, vol. 23, no. 4-5, pp. 363–377, 2004.

[17] M. Kolsch and M. Turk, "Robust hand detection," in *Proc. of the Sixth IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*, 2004.

[18] Q. Chen, N. Georganas, and E. Petriu, "Real-time vision-based hand gesture recognition using haar-like features," in *Proc. of the IEEE Conf. on Instrumentation and Measurement Technology*, 2007.

[19] E.-J. Ong and R. Bowden, "A boosted classifier tree for hand shape detection," in *Proc. of the Sixth IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG)*, 2004.

[20] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2002.

[21] Intel, "Open source computer vision library," http://www.intel.com/technology/computing/opencv/, 2007.

[22] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1, pp. 83–97, 1955.

[23] M. Bennewitz, F. Faber, D. Joho, S. Schreiber, and S. Behnke, "Integrating vision and speech for conversations with multiple persons," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[24] T. Vatahska, M. Bennewitz, and S. Behnke, "Feature-based head pose estimation from images," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.

[25] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.