Recognition of Head Gestures in Multimodal Human-Machine Dialogs

Frank Wallhoff, Alexander Bannat, Jürgen Gast, Tobias Rehrl, Stefan Schwärzler, Gerhard Rigoll
Human-Machine Communication, Department of Electrical Engineering and Information Technologies
{wallhoff,bannat,gast,rehrl,schwaerzler,rigoll}@mmk.ei.tum.de
Matthias Wimmer, Christoph Mayer, Bernd Radig
Image Understanding and Knowledge-Based Systems, Department of Informatics
{wimmerm,mayerc,radig}@in.tum.de
All authors contributed equally

***Abstract*— This paper presents a system for human-machine communication that is able to participate in a simple dialog. Spoken language and head gestures are integrated to pass information between the interacting partners. Head gestures are extracted by image interpretation algorithms basing on machine learning techniques. Our experimental evaluation proofs the capability of the system to recognize head gestures from camera images in a robust way. The system works in real-time and will be presented publicly.**

I. Introduction

Computers quickly solve mathematical problems and memorize an enormous extent of information, but human-computer interaction still lacks intuition. A large amount of time is required for humans to adapt to operating a specific machine. Instead, we aim at granting machines the ability to adapt to typical human behavior. In real world environments, technical systems have to act without manipulation of specifically trained persons. In human-machine communication, the different steps of interaction have to be performed autonomously and robustly. Therefore, interfaces need to infer information from gaze, facial expressions, head gestures, speech and other human communication channels. All processing steps need to be fully automated.

Our system relies on two communication channels to establish natural, human-like communication. Speech recognition and face tracking are utilized to collect information about the human's response to machine actions via head gestures such as head shaking and nodding, see Figure 1. The information gained on these channels is fused to infer high-level information and generate appropriate robot reactions.

A. Related Work

Referring to the survey of Pantic et al. [1], the computational task of facial expression recognition is usually subdivided into three subordinate challenges: face detection, feature extraction, and facial expression classification. After the position and shape of the face in the image are detected in the first step, descriptive features are extracted in the second step. In the third step, high-level information from these features is derived by a classifier. We apply this approach to recognize head gestures.

Models rely on a priori knowledge to represent the image content via a small number of model parameters. This

Fig. 1. The camera is mounted flexible in order to track the human's face. Head gestures are estimated from the acquired images.

representation of the image content facilitates and accelerates the subsequent interpretation task. Cootes et al. [2] introduce modeling shapes with Active Contours which use a statistics-based approach to represent human faces. Further enhancements extended the idea and provided shape models with texture information [3]. Therefore, also skin color, eye color, wrinkles etc. are considered. However, both models rely on the structure of the face image rather than the structure of the real-world face. Therefore, information such as position or orientation in three-dimensional space is not explicitly considered but has to be calculated from the model parameters. Since this mapping is again not provided by the model, it is error-prone and renders them difficult for extracting such information.

Recent research considers modeling faces in 3D space [4], [5]. In contrast to two-dimensional models these models directly provide information about position and orientation of the face. In addition, often texture or shape is also taken into consideration. However, this renders the model fitting

From: 1st Intern. Cotesys Workshop 2008, Technische Universität München

and tracking step slow and therefore we do not consider this additional information in our system.

B. System Overview

Our system realizes a simple dialogue where information is passed from the human to the machine via speech and head gestures. A dialogue manager keeps track of the ongoing communication to estimate when human user or machine response is expected by the dialogue partners. Furthermore, it determines when information is lost and keeps knowledge about the human's position in space via face tracking. The system recognizes simple verbal commands such as "go to the fridge" and asks for confirmation of the given task by short sentences such as "should I go to fridge?". Confirmation is recognized via speech or head gestures, i.e. nodding and head shaking. The inferred redundancy of this system architecture is used to ensure higher robustness. Even in the presence of noisy surroundings retrieving information from head gestures provides accuracy. The other way around, in crowded environments the user face is likely to be occluded and therefore the information gained via speech is more reliable. However, multimodel fusion is a highly complex topic and therefore we refer to [6]

C. Organization of the Paper

The reminder of this paper is structured as follows: Section II introduces our model fitting approach and the extraction of features descriptive for head gesture estimation. Section III provides an overview about Hidden Markov Models and how we utilize them to classify head gestures. Section IV evaluates our approach with respect to the model fitting accuracy and the recognition rate of the Hidden Markov Model. Section IV-B presents conclusions and future work.

