Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing xxx (2006) xxx–xxx
www.elsevier.com/locate/imavis

Visual recognition of pointing gestures for human–robot interaction

Kai Nickel *, Rainer Stiefelhagen

Interactive Systems Labs, Universitaet Karlsruhe, 76131 Karlsruhe, Germany

Abstract

In this paper, we present an approach for recognizing pointing gestures in the context of human–robot interaction. In order to obtain input features for gesture recognition, we perform visual tracking of head, hands and head orientation. Given the images provided by a calibrated stereo camera, color and disparity information are integrated into a multi-hypothesis tracking framework in order to find the 3D-positions of the respective body parts. Based on the hands' motion, an HMM-based classifier is trained to detect pointing gestures. We show experimentally that the gesture recognition performance can be improved significantly by using information about head orientation as an additional feature. Our system aims at applications in the field of human–robot interaction, where it is important to do run-on recognition in real-time, to allow for robot egomotion and not to rely on manual initialization.

Keywords: Person tracking; Gesture recognition; Head orientation; Human–robot interaction

1. Introduction

In the developing field of household robotics, one aspect is of central importance for all kinds of applications that collaborate with humans in a human-centered environment: the ability of the machine for simple, unconstrained and natural interaction with its users. The basis for appropriate robot actions is a comprehensive model of the respective surrounding and in particular of the humans involved in interaction. This requires, for example, the recognition and interpretation of speech, gesture or emotion.

Among the set of gestures intuitively performed by humans when communicating with each other, pointing gestures are especially interesting for communication with robots. They open up the possibility of intuitively indicating objects and locations, e.g., to make a robot change the direction of its movement or to simply mark some object. This is particularly useful in combination with speech recognition as pointing gestures can be used to specify parameters of location in verbal statements (put the cup *there*!).

In this paper, we present a real-time system for visual user modeling (see Fig. 1). Based on images provided by a stereo camera, we combine the use of color and disparity information to locate the user's head and hands. Guided by a probabilistic body model, we find the trajectories of head and hands using a multi-hypothesis tracking framework. In addition, we estimate the orientation of the head, following an appearance-based neural-network approach. Although this is a very basic representation of the human body, we show that it can be used successfully for the recognition of pointing gestures: we present an HMM-based pointing gesture recognizer that detects the occurrence of pointing gestures within natural hand movements and estimates the pointing direction. We show that incorporating head-orientation information helps to improve gesture recognition performance.

The remainder of this paper is organized as follows: In Section 2, we present our system for tracking a user's head and hands. The estimation of head orientation is explained in Section 3. In Section 4, we describe our approach to recognize pointing gestures. Finally, in Section 5, we present

* Corresponding author.
E-mail addresses: nickel@ira.uka.de (K. Nickel), stiefel@ira.uka.de (R. Stiefelhagen).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.12.020

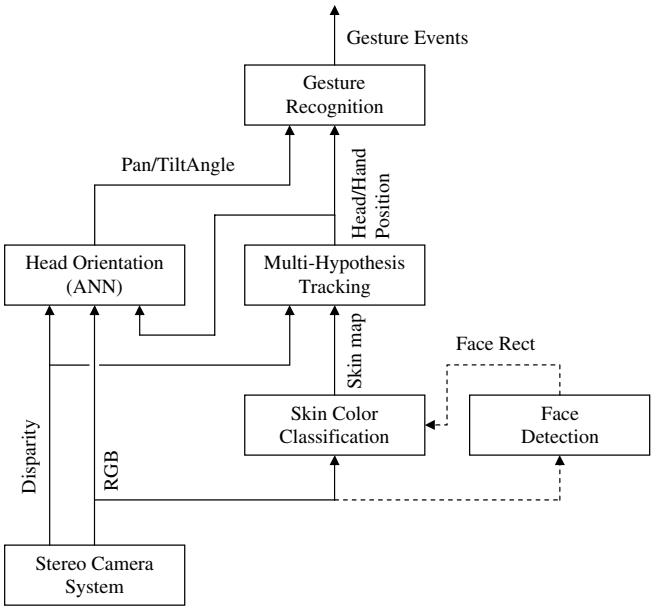

Fig. 1. Overview about the gesture recognition and tracking system presented in this paper.

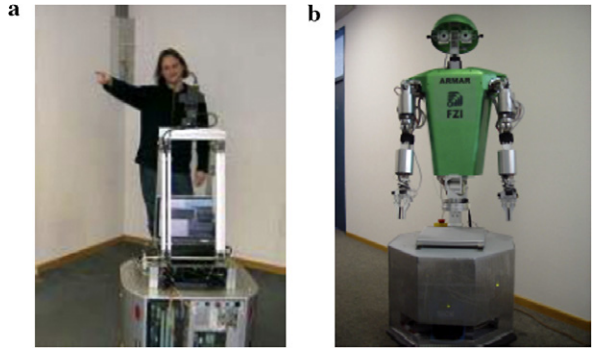

Fig. 2. (a) Interaction with our development system. Software components include: speech recognition and synthesis, person and gesture tracking and multimodal dialogue management. (b) Part of the components have already been integrated in a humanoid robot with 2 arms [3].

experimental results on gesture recognition using all the features provided by the visual tracker.

1.1. Target scenario

To facilitate natural interaction, robots should be able to perceive and understand all the modalities used by humans during face-to-face interaction. Apart from speech, which is probably the most prominent modality used by humans, these modalities also include pointing gestures, facial expressions, head pose, gaze, eye contact and body language for example.

