in: Robotics and Autonomous Systems, 58 (2010) 2, pp. 174-185, Elsevier

Robotics and Autonomous Systems 58 (2010) 174–185

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Estimation of pointing poses for visually instructing mobile robots under real world conditions☆

Christian Martin [a,b,*], Frank-Florian Steege [a], Horst-Michael Gross [a]

[a] Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, Ilmenau, Germany

[b] MetraLabs GmbH - Neue Technologien und Systeme, Ilmenau, Germany

ARTICLE INFO

Article history:
Available online 22 September 2009

Keywords:
Human–Robot interaction
Vision
Pose recognition

ABSTRACT

In this paper, we present an approach for directing a mobile robot under real-world conditions into a target position by means of pointing poses only. Because one important objective of our work is the development of a low-cost platform, only monocular vision at web-cam level should be employed. Our previous approach presented in Gross et al. (2006) [1], Richarz et al. (2007) [2] has been improved by several additional processing steps. Finally, a background subtraction technique and a histogram equalization have been integrated in the preprocessing stage to be able to work in environments with structured backgrounds and under variable lighting conditions. Furthermore, a discriminant analysis was used to find the most relevant input features for the pointing pose estimator. The contribution of this paper is, however, not only the presentation of an approach to estimating pointing poses in a demanding real-world scenario on a mobile robot, but also the detailed and evaluative comparison between different image-preprocessing techniques, alternative feature extraction methods, and several function approximators with the same set of test- and training data. Reasonable combinations of the different methods are tested, and for each component on the processing chain the effect on the accuracy of the target estimation is quantized. The approach presented in this paper has been implemented on the mobile interaction robot Horos to determine the performance and estimation accuracy under real-world conditions. Furthermore, we compared the accuracy of our approach with that of humans performing the same estimation task, and achieved very comparable results for the best estimator.

1. Introduction

In recent years, a lot of research work has been done to develop intelligent mobile robot systems, which can interact even with non-instructed users, making the robots suitable for applications in everyday life. Today's robot systems mainly provide a keyboard, a touch-screen or other input devices for getting input from the user. More and more approaches try to integrate speech recognition onto the robot, but a robust speaker-independent speech recognition is still a hard problem. But besides this verbal communication non-verbal communication also plays a very important role in a dialog between humans. To the best knowledge of the authors, only in a few projects have non-verbal communication aspects in an interactive dialog been already successfully integrated on mobile robots. In the work presented in this paper, we show how a basic non-verbal communication can be realized on a mobile robot. More precisely: We want to deal with the problem, of instructing a mobile robot by the use of pointing gestures/poses.

In the field of service robotics, the possibility to command a mobile robot to a certain target position in the environment, plays an important part in interactions. Gestures or poses (sometimes in combination with spoken commands) is a very intuitive way to instruct the robot to do so without the use of certain input devices (e.g. a keyboard or a joystick).

Because one key objective of our research is the development of a low-cost prototype of a mobile and interactive robot assistant, we are especially interested in vision technologies with a very good price-performance ratio. Therefore, in this work only a low-cost frontal camera of our mobile interaction robot Horos (see Section 3.1) was utilized instead of a high-end stereovision system. To compensate the available deficits of this hardware, we were forced to develop more powerful and robust appearance-based recognition algorithms as it was necessary if stereo approaches were used. Against this background, we were

☆ The research leading to these results has received partial funding from the State Thuringia (TAB-Grant #2006-FE-0154) and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216487 (CompanionAble-Project).

* Corresponding address: Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Lab, P.O. Box 10 05 65, 98684 Ilmenau, Germany.
E-mail address: christian.martin@tu-ilmenau.de (C. Martin).
URL: http://www.tu-ilmenau.de/neurob (C. Martin).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.09.013

particularly interested if it would be possible to robustly estimate a target position at the floor from a pointing pose using only inexpensive hardware and monocular images.

