Recognition of 3D-Pointing Gestures 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. We present a system capable of visually detecting pointing gestures performed by a person interacting with a robot. The 3D-trajectories of the person's head and hands are extracted from image sequences provided by a stereo camera. We use Hidden Markov Models trained on different phases of sample pointing gestures to detect the occurrence of pointing gestures. For the estimation of pointing direction, we compare two approaches: 1) Using the head-hand line for estimation and 2) estimating the 3D-forearm direction. In a person-independent test scenario, our system achieves a gesture detection rate of 88%. For 90% of the detected gestures, the correct pointing target (one out of eight objects) could be determined.

1 Introduction

It is desirable that robots interacting with humans in natural environments should be able to understand and adequately react to human intentions. This interaction should be governed by the very modalities that are involved in the interaction between humans. While speech recognition plays an important role in this, non-verbal means of communication such as gestures and facial expressions must also be taken into consideration.

In this paper we describe a system that is able to recognize human pointing gestures and to determine their direction, thus opening up the possibility of humans communicating intuitively with robots by indicating objects and locations, e.g. to make a robot change its direction of movement or to simply mark some object. This is particularly interesting in combination with speech recognition as pointing gestures can resolve ambiguities and specify parameters of location in verbal statements ("Put the cup *there*!").

Our system was built to function in natural environments, to allow for movements of the robot, to work in real time and recognize gestures directly as they are performed. The system performs three tasks:

- color- and range-based tracking of head and hands,
- classification of both hands' trajectories by means of previously trained pointing gesture models (HMMs),
- determination of the pointing direction.

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 [1], 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 [2] 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 [3].

Hidden Markov Models (HMMs) have successfully been applied to the field of gesture recognition. In [4], Starner and Pentland were able to recognize hand gestures out of the vocabulary of the *American Sign Language* with high accuracy. Becker [5] presents a system for the recognition of *T'ai Chi* gestures based on head and hand tracking. In [6], Wilson and Bobick propose an extension to the HMM framework, that addresses characteristics of parameterized gestures, such as pointing gestures. Jojic et al. [7] describe a method for the estimation of the pointing direction in dense disparity maps.

2 Tracking of Head and Hands

In order to gain information about the location and posture of a person interacting with a robot, 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.

Our setup (see Fig. 1) consists of a fixed-baseline stereo camera head connected to a standard PC. A commercially available library [8] is used to calibrate the cameras, to search for image correspondence and to calculate 3D-coordinates for each pixel.

Fig. 1. 3D tracking system

a. Color image b. Skin-color map c. Disparity map

Fig. 2. Skin-color map and disparity map for a video frame. 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.

2.1 Combining Color and Range Information

Head and hands can be identified by color as human skin color clusters in a small region of the chromatic color space [9]. To model the skin-color distribution, two histograms of color values are built by counting pixel samples belonging to either the skin-color class S^+ or the *not*-skin-color class S^-. 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. 2.b). To eliminate isolated pixels and to produce closed regions, a combination of morphological operations is applied to the skin-color map.

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

In order to initialize and maintain the skin-color model automatically, we search for a person's head in the disparity map of each new frame. Following an approach proposed in [3], 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.

2.2 Tracking

The task of tracking consists in finding a good hypothesis s_t for the positions of head and hands at time t. The decision is based on the current observation (the 3D skin-pixel clusters) and the hypothesis for the last frame, s_{t-1}.

With each new frame, all combinations of the clusters' centroids are evaluated to find a hypothesis s_t that maximizes the product of the following 3 scores:

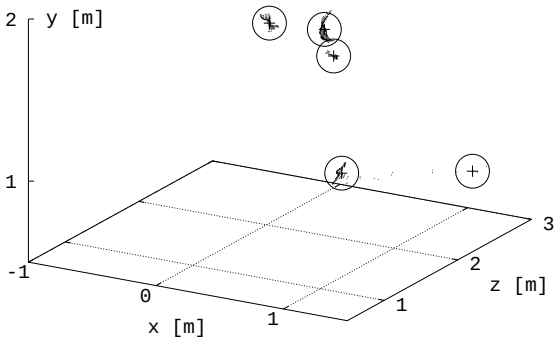

Fig. 3. Skin-colored 3D-pixels are clustered using a k-means algorithm. The resulting clusters are depicted by circles.

- 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* $P(s_t|s_{t-1})$ is a measure for the probability of s_t being the successor of s_{t-1}. It is higher, the closer the positions of head and hands in s_t are to their positions in s_{t-1}. $P(s_t|s_{t-1})$ is set to a value close to zero[1] 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.

2.3 Results

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.