The target scenario we address is a household situation in which a human can ask the robot questions related to the kitchen (such as "What's in the fridge?"), ask the robot to set the table, to switch certain lights on or off, to bring certain objects or obtain recipes from the robot.

Apart from person tracking and gesture recognition, the current components of our system include: a speech recognizer, a dialogue manager and a speech synthesizer. The software is running on a mobile robot platform equipped with a stereo camera head that can be rotated with a pan-tilt unit.

Fig. 2a shows a picture of our system and a person interacting with it. Part of the visual tracking components have already been integrated in ARMAR [3], a humanoid robot with two arms and 23 degrees of freedom (see Fig. 2b).

1.2. Related work

Visual person tracking is of great importance not only for human–robot interaction but also for cooperative multi-modal environments or for surveillance applications. There are numerous approaches for the extraction of body features using one or more cameras.

In their well-known work [11], Wren et al. demonstrate the system Pfinder, that uses a statistical model of color and shape to obtain a 2D representation of head and hands. In addition, Azarbayejani and Pentland [12] show how to calibrate a stereo camera setup automatically based on head/hand blob features – thus adding 3D-coordinates to a Pfinder-like head and hand tracker. Yet, the human silhouette extraction method used in Pfinder relies on a static background assumption, which is – in it's generic form – hardly applicable to our mobile robot scenario.

Based on the additional information that is available from dense stereo processing, Darrell et al. [13] present their approach for person tracking using disparity images, color cues and face detection in an integrated framework. However, they concentrate on silhouette and face tracking, and do not address the problem of hand tracking.

In order to analyze human pointing gestures, there are several approaches that concentrate on different parts of the body. Kahn et al. [18] demonstrate the use of pointing gestures to locate objects. Their system operates on various feature maps (intensity, edge, motion, disparity, color) that are utilized to track head and hands. In [25], Cipolla et al. use the shape of the hand – i.e., the index finger – from two different views in order to locate exactly the pointing destination on a 2-dimensional workspace. Jojic et al. [19] recognize pointing gestures by decomposing the disparity image of a standing subject into two parts: an outstreched arm and the rest of the body. Unlike these approaches, we combine 3D head and hand tracking with head orientation in order to model the *dynamic* motion of pointing gestures instead of static pose.

Hidden Markov Models – well-known for their use in speech recognition [21] – have already been applied successfully to the field of gesture recognition: In [14], Starner and Pentland were able to recognize hand gestures out of the vocabulary of the American Sign Language with high accuracy. Becker [15] presents a system for the recognition of T'ai Chi gestures based on head and hand tracking. In

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

Fig. 3. Feature for locating head and hands: In the skin color map (a), dark pixels represent high skin-color probability. The disparity map (b) is made up of pixel-wise disparity measurements; the brightness of a pixel corresponds to its distance to the camera. Finally, skin-colored pixels are spatially clustered (c); the clusters are depicted by circles.

[16], Wilson and Bobick propose an extension to the HMM framework, that addresses characteristics of parameterized gestures, such as pointing gestures. Poddar et al. [17] recognize different hand gestures performed by a TV weather person. They combine an HMM-based detection of gestures on head and hand movements with spoken keywords.

2. Tracking head and hands

In order to gain information about the location and posture of a person in the vicinity of the robot, we track the 3D-positions of the person's head and hands. These trajectories are important features for the recognition of natural gestures, including pointing gestures. In our approach, we combine color and range information to achieve robust tracking performance.

Our setup consists of a fixed-baseline stereo camera head connected to a standard PC. A commercially available library[1] is used to calibrate the cameras, to search for image correspondence and to calculate 3D-coordinates for each pixel.

2.1. Locating head and hands

The first and maybe the most important thing to be taken into consideration when designing a tracking system is the choice of features. As we aim for real-time operation, we decided to implement a fast 3D-blob tracker. Head and hands can be identified by color as human skin color clusters in a small region of the chromatic color space [20]. To model the skin-color distribution, two histograms (S^+ and S^-) of color values are built by counting pixels belonging to skin-colored and *not*-skin-colored regions respectively, in sample images. By means of the histograms, the ratio between $P(S^+|x)$ and $P(S^-|x)$ is calculated for each pixel x of the color image, resulting in a gray-scale map of skin-color probability (Fig. 3a).

A combination of morphological operations with a 3×3 structuring element is applied to the skin-color map: first, a dilation connects neighboring pixels in order to produce closed regions. Then a combination of two erosions eliminates isolated pixels. A final dilation is used to roughly restore the original region size.

In order to find potential *candidates* for the coordinates of head and hands, we search for connected regions in the thresholded skin-color map. For each region, we calculate the centroid of the associated 3D-pixels which are weighted by their skin-color probability. If the pixels belonging to one region vary strongly with respect to their distance to the camera, the region is split by applying a k-means clustering method (see Fig. 3c). We thereby separate objects that are situated on different range levels, but accidentally merged into one object in the 2D-image.