Furthermore, we assume, that the following requirements and constraints in Human–Robot interaction are fulfilled:

- In principle during interaction with the robot the user is signaling interest in interaction by orienting his upper body to the robot as is usual in polite human–human communication.
- The user has to face the target point, when the pointing pose is shown. This is also a very intuitive and natural behavior known from interaction with humans, because a pointing pose appears implausible when the head is oriented somewhere else.
- The user points to targets on his left side with his left arm and vice versa for the right side.
- The user only points to targets on the floor, because the pointing pose is to provide a target position for the mobile robot.

Scenarios, where these requirements and constraints are not fulfilled, were considered as special cases and therefore not considered in this work.

Additionally, we do not want to make any particular constraints on the environment. That means, that we have to deal with different lighting conditions and also with a structured and possibly non-static background. Last but not least, the pointing pose estimation should work in real-time.

In [1,2] we presented an approach, which allows one to direct a mobile robot to a certain position by means of such pointing poses. The system presented there was capable of estimating the target point of the pointing pose on the floor with a low error, but could only operate in environments with unstructured background and ideal lighting conditions. Besides, a computation time of 3–4 s was required for the estimation of a single target position. These constraints seriously conflict with the requirements for the usage of this approach in robotic real-world applications. Therefore, in this paper we present several conceptual and methodical improvements on this approach making it possible to estimate the target point of a pointing pose also in highly structured environments with variable lighting conditions in real-time. We use the same set of training and test data for all compared methods and give an overview of the usability of different methods for a demanding real-world-application.

This paper is organized as follows: After the introduction, Section 2 describes the State of the Art in the field of pointing pose estimation. Afterwards, in Section 3 the overall architecture of the system and our experimental platform, the mobile robot platform Horos, is introduced. Section 4 explains, how the pointing poses can be estimated and how our entire system is designed. In Section 5 the experiments and results will be presented. The papers ends with conclusions in Section 6.

2. Related work

In the literature there exists a huge number of publications on video-based human–machine interaction approaches using poses or gestures for instructing a technical system. Against this background, Fig. 1 tries to give a systematical, but non-exhaustive overview of criteria that can be used to describe and classify the known vision-based gesture recognition and pointing pose estimation approaches. They can be distinguished by the camera configuration used and the image quality, the kind of preprocessing (like, e.g., user-background segmentation), the way how features are extracted, encoded and represented, the applied recognition algorithm, the mechanism that triggers the recognition process, and – of course – the application fields in which they are suitable and intended for. Fig. 1 also shows that each approach uses another combination of image-preprocessing, feature-selection and classification/target-approximation. To select the best approach for our envisaged application, we compared different methods of feature extraction and classification. The main contribution of this paper is not only the presentation of a real-time and real-world suitable approach to recognize the target of a pointing pose but also the comparison of different methods and the quantization of the effects of each method.

Up to now, a lot of work has been done focusing on integrating gesture recognition into Man–Machine-Interfaces. However, most of this work concentrates on distinguishing different gestures or creating a command alphabet for robot control.

Rogalla et al. [3], for example, presented a system that classifies hand postures for robot control. They use monocular high-resolution color images and extract a hand contour by means of skin color segmentation. This contour is sampled with a fixed number of sampling points, normalized and Fourier-transformed. The Fourier descriptors represent the feature vector that is classified using a model database and a distance measurement.

Paquin and Cohen [4] also use a skin color segmentation to track the hands and the head of a user. They utilize a neural network based approach to classify the trajectories recorded during the progress of the gesture and are able to recognize nine different robot instruction gestures like "stop" or "forward". Triesch and v.d. Malsburg [5] detect and classify hand postures in monocular images by using Compound Bunch Graphs. No explicit segmentation is needed, since their system can cope with highly complex backgrounds. The features used are the responses of Gabor wavelets and color information at the graph nodes. Hand poses are classified using a distance measure to a model graph, taking into account deformation and scaling.

A major problem of all these approaches is however, that the specific commands of the command alphabet have to be known by the user. Another key problem is, that the direct commanding of a mobile robot to a target position is not possible because the commands can only be used as a kind of remote control and typically only one of these commands can be executed at a time, for example "drive forward", then "drive to the left" and again "drive forward" to direct a robot to a position 30° in front of the starting position.