Due to the automatic updates of the skin-color model, the system does not require manual initialization.

[1] $P(s_t|s_{t-1})$ must always be positive, so that the tracker can recover from erroneous static positions.

3 Detection of Pointing Gestures

Intuitively, pointing gestures can be recognized by watching the trajectory of the pointing hand. As mentioned in the previous section, the trajectory provided by the tracking module is affected by measurement noise, discontinuities and mismatches. In the following, we develop a model for pointing gestures - based on the combination of four continuous Hidden Markov Models (HMMs) - that is able to detect the occurrence of a pointing gesture on erroneous trajectories. HMMs have been used for years in continuous speech recognition [10], and have also been applied successfully in the field of gesture recognition (e.g. [4], [5]).

3.1 Features

As mentioned above, the 3D-measurements of the pointing hand are the basis for the HMMs' input features. 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.

We evaluated different transformations of the feature vector, including cartesian, spherical and cylindrical coordinates[2]. In our experiments it turned out that cylindrical coordinates of the hands (see Fig. 4) produce the best results for the pointing task: The radius r represents the distance between hand and body, which makes him an important feature for pointing gesture detection. Unlike the radius in spherical coordinates, r is independent of the hand's height y. The azimuth angle θ lies in the interval $[0, 2\pi)$.

Since we want to prevent the model from adapting 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. The final feature vector is

$$(r, \Delta\theta, \Delta y)\,. \tag{1}$$

[2] See [11] for a comparison of different feature vector transformations for gesture recognition.

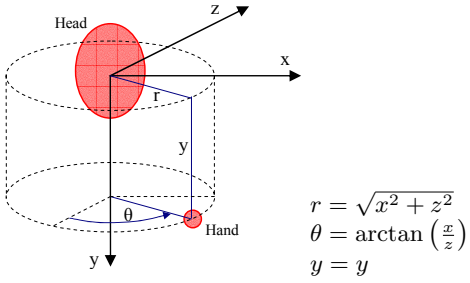

Fig. 4. Cylindrical head coordinate system

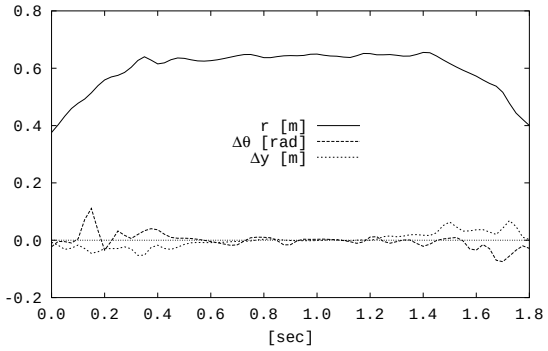

Fig. 5. Feature sequence of a typical pointing gesture

Depending on the performance of the vision system, we capture 10-20 frames per second. To compensate for varying framerates, and to generate measurements that are equidistant in time, we resample the data at a constant rate of 40Hz, using cubic spline interpolation. Fig. 5 shows a plot of the features during the course of a typical pointing gesture.

3.2 Gesture Model

When looking at a person performing pointing gestures, one can easily 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. Compared to the pointing position, the starting position is generally closer to the floor and to the person's body.
- Hold (H): The hand remains motionless at the pointing position.
- End (E): The hand moves away from the pointing position, thereby frequently reversing the path taken in the begin phase, and ending somewhere close to the starting position.

For the task of estimating the pointing direction, it is crucial to locate the hold phase precisely (see Table 1). Therefore, we model the three phases separately: Three dedicated HMMs (M_B, M_H, M_E) were trained exclusively on

	μ	σ
Complete gesture	1.75 sec	0.48 sec
Begin	0.52 sec	0.17 sec
Hold	0.76 sec	0.40 sec
End	0.47 sec	0.12 sec

Table 1. Average length μ of pointing gesture phases. In total, 89 gestures (performed by 10 subjects) were measured. The hold phase shows the highest variance: the observed values range from 0.1 sec to 2.5 sec.

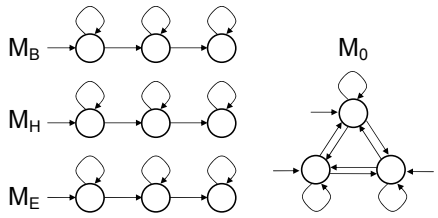

Fig. 6. Left-right HMMs are used for modelling the 3 phases of a pointing gesture, an ergodic HMM represents non-gesture sequences.

data belonging to their phase. We choose the same HMM topology (3 states, left-right) for each of the three models. For each state, a mixture of 2 Gaussian densities represents the output probability.