A mobile robot will have to cope with frequent changes in light conditions. Thus, it is essential to initialize the color model automatically, and to continuously update the model to accommodate changes. In order to do this, we search for a face in the camera image by running a fast face detection algorithm [1] asynchronously to the main video loop (see Fig. 1). Whenever a face is found, a new color model is created based on the pixels inside the face region. Two things are done to prevent the color model from being impaired by a wrongly detected face: first, the new color model will only be accepted if it classifies a high percentage of pixels inside the face region positively, and a high percentage of pixels outside the face region negatively.[2] Second, the new model is merged with the existing model using an update factor α. This factor is chosen such that hard changes in light conditions will be assimilated after some seconds, whereas occasional misdetections of the face do not impair the model significantly.

2.2. Single-hypothesis tracking

The task of tracking consists in finding the best hypothesis s_t for the positions of head and hands at each time t. The decision is based on the current observation O_t and the hypotheses of the past frames, $s_{t-1}, s_{t-2}, \ldots$. We have

[1] SRI Small Vision System, http://www.videredesign.com/svs.htm.

[2] A color model that does not fulfill these requirements would be useless for head and hand tracking.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

formulated this in a probabilistic framework, which includes the following 3 components:

- The observation score $P_o(O_t|s_t)$.
- The posture score $P_p(s_t)$.
- The transition score $P_t(s_t|s_{t-1},s_{t-2},\ldots)$.

With each new frame, all combinations of the 3D-skin-cluster centroids are evaluated to find the hypothesis s_t that exhibits the highest results with respect to the product of the 3 scores.

2.2.1. The observation score

$P_o(O_t|s_t)$ is a measure for the extent to which s_t matches the observation O_t. In order to evaluate this score, we sum up the weights of all skin-pixels that can be found inside ellipses that are placed around the head and hands positions in s_t. The radius of an ellipse is given by the average size of a human head/hand and is scaled with respect to its 3D position.

2.2.2. The posture score

$P_p(s_t)$ is the prior probability of the posture. It is high if the posture represented by s_t is a frequently occurring posture of a human body. It is equal to zero if s_t represents a posture that breaks anatomical constraints. To be able to calculate $P_p(s_t)$, a model of the human body was built from training data. The model comprises a distribution of body height as well as a series of constraints like the maximum distance between head and hand. As can be seen in Fig. 4, the position of the hand relative to the head tends to lie in (but is not limited to) a curved region In addition, a 3-dimensional gaussian mixture model was trained on labeled hand positions, thus representing a probability distribution of hand-positions relative to the head (see Fig. 4).

2.2.3. The transition score

$P_t(s_t|s_{t-1},s_{t-2},\ldots)$ is a measure for the probability of s_t being the successor of the past frames' hypotheses.

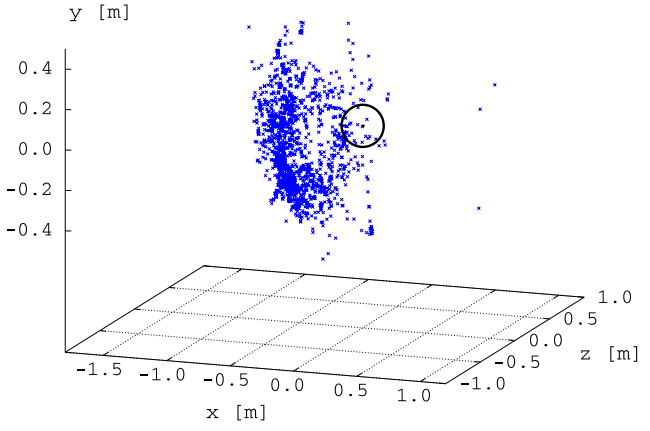

Fig. 4. Observed positions of the right hand relative to the head (depicted by a circle) over a time of 2 min.

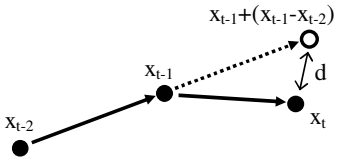

Fig. 5. The transition score considers the distance d between the predicted position and the currently measured position x_t.

Let x_t denote the position of a body part in s_t. According to Fig. 5, the distance between the predicted position and the measured position is given by $d = \|x_{t-1} + (x_{t-1} - x_{t-2}) - x_t\|$. The transition score for body part x is then related to d like follows:

$$P_t^x(s_t|s_{t-1},s_{t-2},\ldots) = \max(1 - \frac{d}{d_{\max}}, \epsilon) \quad (1)$$

where $d_{\max}$ is a limit for the natural motion of the respective body part in the time between the frames. The small value ϵ guarantees that the score remains positive.[3] The final transition score is the product of the three body parts' transition scores.

Our experiments indicate that by using the described method, it is possible to track a person robustly, even when the camera is moving and when the background is cluttered. The tracking of the hands is affected by occasional dropouts and misclassification. Reasons for this can be temporary occlusion of a hand, a high variance in the visual appearance of hands and the high speed with which people move their hands.

2.3. Multi-hypothesis tracking

Accurate tracking of the small, fast moving hands is a hard problem compared to the tracking of the head. Deciding which hand is actually the left or the right one is especially difficult. Given the assumption that the right hand will *in general* be observed more often on the right side of the body, the tracker could perform better, if it were able to correct its decision from a future point of view, instead of being tied to a (wrong) decision it once made.