A much more intuitive and smoother way to direct the robot is through pointing directly at the target position on the ground. There are only a few approaches known that actually try to estimate a pointing direction out of a deictic gesture. All examined approaches use three processing steps: first image preprocessing, second feature extraction and third approximation of the target.

For preprocessing many approaches (Hofemann and Haasch [6], Bennewitz et al. [7], Li et al. [8], Nickel and Stiefelhagen [9], Wilson and Bobick [10] and Hosoya et al. [11]) use skin color detection to determine the position of the pointing hand. In our application with a mobile robot, skin color detection is not suitable as the robot is facing many different backgrounds and lighting conditions. Fig. 2 shows three examples where skin color detection is very difficult or completely fails. Furthermore detecting the position of the hand and the head of the user is not sufficient to estimate the target of the pointing pose with the quality needed for the envisaged application. The reason is that for most users the imaginary line between the fingertip of the pointing hand and the eye of the user does not extend to the target of the pointing pose. Nickel and Stiefelhagen already described this fact in [9] which was confirmed in our tests. To take this fact into consideration, the orientations of the head and the shoulder have to be regarded, which is not possible with skin color detection as the skin of the shoulder is often hidden by clothing. Moreover, skin color detection does only give information about the position but not the orientation of the head.

Hofemann and Haasch [6], Bennewitz et al. [7], Li et al. [8] and Hosoya et al. [11] determine the quality of the pointing

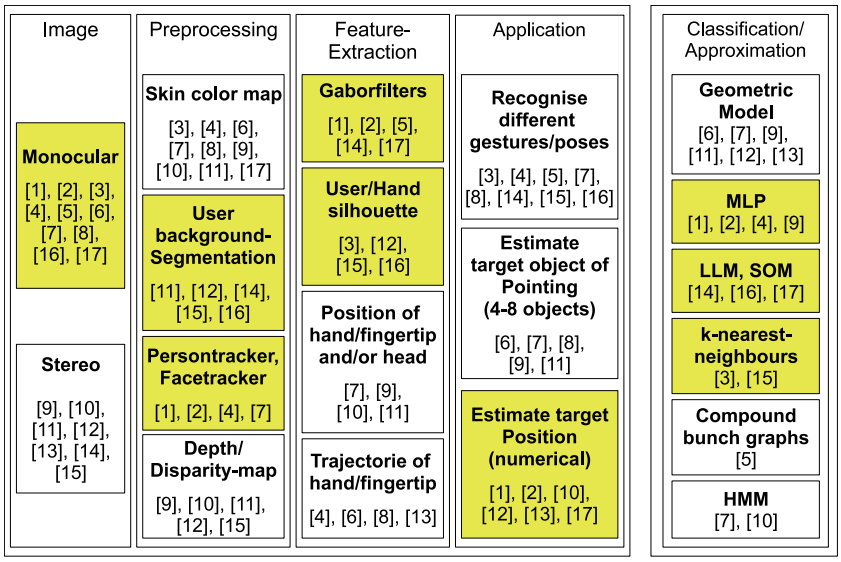


Fig. 1. Systematic overview of significant criteria and aspects to describe and distinguish vision-based gesture recognition and pointing pose estimation approaches. The numbers assigned to each aspect refer to the references and indicate which aspect is employed by which approach. The highlighted boxes describe the aspects we used and compared in our approach presented here.

Fig. 2. Example images from the test dataset captured by a monocular low-cost camera mounted on the interactive mobile robot assistant Horos (see Section 3.1). Skin color detection is very challenging in these images and often fails because of the lighting conditions and the background colors.

pose estimator by pointing at a small number of different target objects (in these cases between 5 and 8). These objects are placed in the space around the pointing person so that for each object the pointing hand has another clearly distinguishable position. In our application the users are pointing at positions lying in the half-space before the pointing user. Only for training and testing 36 different target positions are used, while in the final application of the pointing pose estimator the users can point to any numeric position within the half-space without the need to restrict themselves to discrete objects or positions. As a consequence of this requirement, the position and appearance of the pointing hand for two neighbored targets is only slightly different.