To get a reference value for the output of M_B, M_H and M_E, we train a *null model* (M_0) on short feature sequences ($0.5sec$) which do *not* belong to any pointing gesture. For M_0, we choose an ergodic HMM with 3 states and 2 gaussians per state. Fig. 6 depicts the different model topologies.

All training sequences were hand-labeled to identify the B-, H- and E-phases. The models' parameters were trained by means of the Baum-Welch reestimation equations [10].

3.3 Classification

As we want to detect pointing gestures on-line and immediately after they have been performed, we have to analyze the observation sequence each time a new frame has been processed.

The length of the B-, H- and E-phase varies from one gesture to another. Therefore, we classify not only one, but a series of subsequences $O_{1..n}$, each one starting at a different frame in the past and ending with the current frame t_0. The lengths n of the subsequences are chosen to be within a range of $\mu \mp 1,645 \cdot \sigma$, thus covering 90% of the sequence lengths observed in the training set (see Table 1).

For each of the models M_B, M_H and M_E, we search for the subsequence that maximizes the probability of being produced by the respective model:

$$O_{B,H,E} = argmax\ logP(O_{1..n}|M_{B,H,E}) \tag{2}$$

$P(O|M_0)$ represents the probability, that the sequence O is *not* part of a pointing gesture. We use it as a threshold, and associate $O_{B,H,E}$ with the respective model, if

$$P_{B,H,E} = logP(O_{B,H,E}|M_{B,H,E}) - logP(O_{B,H,E}|M_0) > 0. \tag{3}$$

Fig. 7 shows a plot of the values of P_B, P_H and P_E over a sequence containing two pointing gestures.

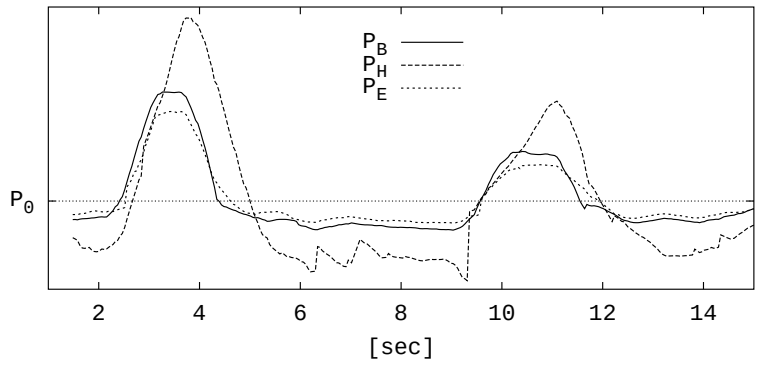

Fig. 7. Output probabilities of the phase-models during a sequence of two pointing gestures

In order to detect a pointing gesture, we have to search for three subsequent time intervals that produce high values for P_B, P_H and P_E. We start the search each time t_E the E-model is stronger than the B-model, then we search backwards for a time t_B where the B-model dominates. If $t_E - t_B$ is inside the reasonable range for a gesture, we continue searching for a time t_H, where the H model is accepted.

In summary, a pointing gesture is detected whenever we find three points in time, $t_B < t_H < t_E$, so that

$$\begin{aligned} P_E(t_E) > P_B(t_E) \wedge P_E(t_E) > 0 \\ P_B(t_B) > P_E(t_B) \wedge P_B(t_B) > 0 \\ P_H(t_H) > 0 \end{aligned} \tag{4}$$

Note: For the calculation of $logP(O_{1..n}|M)$, we use the scaled forward algorithm for HMMs described in [10]. As noted by [5], it is computationally efficient to train and to evaluate the models with *time-reversed* sequences, in order to be able to exploit the recursive nature of the Viterbi algorithm (resp. the forward algorithm). As the reversed sequences all *start* at the same time and *end* at different times, it is sufficient to compute the forward algorithm only once, for the longest sequence. The results for the shorter sequences are just intermediary results of this computation.

4 Estimation of the Pointing Direction

After a pointing gesture has been detected, we have to find out which object or which location has been specified by the gesture. We explored two different approaches (see Fig. 8) to estimate the direction of a pointing gesture: The first approach is based on the assumption that the pointing direction is the extension of the line of sight between the head and the pointing hand. The second approach equates the pointing direction with the direction of the *forearm*.