We implemented multi-hypotheses tracking to allow such kind of rethinking: At each frame, an n-best list of hypotheses is kept, in which each hypothesis is connected to it's predecessor in a tree structure. The tracker is free to choose the path, that maximizes the overall probability of observation, posture and transition. The algorithm performs the following steps:

(1) Build a list of all hypotheses $s_t^{1\ldots m}$ whose P_o and P_p scores exceed a minimum threshold.
(2) For each of these m new hypotheses calculate P_t with respect to each of the n last frame's hypotheses. This

[3] $P_t(s_t—s_{t-1},s_{t-2},\ldots)$ should always be positive, so that the tracker can recover from erroneous static positions.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

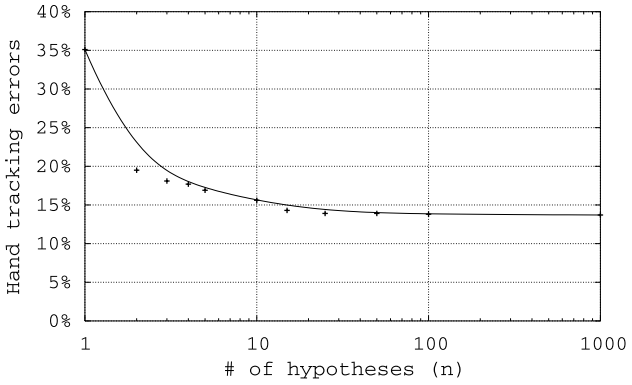

Fig. 6. Percentage of frames with hand-tracking errors in relation to the number of hypotheses per frame (n).

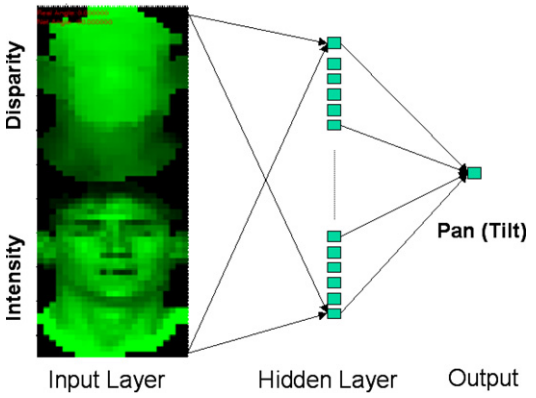

Fig. 7. For head pose estimation, intensity and disparity images of the head are scaled to a constant size of 24×32 pixel and then classified by ANNs.

leads to a total score $P_p \cdot P_o \cdot P_t$ for each combination of old and new hypotheses.

(3) Select the n best hypotheses from this list of $m \cdot n$ combinations. Add each of them as child to their parent hypothesis.

(4) Remove tree branches that have no successor in the current frame's list of hypotheses. Also remove branches that split from the currently best branch more that x seconds ago.[4]

(5) Normalize the total scores of the remaining current hypotheses so that they sum up to 1.

The introduction of multi-hypothesis tracking improves the performance of hand-tracking significantly. Fig. 6 shows the reduction of hand-tracking errors by increasing the number n of hypotheses per frame. In order to detect tracking errors, we labeled head and hand centroids manually. An error is assumed, when the distance of the tracker's hand position to the labeled hand position is higher than 0.15 m. Confusing left and right hand therefore counts as two errors.

3. Head orientation

A body of literature suggests that humans are generally interested in what they look at [4–6]. In addition, recent user studies reported strong evidence that people naturally look at the objects or devices with which they interact [7,9]. In our recorded data, we also noticed that people tend to look at pointing targets in the begin- and in the hold-phase of a gesture (see section 4). This behavior is likely due to the fact that the subjects needed to (visually) find the objects at which they wanted to point.

In order to evaluate, whether this behavior can be used to improve pointing gesture recognition, we need to obtain spatial information about the user's focus. A practical reason to use head orientation to estimate a person's focus of attention, is, that in scenarios addressed in this work, head orientation can be estimated with non-intrusive methods while eye gaze can not. In previous work [2], we found that robust head pose estimation results could be achieved using an appearance based approach, where head pose is estimated from facial images using artificial neural networks (ANN). This approach has proven to work with high-resolution as well as with low-resolution images.

Changes in light condition are one of the main problems of image-based approaches. In order to decrease the problem, we incorporate the disparity image into the ANN's input pattern. This has been shown to reduce the classification error significantly under changed light conditions [23].

The networks we use have a total number of 1597 neurons, organized in 3 layers (see Fig. 7). They were trained with standard back-propagation in a person-independent manner on sample images of rotated heads. Ground truth for the training samples was obtained using a magnetic pose tracker.

The procedure for head pose estimation works like follows: in each frame, the head's bounding box – as provided by the tracker – is resampled to a size of 24×32 pixels and then histogram normalized. Two neural networks, one for pan and one for tilt angle, process the head's intensity and disparity image and output the respective rotation angles. In our test-set, the mean error of person-independent head orientation estimation was 9.7° for pan- and 5.6° for tilt-angle.

4. Recognition of pointing gestures

When modeling pointing gestures, we try to model the typical motion pattern of pointing gestures – and not only the static posture of a person during the peak of the gesture. We decompose the gesture into three distinct phases and model each phase with a dedicated HMM. The features used as the models' input are derived from tracking the position of the pointing hand as well as position and orientation of the head.

[4] We freeze that part of the trajectory that is older than $x = 1$ s, because in a real-time application we do not want to delay the following interpretation of the tracker's output too much. This would conflict with the responsiveness of the system.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

4.1. Phase models

According to [8], the temporal structure of hand gestures can be divided into three phases: preparation, peak and retraction. Knowing the characteristics of these phases may help in the temporal segmentation of gestures from other hand movements.

