A REAL-TIME DEMONSTRATOR FOR VIDEO-BASED RECOGNITION OF DYNAMIC HEAD GESTURES, USING DISCRETE HIDDEN MARKOV MODELS

Frank Althoff, Gregor McGlaun, Manfred Lang, Gerhard Rigoll

Institute for Human-Machine Communication
Technical University of Munich (TUM)
Arcisstr. 21, 80290 Munich, Germany
email: {althoff, mcglaun, lang, rigoll}@ei.tum.de

Major theme:
Image, Acoustic, Speech and Signal Processing

ABSTRACT

This work describes a powerful demonstrator of a video-based approach for detecting and classifying dynamic head gestures. The head of the user is localized via a combination of color- and shape-based segmentation. For a continuous feature extraction, we use a template matching of the nose bridge in combination with selected features derived from the optical flow. The core classification unit consists of discrete Hidden Markov Models (DHMMs). We extensively tested the system in two different domains (desktop Virtual-Reality and automotive environment). In the current state of development, six different gestures can be classified with an overall recognition rate of 97.3% in the VR, and 95.5% in the automotive environment, respectively. The approach works absolutely independent from the image background and additional gesture types can easily be integrated.

1. INTRODUCTION

The development of user interfaces has become a significant factor in the software design process. Growing functional complexity and mostly restriction to purely tactile interaction required extensive learning periods and adaptation by the user to a high degree, which significantly increased user frustration. To overcome these limitations, various new interaction paradigms have been introduced in the course of time. Multimodal interfaces currently resemble the latest step in this development. They enable the user to freely choose among multiple input devices, provide essential means to resolve recognition errors of individual components, and thus lead to systems that can be worked with easily, effectively, and above all intuitively[1]. Besides speech input, the use of gestures provides an interesting alternative for people with certain disabilities. For example, in an automotive environment, head gestures allow control of the in-car devices without losing eye-focus on the street. Hence head gestures offer a highly effective input alternative, as the hands can still be used to drive the car.

This contribution illustrates a robust algorithm along with a real-time demonstrator for video-based recognition of dynamic head gestures. The implemented module has been extensively tested in two different domains: a desktop oriented Virtual-Reality application (DVA) and the operation of various infotainment applications in an automotive environment (AIA). The long-term goal of the research effort is to use the head gesture module as an integral part of an domain-invariant multimodal system architecture.

1.1. Related work

Many research groups have contributed significant work in the field of video-based head gesture recognition. In a system developed by Morimoto[2], movements in the facial plane are tracked by evaluating the temporal sequence of image rotations. These parameters are processed by a dynamic vector quantization scheme to form the abstract input symbols of a discrete HMM which can differentiate between four head gestures (*yes, no, maybe* and *hello*). Based on the IBM PupilCam technology, Davis[3] proposed a real-time approach for detecting user acknowledgments. Motion parameters are evaluated in a finite state machine which incorporates individual timing parameters. Using optical flow parameters as primary features, Tang[4] applies a neural network to classify ten different head gestures. The approach is quite robust with regard to different background conditions. Tang obtained an average recognition rate of 89.2% on a workstation processing 30 frames per second.

1.2. Gesture vocabulary

Before designing specific algorithms, we analyzed and categorized different types of natural dynamic head movements and determined the set of recognizable gestures. As an important result of a dedicated offline analysis of the video ma-

terial, we found out that the majority of gestures (96.39%) has exclusively been composed of rotational movements[5]. Thus, we exclusively consider head gestures that consist of one or a combination of head rotations.

Since the recognition module is to be implemented in various contexts, we define two sets of possible head gestures. The first set (GS_1) contains six gestures: moving the head *left* and *right* (rotation around the *yaw*-axis), *up* and *down* (rotations around the *pitch*-axis), and bending the head left and right (rotation around the *curl*-axis). Additionally, by combining basic movements, two compound gestures are evaluated head *nodding* and head *shaking*. The vocabulary of the second set (GS_2) is designed to exclusively support user acknowledgment decisions. Thus, we reduced it to the gestures head *nodding* and head *shaking*.

2. COLOR-BASED SEGMENTATION

Based on the excellent overview given in[6], we experimented with various techniques. Since a fundamental requirement of our approach is real-time processing capability, we propose a color-based segmentation approach, because it is rotation- and scale-invariant, and the calculation is very fast. Moreover, this method does not require any kind of initialization, and has proved to be highly robust with regard to arbitrary motion in the background. The individual steps of the segmentation process are visualized by the two sequences shown in figure 1.