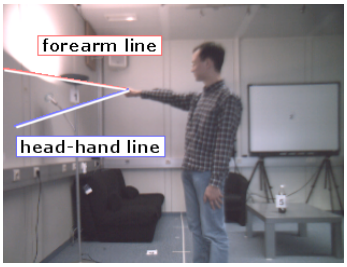

Fig. 8. Extracted pointing direction based on head-hand line resp. forearm line

In order to identify the orientation of the forearm, we calculate the covariance matrix C of the 3D-pixels $x_{1..N}$ within a 20cm radius around the center of the hand μ:

$$C = \frac{1}{N} \sum_{N} (x_n - \mu)(x_n - \mu)^T \tag{5}$$

The eigenvector v^1 with the largest eigenvalue (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 v^1 to be a measure for the direction of the forearm (see Fig. 9). This method assumes that no other objects are present within the critical radius around the hand, as those would influence the shape of the point cloud. We found that in the hold phase this pre-condition is satisfied, because the distance between hand and body as well as between hand and target object is generally sufficient.

Our final estimate of the pointing direction is based on the mean value of all head and hand measurements (resp. forearm measurements) within the hold phase of the respective gesture.

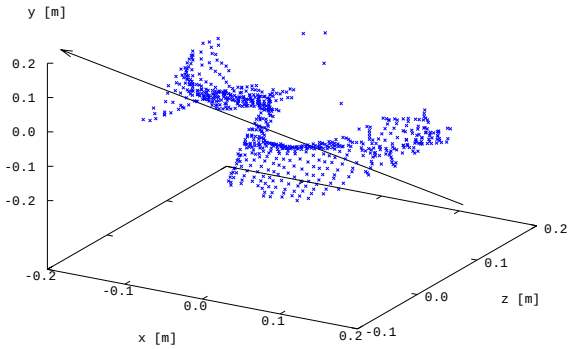

Fig. 9. Estimation of the forearm orientation by means of the first principal component

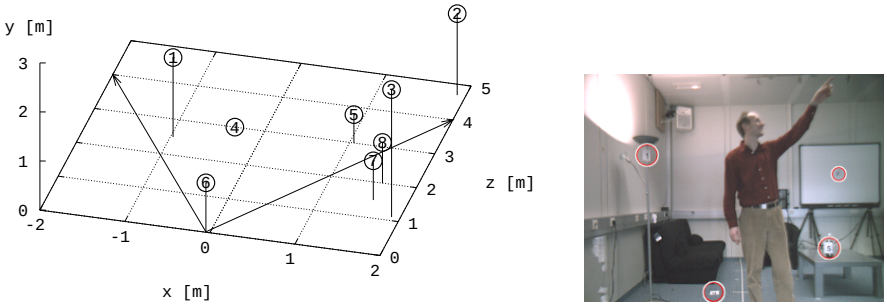

Fig. 10. Test scenario: 8 pointing targets were placed around the room. Target #6 was the camera itself. Arrows depict the camera's field of view.

5 Experiments and Results

In order to evaluate the performance of our system, we prepared an indoor test scenario with 8 different pointing targets. Test persons have been asked to imagine that the tripod-mounted camera was a household robot. They could move around within the robot's field of view, and every now and then show the robot one of the marked objects by pointing on it. (See Fig. 10.)

We captured 25 sequences (24 min of video in total), each showing one of 10 different test persons. The total number of pointing gestures was 280. In 74 cases, a gesture had to be removed from the test set by hand, because either the tracker failed to provide a trajectory for the pointing hand (due to occlusion or general tracking mismatch)[3] or the test person performed two subsequent pointing gestures in one go.

5.1 Pointing Direction

We define two different measures for the accuracy of the extracted pointing direction:

- the angle δ between the extracted pointing line and the ideal line from the pointing hand to the target,
- the percentage of gestures for which the correct target can be identified by choosing the target (1 out of 8) with the lowest δ.

Both the head-hand line and the forearm line have been evaluated using these measures. We used hand-labeled H-phases in order to avoid errors caused by the gesture detection module. Nevertheless, there is an error induced by the stereo vision system, because the camera's coordinates do not comply perfectly with the manual measurements of the targets' positions.

[3] Note that we did only remove a gesture, when at least half of the trajectory was completely wrong or missing.

	Avg. error angle	Target identified
Head-hand line	14.8°	99.1%
Forearm line	42.8°	69.6%

Table 2. Experimental results: a) angle between the extracted pointing line and the ideal line, and b) percentage of gestures for that the correct target (1 out of 8) could be identified.

Table 2 summarizes the results. The good results of the head-hand line indicate, that most people in our test set intuitively relied on the head-hand line, when aiming at a target. The system was able to identify the correct target almost every time.

We believe that the inferiority of the forearm line is mainly the result of erroneous forearm measurements. Unlike the relatively stable head position, the forearm measurements vary strongly during the H-pase. The test persons were pointing with an outstretched arm almost every time, thus reducing the potential benefit even of a more accurate forearm measurement.