We recorded 210 pointing gestures performed by 15 different persons. Looking at the pointing gestures in this dataset, we could easily identify the following phases in the movement of the pointing hand:

- Begin (B): the hand moves from an arbitrary starting position towards the pointing target.
- Hold (H): the hand remains motionless at the pointing position.
- End (E): the hand moves away from the pointing position.

Instead of using one HMM for the complete gesture, we decided to train one dedicated HMM for each of the three phases. The main reason for that is, that we want to be able to detect the hold phase separately. Identifying the hold-phase precisely is of great importance for the correct estimation of the pointing direction. However, the hold-phase has the highest variance in duration and can often be very short (see Table 1), thus potentially showing little evidence in an HMM which models the complete gesture.

The topology of the HMMs has been determined experimentally and is depicted in Fig. 8. Given the amount of available training data (see Section 5), the three phases' models $M_{B,H,E}$ have been found to perform best with three states each,[5] and an output probability that is modeled by a mixture of two Gaussians per state. In addition to the phase models there is a null-model M_0, that is trained on sequences that are any hand movements but no pointing gestures. M_0 acts as a threshold for the phase models' output. All models were trained by means of the EM-Algorithm using the 5-dimensional feature vector presented in Section 4.3.

4.2. Segmentation

For the task of human–robot interaction we need to do run-on recognition, meaning that a pointing gesture has to be recognized immediately after it has been performed. As a consequence, we have to analyze the observation sequence each time a new frame has been processed. There is no chance to correct a wrong decision afterwards.

The length of the three phases varies strongly from one gesture to another. Any fixed size classification window would either contain additional non-phase hand movements or only a fraction of the gesture phase. In both cases, the HMM would not match well with the observation. Therefore, we follow an approach presented in [15] and classify not only one, but a series of sequences $s_{1\ldots n}$. These sequences have different sizes, but they all end with the current frame. The sizes vary between $\mu \pm 2\sigma$ according to Table 1, thus covering the range of phase lengths observed in training. For each of the phases $p \in \{B,H,E\}$, we search for the ideal subsequence $\hat{s}_p$, that contains nothing but the complete gesture phase. We find $\hat{s}_p$ by classifying all sequences and selecting the one with the highest output probability.[6] Because $P(\hat{s}_p|M_0)$ represents the probability that $\hat{s}_p$ is *not* part of a pointing gesture, we use it to normalize the phase-models' output probabilities.

$$\hat{s}_p = \operatorname{argmax} \log P(s_{1\ldots n}|M_p) \qquad (2)$$
$$P_p = \log P(\hat{s}_p|M_p) - \log P(\hat{s}_p|M_0)$$

In order to detect a pointing gesture, we have to search for three subsequent time intervals that have high output probabilities P_B, P_H and P_E. Ideally, the respective model would significantly dominate the other two models in its interval. But as Fig. 9 shows, M_H tends to dominate the other models in the course of a gesture. That is why we detect a pointing gesture whenever we find three points in time, $t_B < t_H < t_E$, so that

$$P_B(t_B), P_H(t_H), P_E(t_E) > 0 \qquad (3)$$
$$P_E(t_E) > P_B(t_E)$$
$$P_B(t_B) > P_E(t_B)$$

Table 1
Average length μ and standard deviation σ of pointing gesture phases

	μ (s)	σ (s)
Complete gesture	1.75	0.48
Begin	0.52	0.17
Hold	0.72	0.42
End	0.49	0.16

A number of 210 gestures performed by 15 test persons have been evaluated.

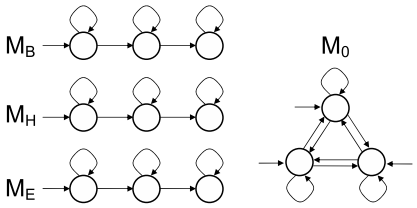

Fig. 8. For modeling the phases of pointing gestures, 3-state HMMs (2 Gaussians per state) are used. An ergodic HMM represents non-gesture sequences.

[5] As the gesture is already decomposed into three dedicated HMMs, there is no semantic meaning of the single states of each HMM.

[6] As pointed out by [15], this can be done quickly by running only one pass of the Viterbi algorithm on the longest sequence.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

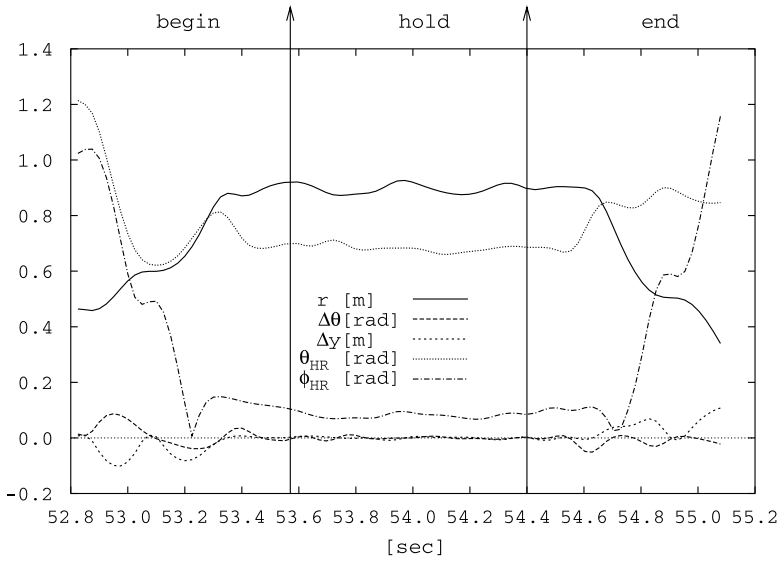

Fig. 11. Feature sequence of a typical pointing gesture.