Given in the standard size of 382x288 pixels, the input image is in standard RGB color format, with each channel composed of 8 bit (figure 1(a)). To differentiate between skin color and background, the image is converted to the YCbCr color space by the following expression:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Since different skin types mostly differ in the luminance component and not with regard to the hue value, the Y-channel can be neglected in the following. Concerning the $CbCr$-plane, skin colors only cover a small fraction. For each of the color vectors, the probability of belonging to human skin can be estimated. To simplify the color distribution, we use an approximation by the following Gaussian function. For specifying the individual parameters, the mean value $\vec{m}$ was calculated (where $\mathbb{E}$ denotes the expectation value):

$$\vec{m} = \mathbb{E}\{\vec{x_i}\} \text{ with } \vec{x_i} = \begin{pmatrix} Cr \\ Cb \end{pmatrix}$$

and the covariance matrix

$$C = \mathbb{E}\{(\vec{x_i} - \vec{m})(\vec{x_i} - \vec{m})^T\}$$

on the basis of 42 random user skin samples $\vec{x_i}$. To filter out non skin-color areas, the histogram of the $CbCr$ part of the input image is multiplied with the Gaussian distribution calculated by:

$$\mathrm{p}(Cr, Cb) = \exp[-0.5(\vec{x_i} - \vec{m})^T C^{-1}(\vec{x_i} - \vec{m})].$$

The resulting histogram is used to project the color image onto a gray-value image (figure 1(b)), in which each skin color value is represented by a gray-value according to the probability specified by $\mathrm{p}(Cr, Cb)$. Afterwards, this gray-value image is binarized differentiating between potential skin colors and background (figure 1(c)).

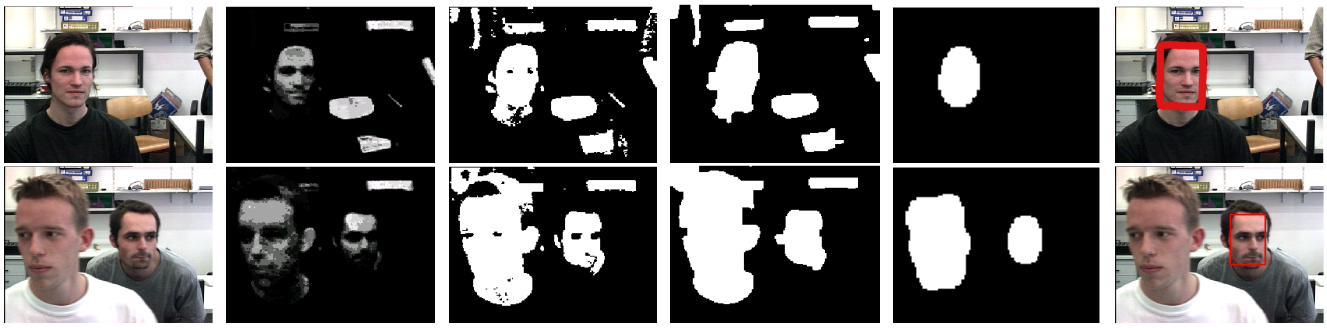
Moreover, we apply a sequence of morphological filters on the binary image. First, a *closing* with a small ellipse eliminates small particles that have occurred due to noise. Then, an *opening* with a medium-sized rectangle tries to cover dark areas like in the eyes. For each blob, potentially occurring leaks will be filled. These leaks can often be found near to the eyes. As they are not skin-colored, they have a negative influence on the correct segmentation of the whole face region. The result of these morphological filters are shown in figure 1(d). Finally, a closing with a longish bigger ellipse removes all areas which do not have the correct size (figure 1(e)). By a bounding box R around the best-fitting ellipse, the position of the potential head candidate is specified (figure 1(f)).

3. TEMPLATE MATCHING

To further improve the quality of the segmentation result, we additionally apply a template matching algorithm. Therefore, a striking, invariable *region of interest* (ROI) in the facial plane has to be identified. A basic requirement for a robust tracking of this ROI should be the independence of special faces. Taking the center of the eyes as ROI results in misclassifications when the user blinks. In this case, the eyes fuse with the rest of the face to one single blob. Moreover, the mouth drops out as a potential ROI, since it changes its form during talking. Therefore, we concentrate on the nose bridge as the key feature. For enlarging the matching criteria, we use a symmetric template including the nose bridge, the area of the eyes, and parts of the eye-brows.

