Real-time Person Tracking and Pointing Gesture Recognition for Human-Robot Interaction

Kai Nickel and Rainer Stiefelhagen

Interactive Systems Laboratories
Universität Karlsruhe (TH), Germany
nickel@ira.uka.de, stiefel@ira.uka.de

Abstract

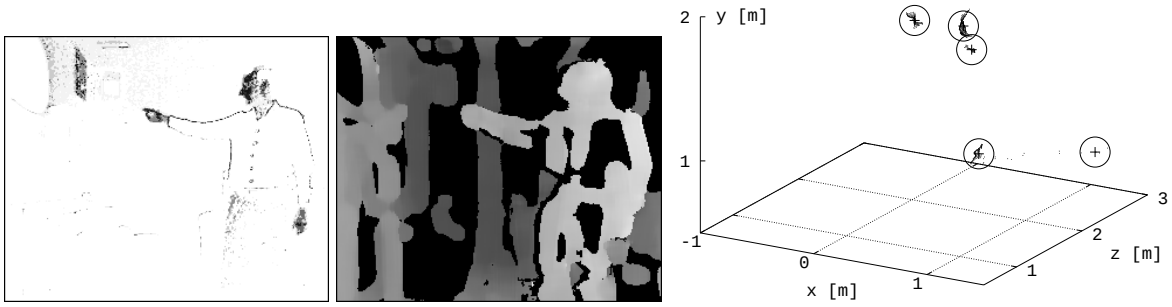
In this paper, we present our approach for visual tracking of head, hands and head orientation. Given the images provided by a calibrated stereo-camera, color and disparity information are integrated into a multi-hypotheses tracking framework in order to find the 3D-positions of the respective body parts. Based on the hands' motion, an HMM-based approach is applied to recognize pointing gestures. We show experimentally, that the gesture recognition performance can be improved significantly by using visually gained 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's egomotion and not to rely on manual initialization.

1 Introduction

In the upcoming field of household robots, 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 current surrounding and in particular of the humans involved in interaction. This might require for example the recognition and interpretation of speech, gesture or emotion.

In this paper, we present our current real-time system for visual user modeling. Based on images provided by a stereo-camera, we combine the use of color and disparity information to track the positions of the user's head and hands and to estimate head orientation. 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 and the estimation of the pointing direction.

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 its direction of 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*!).



a. Skin-color map b. Disparity map c. 3D skin-color clusters

Fig. 1. Feature for locating head and hands. In the skin color map, dark pixels represent high skin-color probability. The disparity map is made up of pixel-wise disparity measurements; the brightness of a pixel corresponds to its distance to the camera. Skin-colored 3D-pixels are clustered using a k-means algorithm. The resulting clusters are depicted by circles.

A body of literature suggests that people naturally tend to look at the objects with which they interact [1] [2]. In a previous work [3] it turned out, that using information about head orientation can improve accuracy of gesture recognition significantly. That previous evaluation has been conducted using a magnetic sensor. In this paper, we present experiments in pointing gesture recognition using our *visually* gained estimates for head orientation.

The remainder of this paper is organized as follows: In Section 2 we present our system for tracking a user's head, hands and head orientation. In Section 3 we describe our approach to recognize pointing gestures and to estimate the pointing direction. In Section 4 we present experimental results on gesture recognition using all the features provided by the visual tracker. Finally, we conclude the paper in Section 5.

1.1 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 [4], 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. Azarbayejani and Pentland [5] describe a 3D head and hands tracking system that calibrates automatically from watching a moving person. An integrated person tracking approach based on color, dense stereo processing and face pattern detection is proposed by Darrell et al. in [6].

Hidden Markov Models (HMMs) have successfully been applied to the field of gesture recognition. In [7], Starner and Pentland were able to recognize hand gestures out of the vocabulary of the American Sign Language with high accuracy. Becker [8] presents a system for the recognition of Tai Chi gestures based on head and hand tracking. In [9], Wilson and Bobick propose an extension to

the HMM framework, that addresses characteristics of parameterized gestures, such as pointing gestures. Jojic et al. [12] describe a method for the estimation of the pointing direction in dense disparity maps.

1.2 Our target scenario: Interaction with a household robot

The work presented in this paper is part of our effort to build technologies which aim at enabling natural interaction between humans and robots. In order to communicate naturally with humans, a robot should be able to perceive and interpret all the modalities and cues that humans use during face-to-face communication. These include speech, emotions (facial expressions and tone of voice), gestures, gaze and body language. Furthermore, a robot must be able to perform dialogues with humans, i.e. the robot must understand what the human says or wants and it must be able to give appropriate answers or ask for further clarifications.