Once a gesture has been detected, its hold-phase is being processed for pointing direction estimation (see Section 4.4), and the system is set to sleep for a small amount of time to avoid the same gesture being recognized multiple times.

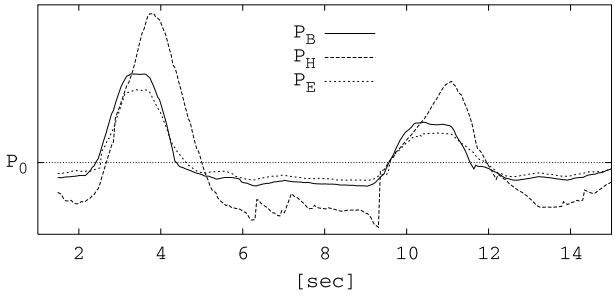

Fig. 9. Log-probabilities of the phase-models during a sequence of two pointing gestures.

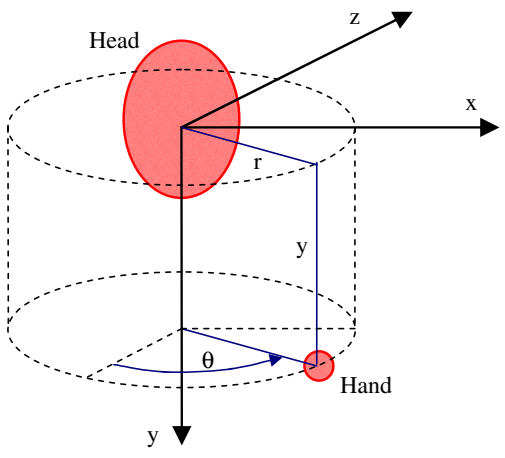

Fig. 10. The hand position is transformed into a cylindrical coordinate system.

4.3. Features

We evaluated different transformations of the hand position vector, including cartesian, spherical and cylindrical coordinates.[7] In our experiments it turned out that cylindrical coordinates (θ,r,y) of the hands (see Fig. 10) produce the best results for the pointing task.

The origin of the hands' coordinate system is set to the center of the head, thus we achieve invariance with respect to the person's location. As we want to train only one model to detect both left and right hand gestures, we mirror the left hand to the right hand side by changing the sign of the left hand's x-coordinate. Since the model should not adapt to absolute hand positions – as these are determined by the specific pointing targets within the training set – we use the deltas (velocities) of θ and y instead of their absolute values.

In order to incorporate head orientation into the feature vector, we calculate the following two features:

$$\theta_{HR} = |\theta_{Head} - \theta_{Hand}|$$
$$\phi_{HR} = |\phi_{Head} - \phi_{Hand}|$$

θ_{HR} and ϕ_{HR} are defined as the absolute difference between the head's and the hand's azimuth and elevation angle respectively. Fig. 11 shows a plot of all feature values during the course of a typical pointing gesture. As can be seen in the plot, the values of the head-orientation features θ_{HR} and ϕ_{HR} decrease in the begin-phase and increase in the end-phase. In the hold-phase, both values are low, which indicates that the hand is "in line" with the head orientation.

[7] See [22] for a comparison of different feature vector transformations for gesture recognition.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

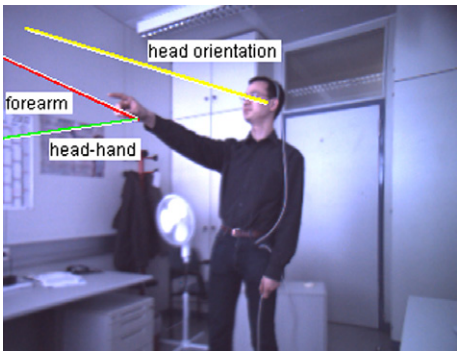

Fig. 12. Different approaches for estimating the pointing direction. (The lines were extracted in 3D and projected back to the camera image.)

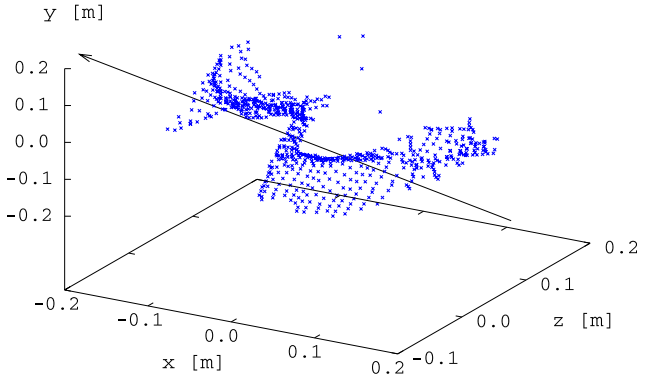

Fig. 13. A principal component analysis of the 3D pixels around the center of the hand reveals the orientation of the forearm (arrow).

4.4. Estimation of the pointing direction

We explored three different approaches (see Fig. 12) to estimate the direction of a pointing gesture: (1) the line of sight between head and hand, (2) the orientation of the forearm, and (3) head orientation. While the head and hand positions as well as the forearm orientation were extracted from stereo-images, the head orientation was measured by means of a magnetic sensor.