Therefore, most approaches which estimate the numeric target position for purpose of robot instruction or virtual pointing often use stereo cameras and a 3D modeling of the scene as shown in Jojic et al. [12], Nickel and Stiefelhagen [9] or Hung et al. [13]. For our application we could not use stereo vision because of the preconditions specified in the introduction and, therefore, needed a feature extraction which is able to extract detailed features from the head, shoulder and arm of the user allowing an implicit description of the pointing pose without using skin color or stereo vision. In [1,2] we used Gabor filters to extract suited features. A similar approach was used by Nölker and Ritter in [14] to classify pointing poses. The disadvantage of this method is, however, that it could not cope with structured backgrounds or other persons beside the user in the image. In Section 4 we describe the improvements we made to be able to get useful features even with structured backgrounds and additional persons in the image.

Another basic approach to extract features for pointing poses is to separate the user from the background in some way and to use the features of the silhouette to identify the pose. Such a silhouette-based approach is employed by Rogalla et al. [3], Urano et al. [15] and Takahashi and Tanigawa [16]. In Section 5 we give an overview of the results we achieved with this relatively simple method of feature extraction and compare it with the results of the Gabor filter-based one.

After the features are extracted the target of the pointing pose has to be estimated. In most cases, a line is drawn between the head or the eye of the user and the fingertip of the pointing hand. The endpoint of the line on a wall or on a predefined object then is taken as the target of the pointing pose (Hosoya et al. [11], Hung et al. [13], Jojic et al. [12], Nickel and Stiefelhagen [9]), or the object next to a defined region around the pointing hand is referred as the target (Hofemann ans Haasch [6]). We did not investigate these methods in our approach because of the existing diversity of our application, to visually command a robot to any target position in a restricted operation area by pointing, and the significant effects on the target position caused by already slightly different hand positions. Instead in [1,2] we used a cascade of several neural function approximators to estimate the target position. A similar system with one function approximator (a Multi-Layer Perceptron—MLP) is used by Stiefelhagen [17] to estimate the line of sight of a user. Certainly other authors use other neural networks as approximators: Krueger and Sommer [18] employ a Local Linear Map (LLM) to estimate the head-position of a person, Takahashi and Tanigawa [16] a Self-Organizing Map (SOM) to classify different poses, and Paquin and Cohen [4] compare the extracted features with labeled data with a method very similar to a *k*-Nearest Neighborhood method. We implemented those classifiers, that have been successfully used in the different

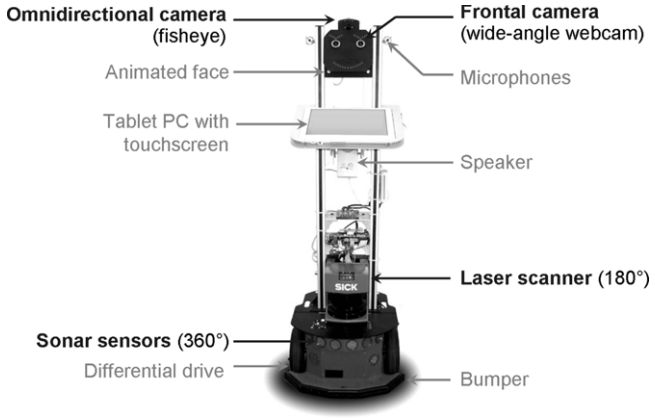


Fig. 3. Mobile service robot Horos used for experimental investigation of the pointing pose estimation. The images for the estimation of the pointing target were taken with the webcam located in the left eye.

aforementioned approaches and compared the results with our function approximator presented in [1,2] to determine the best method for this target position estimation task.

The contribution of this paper is, however, not only the presentation of an approach to estimate pointing poses in a demanding real-world scenario on a mobile robot, but also the detailed and evaluative comparison between different image-preprocessing techniques, alternative feature extraction methods, and several function approximators with the same set of test- and training data. Reasonable combinations of the different methods are tested, and for each component on the processing chain the effect on the accuracy of the target estimation is quantized.

3. Robot platform and system architecture

In the following section our experimental robot platform Horos and the overall architecture of the developed pose estimation system are described.