II. Model-based Feature Extraction

This section details our model fitting approach and the extraction of descriptive features. Robust model fitting forms the basis of the extraction of accurate feature values ameliorating the classification results.

A. Model Fitting

This section introduces model-based analysis of face images. Models impose knowledge about the object of interest and reduce the large amount of image data to a small number of expressive model parameters. Model fitting is the computational challenge of finding the model configuration describing the content of the image best [7]. In general, model fitting consists of two components: the fitting algorithm and the objective function. The *objective function* $f(I,p)$ yields a comparable value that determines how accurately a parameterized model p fits to an image I. The *fitting algorithm* searches for the model parameters p that optimize the objective function. Yet, this paper shall not elaborate on them but we refer to [7] for a recent overview and categorization.

The objective function, which we consider the core component of the model fitting process, is often designed manually using the designer's domain knowledge and intuition. Afterwards, its appropriateness is subjectively determined by inspecting its result on example images and example model parameters. If the result is not satisfactory the function is tuned or redesigned from scratch [8], [9]. Since the design-inspect loop is iteratively executed, manually designing the objective function is highly time-consuming, see Figure 2 left.

B. Learning Objective Functions for Face Model Fitting

In contrast, we utilize the approach of [10] to learn the objective function rather than designing it manually, see Figure 2 right. This approach is based on general properties of ideal objective functions. The key idea behind the approach is that if the function used to generate training data is ideal, the function learned from the data will also be approximately ideal. Furthermore, we provide a large number of image features. The learning algorithm is able to consider this vast amount of features and the resulting objective function allows model fitting with both good runtime performance and great accuracy.

1) Ideal Objective Functions: Ideally, the objective function for fitting a model point has two properties. First, its global minimum corresponds to the correct position of the model point. Second, it has no further local minima. Equation 1 depicts an *ideal* objective function $f_n^\star$. It simply computes the Euclidean distance between the correct location $\hat{x}_n^\star$ of the n^{th} model point and a location u in the image I. Note that the vector of correct model points $\hat{x}^\star$ must be specified manually.

The function $f_n^\star$ already shows ideal characteristics. Unfortunately, this function is not able to be used for previously unseen images, because it must know of the correct locations of the model points $\hat{x}^\star$, which have to be manually specified beforehand. However, our approach uses $f_n^\star$ to generate training data for learning an additional objective function f_n^ℓ that does not require knowledge of $\hat{x}^\star$.

$$f_n^\star(I,u) = |u - \hat{x}_n^\star| \tag{1}$$

2) Applying Machine Learning: We annotate a set of images with the correct model points $\hat{x}^\star$. For each $\hat{x}_n^\star$, the ideal objective function returns the minimum $f_n^\star(I,\hat{x}_n^\star) = 0$ by definition. Further coordinate-to-value correspondences are automatically acquired by varying $\hat{x}_n^\star$ along the perpendicular and recording the value returned by the ideal objective function in the second step.

Finally, the calculation rules of the objective function are learned with tree-based regression [11]. The advantage of this machine learning approach is that it only selects relevant features, and therefore, the values of far fewer image features need to be computed during the process of fitting the model.

As demonstrated in [10], this approach is comparable to state-of-the-art approaches. This approach does not require expert knowledge and it is domain-independently applicable.