We have developped and integrated several components for human-robot interaction with a mobile household robot. The target scenario we addressed 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 to obtain suggested recipes from the robot. The current software compontents of the robot include a speech recognizer (user-independent large vocabulary continuous speech), a dialogue component, speech synthesis and the vision-based tracking modules (face- and hand-tracking, gesture recognition, head pose). The vision-based components are used to

- locate and follow the person being tracked
- to disambiguate objects that were referenced during a dialogue ("Switch on *this* light ", "Give me this cup"). This is done by using both speech and detected pointing gestures in the dialogue model.

Figure 2 shows a picture of the mobile robot and a person interacting with it.

Fig. 2. Interaction with the mobile robot. Software components of the robot include: speech recognition, speech synthesis, person and gesture tracking, dialogue management and multimodal fusion of speech and gestures.

2 Tracking Head and Hands

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

In addition to the position of the head, we also measure head orientation using neural networks trained on intensity and disparity images of rotated heads.

Our setup consists of a fixed-baseline stereo camera head connected to a standard PC. A commercially available library [13] 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

Head and hands can be identified by color as human skin color clusters in a small region of the chromatic color space [14]. To model the skin-color distribution, two histograms (S^+ and S^-) of color values are built by counting pixels belonging to skin-colored respectively *not*-skin-colored regions 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 grey-scale map of skin-color probability (Fig. 1.a). To eliminate isolated pixels and to produce closed regions, a combination of morphological operations is applied to the skin-color map.

In order to initialize and maintain the skin-color model automatically, we search for a person's head in the disparity map (Fig. 1.b) of each new frame. Following an approach proposed in [6], we first look for a human-sized connected region, and then check its topmost part for head-like dimensions. Pixels inside the head region contribute to S^+, while all other pixels contribute to S^-. Thus, the skin-color model is continually updated to accommodate changes in light conditions.

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. 1.c). We thereby separate objects that are situated on different range levels, but accidentally merged into one object in the 2D-image.

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 (the 3D skin-pixel clusters) and the hypotheses of the past frames, $s_{t-1}, s_{t-2}, \ldots$.

With each new frame, all combinations of the clusters' centroids are evaluated to find the hypothesis s_t that exhibits the highest results with respect the product of the following 3 scores:

– The *observation score* $P(O_t|s_t)$ is a measure for the extent to which s_t matches the observation O_t. $P(O_t|s_t)$ increases with each pixel that complies with the hypothesis, e.g. a pixel showing strong skin-color at a position the hypothesis predicts to be part of the head.
– The *posture score* $P(s_t)$ 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 $P(s_t)$, a model of the human body was built from training data. The model consists of the average height of the head above the floor, a probability distribution (represented by a mixture of Gaussians) of hand-positions relative to the head, as well as a series of constraints like the maximum distance between head and hand.
– The *transition score* $P(s_t|s_{t-1}, s_{t-2}, \ldots)$ is a measure for the probability of s_t being the successor of the past frame's hypotheses. It is higher, the better the positions of head and hands in s_t follow the path defined by s_{t-1} and s_{t-2} (see Fig. 3)[1]. The transition score is set to a value close to zero[2] if the distance of a body part between $t-1$ and t exceeds the limit of a natural motion within the short time between two frames.

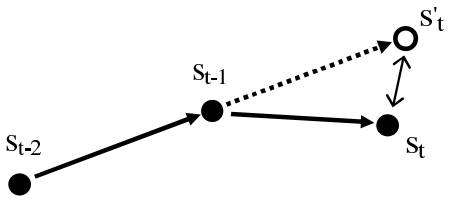


Fig. 3. The transition score considers the distance between the predicted position s'_t and the currently measured position s_t.

2.3 Multi-Hypotheses Tracking

Accurate tracking of the small, fast moving hands is a hard problem compared to the tracking of the head. The assignment of which hand actually being the left resp. the right hand 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 was 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

[1] In our experiments, we did not find strong evidence for a potential benefit of having a more complex motion model (e.g. a Kalman filter) for the movements of the hands.
[2] $P(s_t|s_{t-1}, s_{t-2}, \ldots)$ should always be positive, so that the tracker can recover from erroneous static positions.

is connected to it's predecessor in a tree-like structure. The tracker is free to choose the path, that maximizes overall probability of observation, posture and transition. In order to prevent the tree from becoming too large, we limit both the number n of hypotheses being kept at each frame, as well as the maximum length b of each branch. By setting b e. g. to a value of 15 (which represents 1sec at 15 FPS), it is possible to fix that part of the trajectory that is older than 1sec. This is important, as in a real-time application we do not want to delay the following interpretation of the tracker's output too much, as this would conflict with the responsiveness of the system.