In order to identify the orientation of the forearm, we calculate the covariance matrix C of the 3D-pixels that lie within a 20cm radius around the center of the hand. The eigenvector e_1 with the largest eigenvalue (the first principal component) of C denotes the direction of the largest variance of the data set. As the forearm is an elongated object, we expect e_1 to be a measure for the direction of the forearm (see Fig. 13). This approach assumes that no other objects are present within the critical radius around the hand, as those would influence the shape of the point set. We found that in the hold phase, this assumption generally holds.[8]

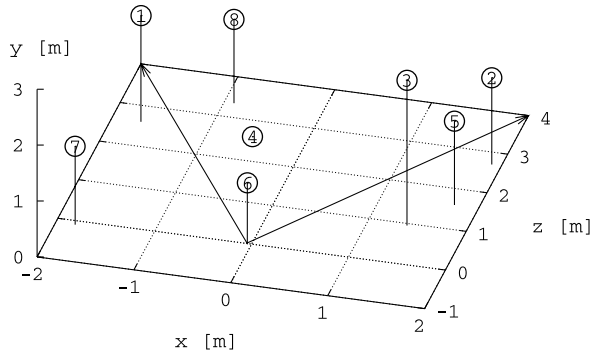

Fig. 14. Target positions in the test set. Target #6 is equal to the camera position. The arrows indicate the camera's field of view.

5. Experiments and results

In order to evaluate the performance of gesture recognition, we prepared an indoor test scenario with 8 different pointing targets (see Fig. 14). Test persons were asked to imagine the camera were a household robot. They were to move around within the camera's field of view, every now and then showing the camera one of the marked objects by pointing on it. In total, we captured 129 pointing gestures by 12 subjects. The recorded video was then analyzed offline.

5.1. Pointing direction

For evaluating the quality of pointing direction estimation, we labeled the hold phases manually. Thus, we are not affected by potential gesture recognition errors. Nevertheless, there is an error induced by the stereo vision system, because the camera coordinates do not comply perfectly with the manual measurements of the targets' positions.

In our experiment, the head–hand line achieved an average precision of 25°, allowing for 90% correct target identification (see Table 2). The forearm line performed noticeably worse than the head–hand line. We believe that this is mainly the result of erroneous forearm measurements.[9] Unlike the relatively stable head position, the forearm measurements vary strongly during the hold phase. The results of pointing direction estimation based on head orientation are comparable to the ones obtained with the head–hand line. In this experiment, however, head orientation was not extracted visually, but with a magnetic sensor attached to the subjects' head.

5.2. Gesture recognition

In order to determine the gesture recognition performance, we used a leave-one-out evaluation strategy;

[8] Nevertheless, we reject the forearm measurement, when the ratio e_1/e_2 of the first and the second principal component is <1.5.

[9] The test persons were pointing with an outstretched arm almost every time, thus reducing the potential benefit even of a more accurate forearm measurement.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

Table 2
Comparison of three approaches for pointing direction estimation: (a) average angle between extracted 3D pointing line and ideal line to the target, (b) percentage of gestures for which the correct target (1 out of 8) was identified, and (c) availability of measurements during the hold-phase

	Head–hand line	Forearm line	Sensor head orientation
(a) Average error angle	25°	39°	22°
(b) Targets identified	90%	73%	75%
(c) Availability	98%	78%	(100%)

i.e., we trained the Hidden Markov models on data from 11 of the 12 subjects and evaluated on the remaining person. Two measures are used to quantify the performance: the *recall* value is the percentage of pointing gestures that have been detected correctly, while the *precision* value is the ratio of the number of correctly detected gestures to the total number of detected gestures (including false positives). The results given in Table 3 are averaged over all persons.

To find out whether the HMM-based pointing gesture recognizer can benefit from head orientation, we ran the evaluation three times with different feature vectors: (1) hand position only, (2) hand position + head orientation obtained with the attached sensor, and (3) hand position + head orientation obtained visually with ANNs.

Our baseline system without head-orientation scored at about 80% recall and 74% precision in gesture recognition. When head orientation was added to the feature vector, the results improved significantly in the precision value from about 74–87%, while the recall value remained at a similarly high level. In other words, the number of false positives could be reduced by 50% relative considering head orientation as an additional cue. It is interesting to note that although there were noise and measurement errors in the visual estimation of head orientation, there was no significant difference in gesture recognition performance between visually and magnetically extracted head orientation.

In addition to gesture detection, we also evaluated pointing direction estimation on the automatically detected hold phases. By including head orientation, the average error was reduced from 19.4° to 16.9°. As the pointing direction estimation is based on the head- and hand-trajectories – which are the same in both cases – the error reduction is the result of the model's increased ability of locating the gesture's hold-phase precisely.

Table 3
Performance of person-independent pointing gesture recognition with and without including head orientation to the feature vector

	Recall (%)	Precision (%)	Error (°)
No head orientation	79.8	73.6	19.4
Sensor head orientation	78.3	86.3	16.8
Visual head orientation	78.3	87.1	16.9

6. Conclusion

We have demonstrated a real-time vision system which is able to detect pointing gestures, and to estimate the pointing direction. The person tracking component integrates color and depth information in a probabilistic framework in order to robustly obtain the 3D-trajectories of head and hands. By following a multi-hypotheses approach, we could improve hand tracking and achieve about 60% relative error reduction.