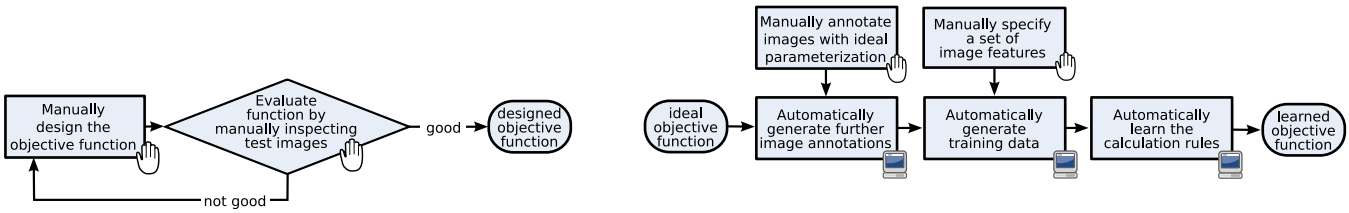

Fig. 2. left: the design approach. right: the learn approach.

As a result, this approach yields more robust and accurate objective functions, which greatly facilitate the task of the associated fitting algorithms. Accurately estimated model parameters in turn are required to infer correct high-level information, such as head gestures.

C. Feature Selection

We utilize a rigid, three-dimensional model of human faces in our system because it inherently considers position and orientation of the face in space. Our experimental evaluation indicates that head gestures are robustly recognizable from this information. The small amount of model parameters guarantees a short calculation time which in term provides real-time capability.

Five model parameters are considered to train a classifier for the recognition of head gestures. The data vector d_i extracted from a single image I_i is composed of the in-plane transition of the face and the three rotation angles (pitch, yaw and roll). However, we do not utilize the absolute values of the five parameters but temporal parameter changes. Due to several advantages we apply Hidden Markov Models - presented in Section III-A - for classification.

III. Head Gesture Estimation

In section we present our approach for recognizing two different head gestures (nodding, shaking). In addition a third state models the absence of head movements. For simplicity we will refer to all three observations as "'head gestures"' in the reminder of this paper. In the following we will present a short introduction to Hidden Markov Models(HMMs). For further information about HMMs we refer to [12].

A. Hidden Markov Models

We utilize Continuous Hidden Markov Models (HMMs) [12], [13] to derive the head gesture from the extracted feature values. A Hidden Markov Model λ relies on J internal emitting states q_j, a state transition matrix A including the non emitting start and end state (q_0 and $q_{(J+1)}$) and the (continuous) production probability vector $\vec{b} = [b_1 \dots b_J]^{\mathrm{T}}$ to calculate the probability that a sequence of feature vectors is produced by a particular head gesture.

The elements $a_{q_j q_{(j+1)}}$ of the matrix A represent the transition probabilities from state q_j to state $q_{(j+1)}$ (1st order Markov Model). The elements b_j in a certain state j for a D-dimensional observation $\vec{x}_j$ are given by a multivariate Gaussian distribution consisting of a mean value vector $\vec{\mu}_j$ and a covariance matrix Σ_j.

$$b_j(\vec{x}_j, \vec{\mu}_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_j|}} e^{-\frac{1}{2}(\vec{x}_j - \vec{\mu}_j)^T \Sigma_j^{-1} (\vec{x}_j - \vec{\mu}_j)} \quad (2)$$

They describe the probability of a given observation $\vec{x}_j$ in a state q_j.

During the training phase the unknown parameters in A and $\vec{b}$ are calculated. For this purpose the well-known Baum-Welch-Estimation procedure [12] can is applied.

The Hidden Markov Model parameters are computed by the following maximum-likelihood decision:

$$\lambda = \underset{\lambda}{\operatorname{argmax}}\ P(X|\lambda) \quad (3)$$

Where X represents a vector of observations and λ a model parameterization.

B. Training HMMs for Head Gesture Estimation

In total, fourteen different persons constitute the model for classifying head gestures. We record two sequences per person and head gesture (nodding, shaking, neutral). The model is tracked through these short image sequences consisting of roughly $n = 26$ frames I_i, $1 \leq i \leq n$ and the model parameters are exploited to train a classifier. Per training image sequence we create one set of data vectors $(d_1, \dots, d_{20})$ of fixed size. Each of these sets forms one observation to train the HMM as described in section III-A. Note, that therefore the HMM determines the head gesture for a sequence of images rather than for a single image. In total we present the HMM $14 \times 3 \times 2 = 74$ observations. The only parameter given manually is the number of states J. We train different HMMs to correctly determine this parameter. Inspection of the training errors shows that the best parameterization is $J = 5$, see Section IV-B for further details.