2.4 Head Orientation

Our approach for estimating head-orientation [submitted to FG'04 separately] is view-based: In each frame, the head's bounding box - as provided by the tracker - is scaled to a size of 24x32 pixels. 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. The networks we use have a total number of 1597 neurons, organized in 3 layers. They were trained in a person-independent manner on sample images of rotated heads.

2.5 Results

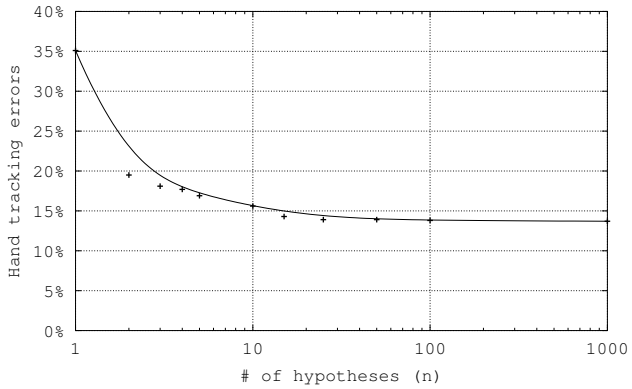


Fig. 4. Percentage of frames with hand-tracking errors in relation to the number of hypotheses per frame (n).

Our experiments indicate that by using the method described, 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 misclassifications. Reasons for this can be temporary occlusions of a hand, a high variance in the visual appearance of hands and the high speed with which people move their hands.

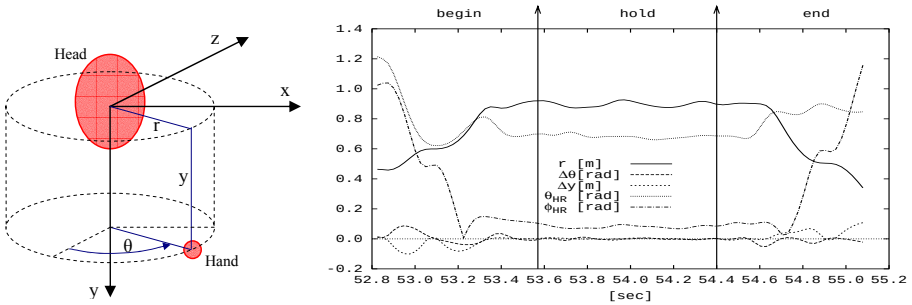


Fig. 5. The hand position is transformed into a cylindrical coordinate system. The plot shows the feature sequence of a typical pointing gesture.

The introduction of multi-hypotheses tracking improves the performance of hand-tracking significantly. Fig. 4 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 0.15cm. Confusing left and right hand therefore counts as two errors.

In our test-set, the mean error of person-independent head orientation estimation was 9.7^o for pan- and 5.6^o for tilt-angle.

3 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.

3.1 Phase Models

When looking at a person performing pointing gestures, one can identify three different 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.

We evaluated pointing gestures performed by 15 different persons, and measured the length of the separate phases (see Table 1). 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 (only 0.1sec), thus potentially showing little evidence in an HMM

which models the complete gesture. So especially with respect to this fact, we train one dedicated HMM for each of the three phases. In addition to that, there is a null-model, that is trained on sequences that are any hand movements but no pointing gestures.

	μ	σ
Complete gesture	1.75 sec	0.48 sec
Begin	0.52 sec	0.17 sec
Hold	0.72 sec	0.42 sec
End	0.49 sec	0.16 sec

Table 1. Average length μ and standard deviation σ of pointing gesture phases. A number of 210 gestures performed by 15 test persons have been evaluated.

3.2 Segmentation

For the task of human-robot interaction we need run-on recognition, meaning that a pointing gesture has to be recognized immediately after it has been performed. So at each frame, we have to search backwards in time for three subsequent feature sequences that have high probabilities of being produced by the begin-/hold-/end-model respectively. The lengths of the sequences to be evaluated vary between $\mu \pm 2\sigma$ according to table 1. The null-model acts as a threshold, such that the phase-models' output must exceed the null-model's output during the course of a gesture. Once a gesture has been detected, its hold-phase is being processed for pointing direction estimation (see section 3.4), and the system is set to sleep for a small amount of time to avoid the same gesture being recognized multiple times.

3.3 Features

We evaluated different transformations of the hand position vector, including cartesian, spherical and cylindrical coordinates[3]. In our experiments it turned out that cylindrical coordinates of the hands (see Fig. 5) 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.

[3] See [16] for a comparison of different feature vector transformations for gesture recognition.

	Head-hand line	Forearm line	Head orientation
Avg. error angle	25°	39°	22°
Targets identified	90%	73%	75%
Availability	98%	78%	(100%)

Table 2. Comparison of three different approaches for pointing direction estimation: a) average angle between the extracted pointing line and the 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.