For each of these head candidates, we calculate a measure of how good the template matches the current image region R. This is done by determining the match quality of the template and the input image column by column and row by row. The result of this match depends both on the quality of the template and the special kind of the matching algorithm. We use the standard gray-level correlation

$$c(x, y) = \sum_{(u,v) \in R} t(u, v) b(x + u, y + v),$$



(a) (b) (c) (d) (e) (f)

Fig. 1. Individual steps of the segmentation process: the input image given in standard RGB format with a size of 382x288 pixels (a), skin-color information coded in a gray-value image (b), binarized image due to a certain threshold differentiating between potential areas of skin-color and background (c), result of a sequence of morphological filters to improve the segmentation result (d), final closing with a longish ellipse to identify potential head candidates (e), and marking head regions in the original input image (f).

where the template $t(x, y)$ is defined over $\mathbb{R}$ and $b(x, y)$ denotes the input image. We relate the individual gray-values to the medium gray-value and normalize them by their standard deviation. Using the gray-scale correlation instead of the sum of absolute gray-scale differences, changes in the light conditions can easily be handled.

If the resulting match value is below a certain threshold, the head candidate will not be accepted as a potential position of a head. If more than one candidate exceeds the threshold, the best correlation candidate is taken for further processing. This can be seen in the lower series of images in figure 1, where the right candidate is preferred.

The native segmentation phase is exited, if a head candidate is found. All further calculation steps are done within the region found as a result of the native segmentation process. In case the area gets to small or no blob is found anymore, the search is extended to the complete image. This principle guarantees for an integral robust localization of the head and a fast tracking of the head regions in the image.

4. CONTINUOUS FEATURE EXTRACTION

The tracking module calculates the spacio-temporal movements of head candidates in the image sequences and provides the basic data for the subsequent classification process. We apply a hybrid combination of the Averaged Optical Flow (AOF) and the continuous template matching of the nose bridge. Hence, the template matching is used to estimate the position of the nose bridge in the tracked face region. The optical flow calculates motion vectors of certain areas of interest in subsequent images. The approach tries to find solutions to the known flow equation $\nabla I \cdot \vec{v} + I_t = 0$. Hence, $I = I(x, y, t)$ denotes the luminance, which depends of the local coordinates x, y, and the time t. Moreover, $\vec{v} = (\frac{dx}{dt}; \frac{dy}{dt})^T$ represents the vectored velocity of the head movement. For calculating $\vec{v}$, the standard Lucas-Kanade algorithm[7] is used. For reasons of system performance, we apply this local method instead of techniques operating on the whole image (e.g. the *Horn-Schunk* algorithm). In common implementations, the AOF is usually computed over a rectangle containing the whole head. Yet, we have found out that the bounding box around the head region in itself is sometimes not sufficient for adequate classification results. Concerning rotations of the head in the image plane (yaw- and $pitch$-axis), the resulting bounding box does not change significantly. This especially holds for segmentation results that cover bigger parts of the body. A nod of the user could not be detected, because the total movement is completely enclosed in the primary rectangle. Thus, the result of the segmentation is only used to restrict the search area for the extraction of the features. We apply the AOF technique to detect rotational movements around the nose bridge. For this purpose, the face is separated into two halves each bounded by a rectangle. In our approach, we use the already localized nose bridge as an approximation for an element on the vertical symmetry axis of the face. With respect to this point, the position of two 40x40 pixel squares are calculated. For each half of the face, the AOF is separately calculated. As an effect of the implementation, we get two competing outputs for the AOF for each side of the face. Concerning the gesture sets GS_1 and GS_2, the AOF of the left and the right side of the face make for almost identical results. Thus the two redundant features are supposed to confirm each other. The nose bridge is used as an origin of the local coordinate system of the rectangles bounding the face halves. In combination with the relative movement of the nose bridge, we are able to distinguish between horizontal movements of the user and head shaking itself. With horizontal movements, the AOF is zero, as the

offset generated by the movement is compensated by the offset of the relative coordinate system.