We could show that the human behavior of looking at the pointing target can be exploited for automatic pointing gesture recognition. By using visual estimates for head orientation as additional features in the gesture model, both the recognition performance and the quality of pointing direction estimation increased significantly. In an experiment (human–robot interaction scenario) we observed a 50% relative reduction of the number of false positives produced by the system and a 13% relative reduction in pointing direction error when using the additional head-orientation features. We explored different approaches for extracting the pointing direction and found the head–hand line to be a good estimate.

It is clear that in a natural interaction scenario, pointing gestures usually co-occur with speech. And although the pure visual recognition performs well in the domain of human–robot interaction, we believe that substantial improvements can be made by considering the multi-modal nature of the gesture. In [24] we present an attempt to merge the output of this gesture recognizer with the output of a speech recognizer. Guided by a multi-modal dialogue manager, the detection and interpretation of deictic actions can be improved compared to the results using only one of the modalities.

Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588 "Humanoide Roboter".

References

[1] P. Viola, M. Jones, Robust real-time object detection, ICCV Workshop on Statistical and Computation Theories of Vision, Vancouver, Canada, July 2001.
[2] R. Stiefelhagen, J. Yang, A. Waibel, Simultaneous Tracking of Head Poses in a Panoramic View, International Conference on Pattern Recognition – ICPR 2000, Barcelona, Spain, Sept. 2000.
[3] T. Asfour, A. Ude, K. Berns, R. Dillmann, Control of Armar for the realization of anthropometric motion patterns, in: Proc. of the 2nd IEEE-RAS Conference on Humanoid Robots (Humanoids 2001), Tokyo, Japan, Nov. 2001.
[4] A.L. Yarbus, Eye movements during perception of complex objectsEye Movement and Vision, Plenum Press, New York, 1967, pp. 171–196.
[5] P. Barber, D. Legge, Perception and informationChapter 4: Information Acquisition, Methuen, London, 1976.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020

[6] A.J. Glenstrup, T. Engell-Nielsen, Eye Controlled Media: Present and Future State, Technical Report, University of Copenhagen, 1995.
[7] P.P. Maglio, T. Matlock, C.S. Campbel, S. Zhai, B.A. Smith, Gaze and speech in attentive user interfaces, in: Proceedings of the International Conference on Multimodal Interfaces, 2000.
[8] V.I. Pavlovic, R. Sharma, T.S. Huang, Visual Interpretation of Hand Gestures for Human–Computer Interaction: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 677–695, vol. 19, No. 7, July 1997.
[9] B. Brumitt, J. Krumm, B. Meyers, S. Shafer, Let there be light: comparing interfaces for homes of the future, IEEE Personal. Commun. (2000).
[11] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997).
[12] A. Azarbayejani, A. Pentland, Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. in: Proceedings of 13th ICPR, 1996.
[13] T. Darrell, G. Gordon, M. Harville, J. Woodfill, Integrated person tracking using stereo, color, and pattern detection. IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998.
[14] T. Starner, A. Pentland, Visual Recognition of American Sign Language Using Hidden Markov Models. M.I.T. Media Laboratory, Perceptual Computing Section, Cambridge MA, USA, 1994.
[15] D.A. Becker, Sensei: A Real-Time Recognition, Feedback and Training System for T'ai Chi Gestures. M.I.T. Media Lab Perceptual Computing Group Technical Report No. 426, 1997.
[16] A.D. Wilson, A.F. Bobick, Recognition and interpretation of parametric gesture, Intl. Conf. Comput. Vis. ICCV (1998) 329–336.
[17] I. Poddar, Y. Sethi, E. Ozyildiz, R. Sharma. Toward Natural Gesture/Speech HCI: A Case Study of Weather Narration. Proc. Workshop on Perceptual User Interfaces (PUI98), San Francisco, USA, 1998.
[18] R. Kahn, M. Swain, P. Prokopowicz, R. Firby, Gesture recognition using the Perseus architecture, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 734–741, 1996.
[19] N. Jojic, B. Brumitt, B. Meyers, S. Harris, T. Huang. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. IEEE Intl. Conference on Automatic Face and Gesture Recognition, Grenoble, 2000.
[20] J. Yang, W. Lu, A. Waibel. Skin-color modeling and adaption. Technical Report of School of Computer Science, CMU, CMU-CS-97-146, 1997.
[21] L.R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[22] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, A. Pentland, Invariant features for 3-D gesture recognition, 2nd Intl. Workshop on Face and Gesture Recognition, Killington VT, 1996.
[23] E. Seemann, K. Nickel, R. Stiefelhagen, Head Pose Estimation Using Stereo Vision For Human–Robot Interaction, in: 6th Intl. Conf. on Automatic Face and Gesture Recognition, Seoul, Korea, 2004.
[24] H. Holzapfel, K. Nickel, R. Stiefelhagen, Implementation and Evaluation of a Constraint Based Multimodal Fusion System for Speech and 3D Pointing Gestures, in: 6th Intl. Conf. on Multimodal Interfaces, State College, USA, 2004.
[25] R. Cipolla, P.A. Hadfield, N.J. Hollinghurst, Uncalibrated Stereo Vision with Pointing for a Man–Machine Interface, in: Proc. of the IAPR Workshop on Machine Vision Application, pp. 163–166, 1994.

Please cite this article in press as: K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image Vis. Comput. (2006), doi:10.1016/j.imavis.2005.12.020