In our recorded data, we noticed that people tend to look at pointing targets in the begin- and in the hold-phase of a gesture. This behavior is likely due to the fact that the subjects needed to (visually) find the objects at which they wanted to point. Also, it has been argued before that people generally tend to look at the objects or devices with which they interact (see for example the recent studies in [1] and [2]).

In a previous work [3] it has been shown, that using information about head orientation improves accuracy of gesture recognition significantly. While that evaluation has been conducted using a magnetic sensor, we are now using the visual measurements for head orientation. We calculate the following two features:

$$\theta_{HR} = |\theta_{Head} - \theta_{Hand}| \qquad (1)$$
$$\phi_{HR} = |\phi_{Head} - \phi_{Hand}|$$

θ_{HR} and ϕ_{HR} are defined as the absolute difference between the head's azimuth/elevation angle and the hand's azimuth/elevation angle. Fig. 5 shows a plot of all features 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 head orientation.

3.4 Estimation of the Pointing Direction

We explored three different approaches (see Fig. 6) 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. As we did not want this evaluation to be affected by gesture recognition errors, all gestures have been manually labeled.

The results (see table 2) indicate that most people in our test set intuitively relied on the head-hand line when pointing on a target. This is why we suggest the use of the head-hand line for pointing direction estimation and also use this line in all applications of our system.

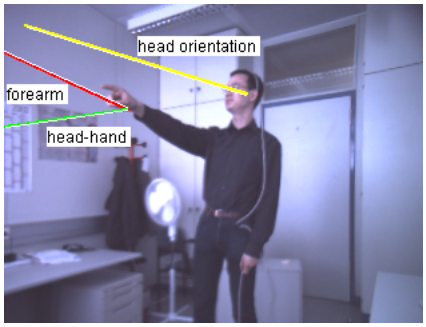


Fig. 6. Different approaches for estimating the pointing direction. (The lines were extracted in 3D and projected back to the camera image.)

	Recall	Precision	Error
Sensor Head-Orientation	78.3%	86.3%	16.8°
Visual Head-Orientation	78.3%	87.1%	16.9°
No Head-Orientation	79.8%	73.6%	19.4°

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

4 Experiments and Results

In order to evaluate the performance of gesture recognition, we prepared an indoor test scenario with 8 different pointing targets. Test persons were asked to imagine the camera was 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.

Our baseline system without head-orientation scored at about 80% recall and 74% precision in gesture recognition (see table 3). When head-orientation was added to the feature vector, the results improved significantly in the precision value: the number of false positives could be reduced from about 26% to 13%, while the recall value remained at a similarly high level.

With head-orientation, also the average error in pointing direction estimation 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.

Although there was 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.

s

5 Conclusion

We have demonstrated a real-time 3D vision system which is able to track a person's head and hands robustly, detect pointing gestures, and to estimate the pointing direction. By following a multi-hypotheses approach in the search for head and hands, 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.

Acknowledgments

References

1. P.P. Maglio, T. Matlock, C.S. Campbel, S. Zhai, and B.A. Smith. Gaze and speech in attentive user interfaces. *Proceedings of the International Conference on Multimodal Interfaces*, 2000.
2. B. Brumitt, J. Krumm, B. Meyers, and S. Shafer. Let There Be Light: Comparing Interfaces for Homes of the Future. *IEEE Personal Communications*, August 2000.
3. Anonymous.
4. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 1997.
5. A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. *Proceedings of 13th ICPR*, 1996.
6. T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998.
7. T. Starner and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. M.I.T. Media Laboratory, Perceptual Computing Section, Cambridge MA, USA, 1994.
8. 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.
9. A.D. Wilson and A.F. Bobick. Recognition and Interpretation of Parametric Gesture. *Intl. Conference on Computer Vision ICCV*, 329-336, 1998.
10. I. Poddar, Y. Sethi, E. Ozyildiz, and R. Sharma. Toward Natural Gesture/Speech HCI: A Case Study of Weather Narration. *Proc. Workshop on Perceptual User Interfaces (PUI98)*, San Francisco, USA. 1998.
11. R. Kahn, M. Swain, P. Prokopowicz, and R. Firby. Gesture recognition using the Perseus architecture. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 734–741, 1996.
12. N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. *IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000.

13. K. Konolige. Small Vision Systems: Hardware and Implementation. *Eighth International Symposium on Robotics Research*, Hayama, Japan, 1997.
14. J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaption. Technical Report of School of Computer Science, CMU, CMU-CS-97-146, 1997.
15. L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 77 (2), 257–286, 1989.
16. L. W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, and A. Pentland. Invariant features for 3-D gesture recognition. *Second International Workshop on Face and Gesture Recognition*, Killington VT, 1996.