5.2 Gesture Detection

Two measures for the quality of the gesture detection were calculated:

- the detection rate (*recall*) is the percentage of pointing gestures that have been detected correctly ,
- the *precision* of the gesture detection is the ratio of the number of correctly detected gestures to the total number of detected gestures (including false positives).

In order to determine the person-independent recognition quality, we used a leave-one-out strategy; i.e, we trained the Hidden Markov models on data from nine subjects and evaluated performance on the remaining tenth person. Altogether, the test data sets contained 89 pointing gestures.

In the person-dependent evaluation, we had five data sets for each of three persons, and we applied the same leave-one-out strategy on the five data sets of each person. In this case the total number of pointing gestures in the test sets was 117.

As in section 5.1, we measured the quality of the extracted pointing direction using the head-hand line. In order to get an impression of the performance of the complete system, we used the automatically detected H-phases instead of hand-labeled ones.

Table 3 shows the average results for the person-dependent and person-independent test sequences. In both cases, we achieve a pointing gesture detection rate of around 88%. While the detection rate is quite similar for both cases, the person-dependent test set has a lower number of false positives compared to the person-independent test set, resulting in a higher classification accuracy. In

	Detection rate (Recall)	Precision	Avg. error angle	Targets identified
person-dependent	88.2%	89.3%	12.6°	97.1%
person-independent	87.6%	75.0%	20.9°	89.7%

Table 3. Evaluation of the quality of pointing gesture detection. The person-independent results are the average results on ten subjects. For the person-dependent case, average results on three subjects are given (see text for details).

addition, estimation of the pointing direction is better in the person-dependent case, resulting in a 97% correctly identified pointing targets. This indicates that it is easier to locate the H-phase correctly when the models are trained individually for each subject. However, even for the person-independent test case, close to 90% of the time the correct target could be identified.

6 Conclusion

We have demonstrated a vision system which can track a person's head and hands in 3D in real time. Robust tracking is achieved in our system by combined use of color and range features. The presented tracking system is also able to detect the occurrence of pointing gestures as well as the 3D pointing direction. We use Hidden Markov Models to detect pointing gestures based on the 3D-trajectories of a user's hands. By using separate Hidden Markov Models for different gesture phases, high detection rates, even on defective trajectories, could be achieved. For the estimation of the pointing direction, we comparatively used the line of sight between head and hands and the estimated forearm direction. We found the line of sight to be a good estimate for the pointing direction. In a person-independent test scenario, our system achieves a gesture detection rate of 88%. For 90% of the detected gestures, the correct pointing target (one out of eight objects) could be determined. With pointing gesture models that were trained individually for different subjects, pointing targets could be correctly identified 97% of the time. The system runs at approximately 10 fps on a 2.8GHz Pentium PC.

Acknowledgements

We would like to thank Christian Fuegen for insightful discussions on the use of HMMs. Also thanks to everybody who participated in our data collection.

This research is partially supported by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588 "Humanoide Roboter".

References

1. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-Time Tracking of the Human Body. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997.
2. Azarbayejani, A., Pentland, A.: Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. Proceedings of 13th ICPR, 1996.
3. Darrell, T., Gordon, G., Harville, M., Woodfill, J.: Integrated person tracking using stereo, color, and pattern detection. IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998.
4. Starner, T., Pentland, A.: Visual Recognition of American Sign Language Using Hidden Markov Models. M.I.T. Media Laboratory, Perceptual Computing Section, Cambridge MA, USA, 1994.
5. Becker, D.A.: 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.
6. Wilson, A.D., Bobick A.F.: Recognition and Interpretation of Parametric Gesture. Intl. Conference on Computer Vision ICCV, 329-336, 1998.
7. Jojic, N., Brumitt, B., Meyers, B., Harris, S., Huang, T.: Detection and Estimation of Pointing Gestures in Dense Disparity Maps. IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, 2000.
8. Konolige, K.: Small Vision Systems: Hardware and Implementation. Eighth International Symposium on Robotics Research, Hayama, Japan, 1997.
9. Yang, J., Lu, W., Waibel, A.: Skin-color modeling and adaption. Technical Report of School of Computer Science, CMU, CMU-CS-97-146, 1997.
10. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE, 77 (2), 257-286, 1989.
11. Campbell, L.W., Becker, D.A., Azarbayejani, A., Bobick, A.F., Pentland, A.: Invariant features for 3-D gesture recognition. Second International Workshop on Face and Gesture Recognition, Killington VT, 1996.