As a third feature, we use the difference vector of the nose bridge between two frames. Based on the template matching outlined in section 3, we can establish a local Cartesian coordinate system with its zero point in the lower left corner of the rectangle R (see figure 1(f)). Let j be an integer indexing each frame F of a video sequence. For the j-th frame F_j, let the center of the template be denoted by $\vec{c}_j$. Referring to the previous frame F_{j-1}, we can express the motion of the nose bridge by the difference vector $\vec{d}_j = \vec{c}_j - \vec{c}_{j-1}$. This vector is transformed into a polar representation using the absolute value $\|\vec{d}_j\|_2$ and the phase φ. Let $d_{j,1}$ be the x-component and $d_{j,2}$ denote the y-component of $\vec{d}_j$, then the phase can be calculated via

$$\varphi_j = \begin{cases} 0 & \text{, if } d_{j,1} = 0 \\ \arctan(\frac{d_{j,2}}{d_{j,1}}) & \text{else} \end{cases}$$

Using φ_j and $\|\vec{d}_j\|_2$, we can specify the direction and the speed of the head motion for each frame. These three features provide the basis for the classification process that is described in the subsequent section.

5. CLASSIFICATION

Modeling head gestures, a fundamental aspect is tolerance of small divergences regarding the temporal run and the duration. In the field of stochastic approaches, Hidden Markov Models very well cope with molding on time variant patterns. In addition, they show a robust behavior on small breaks during a gesture, which are likely to appear when the head moves through the inflection point within a gesture. In the current implementation, we use Discrete Hidden Markov Models (DHMMs) composed of five states for the classification of head gestures. As mentioned in[8], DHMMs in general take more parameters, but the calculation is easier in the recognition process. The generation of the discrete symbols that are fed into the DHMMs can be split up into two steps. First the optical flow and the arithmetic mean is computed over the regions which are in close vicinity to the nose bridge. Then, both vectors as well as the speed vector of the nose bridge, whose position has been determined by a template matching, are discretized to integer values between 0 and 5. Hence the symbol 0 represents *no movement*. The symbols 1 to 4 are generated by applying the mapping scheme sketched in figure 2.

If $\theta = 0°$, the two dimensional space is symmetrically partitioned into four motion sectors (from the viewpoint of the camera). In different test runs, we varied the partition types by applying different values for θ. Additional explanations can be found in section 7.

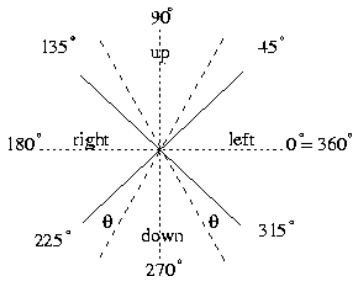


Fig. 2. Mapping scheme for φ (viewpoint: camera)

As the discrete pixel movements range between zero and three pixels, a more subtle distinction is not suggestive. The symbols s_1, s_2, and s_3 are canonically coded into a value representing the final symbol, using the known formula $s_1 + 5s_2 + 5^2 s_3$. The classifier evaluates the symbol sequences and puts out a probability vector for each DHMM. Finally, the result is returned in terms of an n-best list.

6. SPOTTING

In the current state of development, the recognition process is automatically triggered, when any kind of head movement is detected. For this purpose, the absolute value of difference vector of the nose bridge (see section 4) is evaluated. If two head gestures directly follow one another, a number of five or more idle frames must be detected between these two gestures to separate them. Otherwise, the recognition process continues, which consequently leads to wrong results. The improvement of the segmentation between single gestures is part of current work. Hence we are about to implement a technique proposed in [9], which applies an improved normalized Viterbi algorithm for a continuous observation of the HMM output scores. This approach allows for an integrated spotting and classification at a time.

7. EVALUATION

The recognition module system has been implemented on an Intel Pentium IV machine with 512KByte cache and 1 GByte memory under the Linux operating system (Kernel 2.4.20). We have evaluated both the time performance and the recognition rates in the various domains. A single input frame is composed of an RGB image with 288x384 pixels.

7.1. Test environment and procedure

The approach has been evaluated in two different application domains. One test series was run under optimized conditions in the computer-vision laboratory of the institute. We shielded the environment from glares of the sun, and

G↓ H→	Up	Down	Left	Right	Shake	Nod
Up	**96.3**	0.4	0.1	0.1	0.6	2.5
Down	0.5	**96.1**	0.2	0.2	0.2	2.8
Left	0.2	0.1	**98.0**	0.8	0.8	0.1
Right	0.1	0.1	0.5	**96.6**	1.9	0.8
Shake	0.1	0.3	1.4	1.0	**97.0**	0.2
Nod	1.3	2.5	0.1	0.3	0.2	**95.6**

Table 1. Recognition rates with regard to gesture set GS_1 in the VR (G: actual head gesture, H: HMM gesture modeling)