C. Real-time Face Gesture Recognition

The dialogue manager estimates when human response via head gestures is expected and focuses the camera on the face under investigation. The model is fitted to the first image I_0 obtained from the human interaction partner and tracked in the subsequent images. The HMM is presented the extracted data vector d_i which is assembled as described in Section II-C. Note, that the HMM relies not only on the data vector currently presented but also on all precedent data vectors. The result for I_i is created by the HMM regarding the vectors d_0 to d_i. Hence, we get a sequence of head gesture estimations over time. When a particular head gesture

appears in the sequence with a predetermined frequency the dialogue manager is notified.

IV. Experimental Evaluation

This section inspects the accuracy of our model fitting approach and the classification accuracy of the HMM. The fitting accuracy is important because it has impact on the tracking of the model which in turn influences the quality of the extracted feature values. The recognition rate of the HMM is important because it affects the machine reactions and therefore the dialogue smoothness. To show the reliability of the system, we apply 6 fold cross validation, see Table I.

A. Model Fitting Accuracy

Our evaluation iteratively executes the process of model fitting and investigates the fitting error, see also [14]. We measure the cumulative error distribution of the fitted models with respect to manually specified model parameters. Figure 3 illustrates that each execution of the fitting step improves the model parameters. However, more than 12 iterations do not improve the model parameters significantly any more.

Models with a high distance from the correct fit become even more at every iteration. Since the training data did not contain image annotations for these cases the objective function's value is arbitrary. Therefore, the models are displaced in a unpredictable way by the fitting algorithm.

B. Head Gestures

We evaluate the trained HMM with a 6-fold cross validation. We split the data recorded in Section III-B in six non-overlapping parts. Five parts are taken for training and

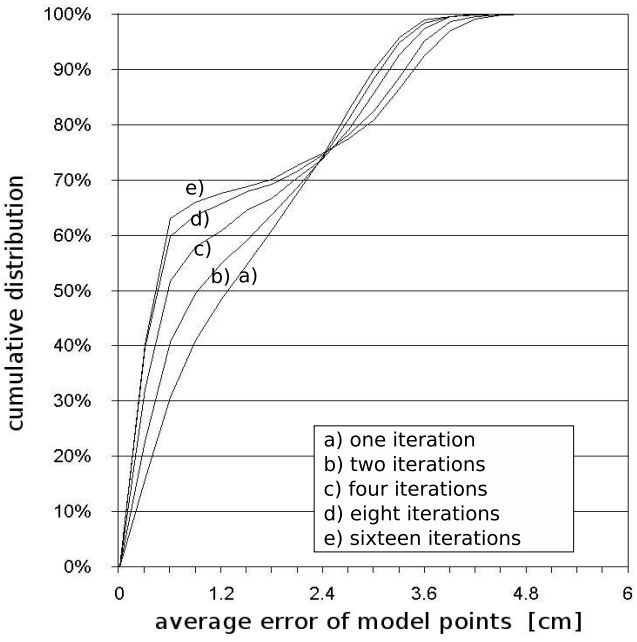

Fig. 3. Iteratively executing the fitting process increases the fitting accuracy.

Classified As	Sequence Label		
	Shaking	Neutral	Nodding
Shaking	**100%**	0%	0%
Neutral	6%	**88%**	6%
Nodding	6%	16%	**78%**
Mean error rate	11.33%		

classification result with three states

Classified As	Sequence Label		
	Shaking	Neutral	Nodding
Shaking	**94%**	5%	0%
Neutral	16%	**77%**	7%
Nodding	0%	22%	**78%**
Mean error rate	17.00%		

classification result with four states

Classified As	Sequence Label		
	Shaking	Neutral	Nodding
Shaking	**100%**	0%	0%
Neutral	11%	**89%**	0%
Nodding	0%	6%	**94%**
Mean error rate	5.67%		

classification result with five states

Classified As	Sequence Label		
	Shaking	Neutral	Nodding
Shaking	**100%**	0%	0%
Neutral	6%	**88%**	6%
Nodding	0%	6%	**94%**
Mean error rate	6.00%		

classification result with six states

Classified As	Sequence Label		
	Shaking	Neutral	Nodding
Shaking	**100%**	0%	0%
Neutral	6%	**94%**	0%
Nodding	6%	6%	**88%**
Mean error rate	6.00%		

classification result with seven states

TABLE I

This table presents recognition rates of our HMM trained with three to seven states. The results are obtained from a 6-fold cross validation.