G↓ H→	Up	Down	Left	Right	Shake	Nod
Up	**95.2**	1.2	0.3	0.3	0.2	2.8
Down	1.9	**94.5**	0.4	0.3	0.1	2.8
Left	0.2	0.6	**95.0**	2.2	1.1	0.9
Right	0.5	0.7	1.2	**94.2**	3.1	0.3
Shake	0.3	0.3	2.2	2.5	**93.9**	0.8
Nod	1.5	3.0	0.5	0.2	0.6	**94.2**

Table 2. Recognition rates with regard to gesture set GS_1 in the automotive environment

G↓ H→	Shake	Nod
Shake	**98.2**	2.2
Nod	1.8	**97.8**

G↓ H→	Shake	Nod
Shake	**96.8**	3.9
Nod	3.2	**96.1**

Table 3. Recognition rates with regard to gesture set GS_2 in the VR (left) and the automotive environment (right)

used a flicker-free light. The scene background was native consisting of different objects. During the data collection, the test subjects sat on a chair in front of a camera (distance 60cm). They had to interact in different desktop-VR scenarios, using head gestures of both test sets GS_1 and GS_2.

In the second test series, we focused to evaluate head gestures under preferably realistic conditions. The test domain was an automotive environment in a driving simulator. Driving the test car in the simulation, the trial participants had to perform head gesture interaction with different in-car infotainment devices. Yet, we did not consider any influences of artificial vibrancies or forces implicated by bumps, curves, or braking. The camera, which had the same sample rate as in the VR-desktop environment, was positioned on the dash board over the steering wheel with an approximate distance of 45 cm. To simulate alternating light conditions, we shaded the laboratory, and used a set of spotlights.

In both test environments, a set of gestures with moving objects in the background were evaluated. The video data was captured. In case subsequent gestures have been made, they were not manually segmented in order to analyze system behavior with respect to the prototypical spotting.

7.2. Results

We used a total of 153 video sequences of ten different test subjects, and 120 sequences of eight subjects in the automotive environment. The training corpus for the DHMMs has consisted of 32 selected symbol sequences. It contained gestures of four persons of different skin colors and one person wearing glasses. We have used the Baum-Welch method for the training. The data for test and training has been strictly disjoint.

Tables 1 to 3 show the recognition results. In both domains, a strong affinity between direction related gestures (*up*, *down*, *nod*, and *left*, *right*, *shake*, respectively) could be observed. This effect has been aggravated, when the gestures are made very quickly. Then, the resulting symbol sequence corresponding to the gesture has contained too few elements, so no good match for an DHMM has been found.

Particularly, the good recognition rates for the reduced set G_2 have been due to the training corpus containing a great variety of gestures of different durations. Considering all types of gestures, the performance is better in the VR-desktop scenarios than in the automotive environment, where head movements often were less distinct. This particularly happened in cases the subjects did not have a frontal view into the camera. In 20 evaluated test sequences, the head of the test participant was initially rotated by approximately $40°$. In some cases, blinking or even moving the pupils have had a negative impact on the computation of the AOF, which has consequently lead to misclassifications. Moreover, we have observed that the AOF is more likely to be error-prone to bad light conditions than the template matching algorithm and the feature extracted from it.

8. DEMONSTRATION PLATFORM

To evaluate the individual parameters and to demonstrate the usefulness of our head recognition system, we designed and implemented a dedicated graphical user interface (GUI) as the primary front-end to the control several basic functionalities. The GUI mainly consists of four areas (see figure 3): the main window and three output windows for showing different stages of preprocessing and classification.

The main window(1) serves for adjusting the individual parameters, coordinating specific input sources and selecting the classification approach. Concerning the initial default settings for the preprocessing stage, the search region is set to $\delta x = 3$px, $\delta y = 3$px (px: pixels) with regard to $1/8$ of the image size (384x288 px), the skin color threshold (SC) is 2. Moreover, the initial values for the semi-axes $(a; b)$ of the ellipsoidal structuring elements are: $(22; 16)$ for the head structuring element (MO_1), $(2; 2)$ for the noise structuring element (MO_2), and $(5; 5)$ for the eye structuring element (MO_3). The confidence for accepting a head candidate is 0.7 by default. Concerning the tracking algo-

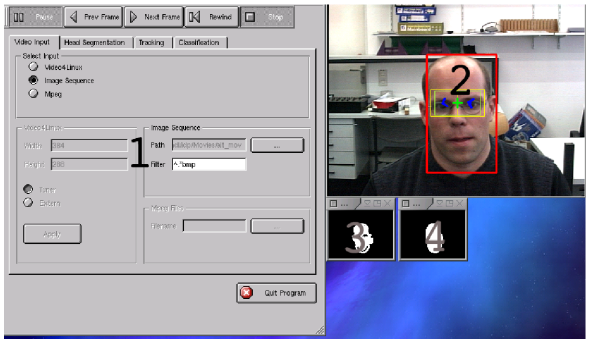


Fig. 3. A screenshot of the demonstrator GUI

rithm, the block size in which the AOF is computed, is a 40x40 px rectangle. Moreover, the default threshold for accepting the nose bridge template is 0.7. All parameters can be individually adjusted during an offline evaluation.

The three output windows are structured, as follows. The main output windows(2) shows the video source and marks relevant regions in the image. A segmented head is framed by a red bounding box, the center of the nose bridge is shown by a green cross and the search region for the nose bridge is indicated by a green rectangle. Concerning the AOF in the two halves of the face, the resulting vectors are visualized by blue arrows. The second output window(3) shows the result of the skin-color segmentation and the last window(4) visualizes the result of the morphological filters applied to the input image that identifies the set of potential head candidates.

9. CONCLUSION AND FUTURE WORK

Head gestures offer strong potential for an intuitive, efficient, and robust human-machine communication. They represent an interesting input alternative, especially in environments, where tactile interaction is difficult or error-prone. We discussed an HMM-based approach for video-based head gesture recognition and described a real-time demonstrator to be used in different application scenarios.

The existing system offers various extension possibilities. In a current approach, the head gesture recognition unit is to be coupled with a natural speech recognizer, using an early feature fusion. This allows for further improvement of the overall recognition rates and benefits from the fact that many user inputs (especially confirmation and negation) are *redundantly* linked (i.e. temporally overlapping). In a late semantic fusion approach based on a client-server architecture[10], the output of the head gesture recognition unit is combined in a central integration unit. Applying context knowledge, the integrator can dynamically vary the vocabulary of the head gesture recognizer. If, for example, in a system dialogue a yes-no answer is expected, the system could instruct the recognizer to load configuration GS_2, as other input does not make sense in this system context. By this, we expect a remarkable improvement of the recognition rate and time performance.

10. ACKNOWLEDGMENTS

The presented work has partly been supported by the FERMUS project, which is a cooperation of the BMW Group (Munich, Germany), the DaimlerChrysler AG (Stuttgart, Germany), the SiemensVDO AG (Wetzlar, Germany), and the Institute of Human-Machine Communication at the Technical University Munich (Germany). Furthermore, we would like to thank our students for their meticulousness in co-developing and evaluating several system components.

11. REFERENCES

[1] S. L. Oviatt et al., "Multimodal interface research: A science without borders," *Proc. of the 6th Int. Conf. on Spoken Lang. Processing (ICSLP)*, China 2000.

[2] C. Morimoto et al., "Recognition of head gestures using hidden markov models," *Proc. of IEEE Int. Conf. on Pattern Recognition*, Vienna, Austria 1996.

[3] J. Davis and S. Vaks, "A perceptual user interface for recognizing head gesture acknowledgements," *In WS on Perceptive User Interfaces (PUI 01)*, USA 2001.

[4] J. Tang and R. Nakatsu, "A head gesture recognition algorithm," *In Proc. of the 3rd Int. Conf. on Multimodal Interface*, Beijing, China 2000.

[5] Frank Althoff, Gregor McGlaun, Manfred Lang, and Gerhard Rigoll, "Evaluating multimodal interaction patterns in various application scenarios," in *Proc. of Gesture Workshop 2003*, April 2003, Genua, Italien.

[6] M. Yang et al., "Detecting faces in images: A survey," *In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, pp. 35–58, 2002.

[7] Barron, Fleet, and Beauchemin, "Performance of optical flow techniques," *IJCV*, vol. 12:1, pp. 43–77, 1994.

[8] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, February 1989, vol. 77:11.

[9] P. Morguet et al., "Spotting dynamic hand gestures in video image sequences using hidden markov models," *Proc. of ICIP 99*, pp. 193–197, Chicago, USA 1998.

[10] G. McGlaun et al., "A new approach for the integration of multimodal input based on late semantic fusion," *In Proc. of USEWARE 2002*, Darmstadt, Germany 2002.