the remaining sixth part is taken for testing. The process is iterated six times and the average of the resulting accuracy values is inspected. Note, that in contrast to the online execution sequences of a fixed length are presented to the classifier. In addition, we vary the number of states J (three to seven) of the Hidden Markov Model, see Section III-A. Again, 6-fold cross validation is utilized to inspect the recognition accuracy. Table I shows that based on the mean error rate the best results are achieved for $J = 5$. Table I provides an overview of the accuracy values.

sectionConclusions In this paper we present a system that realizes a simple dialogue between a human and a machine. Two different communication channels are regarded: The machine receives simple commands and asks for confirmation via spoken language. Furthermore, head gestures are recognized via model-based image understanding techniques and classification with Hidden Markov Models. The system operates without manual control and all important algorithms base on objective machine learning techniques instead of subjective manual design. Future work focuses on increasing the robustness with respect to real-life scenarios (lighting conditions, multiple points of view, etc.) and integrating facial expressions into the classification process.

V. Acknowledgement

This ongoing work is partly supported by the DFG excellence initiative research cluster *Cognition for Technical Systems CoTeSys*, see `www.cotesys.org` for further details.

References

[1] M. Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, 2000. [Online]. Available: citeseer.ist.psu.edu/pantic00automatic.html

[2] T. F. Cootes and C. J. Taylor, "Active shape models – smart snakes," in *Proceedings of the 3rd British Machine Vision Conference*. Springer Verlag, 1992, pp. 266 – 275.

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *5th European Conference on Computer Vision*, H. Burkhardt and B. Neumann, Eds., vol. 2. Freiburg, Germany: Springer-Verlag, 1998, pp. 484–498.

[4] J. Ahlberg, "Candide-3 – an updated parameterized face," Linköping University, Sweden, Tech. Rep. LiTH-ISY-R-2326, 2001.

[5] V. Blanz, K. Scherbaum, and H. Seidel, "Fitting a morphable model to 3d scans of faces," in *Proceedings of International Conference on Computer Vision*, 2007.

[6] M. Guiliani, M. Kassecker, S. Schwärzler, A. Bannat, J. Gast, F. Wallhoff, C. Mayer, M. Wimmer, C. Wendt, and S. Schmidt, "Mudis - a multimodal dialogue system for human-robot interaction," in *1st International Workshop on Cognition for Technical Systems*, 2008, to appear.

[7] R. Hanek, "Fitting parametric curve models to images using local self-adapting seperation criteria," PhD thesis, Dep of Informatics, Technische Universität München, 2004.

[8] S. Romdhani, "Face image analysis using a multiple feature fitting strategy," Ph.D. dissertation, University of Basel, Computer Science Department, Basel, CH, January 2005.

[9] D. Cristinacce and T. F. Cootes, "Facial feature detection and tracking with automatic template selection," in *7th IEEE International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, April 2006, pp. 429–434.

[10] M. Wimmer, F. Stulp, S. Pietzsch, and B. Radig, "Learning local objective functions for robust face model fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 8, 2008.

[11] R. Quinlan, "Learning with continuous classes," in *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, A. Adams and L. Sterling, Eds., 1992, pp. 343–348. [Online]. Available: http://citeseer.ifi.unizh.ch/quinlan92learning.html

[12] L.R.Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE 77*, 1989.

[13] F. Wallhoff, M. Zobl, and G. Rigoll, "Action segmentation and recognition in meeting room scenarios," in *Proceedings on IEEE International Conference on Image Processing*, 2004.

[14] D. Simon, M. Hebert, and T. Kanade, "Real-time 3-D pose estimation using a high-speed range sensor," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '94)*, vol. 3, May 1994, pp. 2235–2241.