3.1. Experimental robot platform

The approach described in this paper was developed and tested on our mobile robot Horos (**HO**me **RO**bot **S**ystem). Horos' hardware platform is an extended Pioneer II based robot from ActivMedia. It integrates an on-board PC (Pentium M, 1.6 GHz) and is equipped with a SICK LMS200 laser range-finder and a ring of sonar sensors (see Fig. 3).

For the purpose of HRI, the robot was equipped with different interaction oriented modalities. This includes a tablet PC for touch-based interaction, speech recognition and speech generation. Horos was further extended by a simple robot face which integrates an omnidirectional fisheye camera situated in the center of the head, a camera with a telephoto lens mounted on a tilting socket on the forehead, and a wide-angle camera in one of the eyes.

Horos is controlled by a highly flexible and extensible control architecture described in [19]. The approach described in this paper was implemented in this control architecture framework, which allows one to use other existing modules for our application, e.g. the speech recognition system can be used as a trigger signal ("Horos, go there!") for the start of the estimation for the target point.

3.2. Overall architecture

Fig. 4 shows the overall architecture of our system. A multimodal person tracker [1,2] is utilized to determine the direction ϕ_{user} and the distance d_{user} of the user to the robot. When the user wants to use a pointing pose to direct the robot, he can trigger the estimation by means of a voice command.

The feature extraction estimates the radius r_{pose} and the angle ϕ_{pose} of the pointing pose in a user-centered polar coordinate system (see Section 4.1). With the tracking result from the person tracker and the estimated radius and angle from the pointing pose estimator, the referred goal point $(x_{\text{goal}}, y_{\text{goal}})$ on the ground can be computed in a local, robot-centered coordinate system. Given the current pose of the robot $(x_{\text{robot}}, y_{\text{robot}}, \phi_{\text{robot}})$, the local goal point can be translated in the world coordinate system of the environment model. This enables the robot to move to the referred target point avoiding obstacles during the movement by means of the navigation module.

To embed the estimation process in an interactive dialog, a speech recognition module can be used as a trigger signal. A first speech command (e.g. "Horos") is used to start the estimation process, while a second command (e.g. "Go there!") is utilized to finish the process and start the autonomous movement of the robot. Additionally, an interrupt command (e.g. "Stop!") enables the user to interrupt and stop the movement of the robot.

4. Estimation of pointing poses

In the following section the estimation of the pointing pose based on monocular images is explained in detail.

4.1. Training-data and ground-truth

To develop the Pointing Pose Estimator, a labeled set of images of subjects pointing to target points on the floor was required to train the system. We encoded the target points on the floor as (r, φ) coordinates in a subject-centered polar coordinate system (see Fig. 5) and placed the robot with the camera in front of the subjects. Moreover, we limited the valid area for targets to the half space in front of the robot with a value range for r from 1 to 3 m and a value range for φ from $-120°$ to $+120°$. The 0° direction is defined as the user-robot-axis, negative angles are on the user's left side. With respect to a predefined maximum user distance of 2 m, this spans a valid pointing area of approximately 6 by 3 m on the floor in front of the robot in which the indicated target points may lie.

Fig. 5 shows the configuration we chose for recording the training data. The subjects stood at distances of 1, 1.5 and 2 m from the robot. Three concentric circles with radii of 1, 2 and 3 m are drawn around the subject, being marked every 15°. Positions outside the specified pointing area are not considered. The subjects were asked to point to the markers on the circles in a defined order and a monocular image was recorded by the frontal camera of the robot each time. Pointing was performed as a defined pose, with outstretched arm and the user fixating the target point (see Fig. 5, right). All captured images are labeled with distance, radius and angle, thus representing the ground truth used for training and for the comparing experiments with human viewers (see Section 5). This way, we collected a total of 2340 images of 26 different interaction partners (90 different poses for each subject). This database was divided into a training subset and a validation subset containing two complete pointing series (i.e. two sample sets each containing all possible coordinates (r, φ) present in the training set). The latter was composed from 7 different persons and includes a total of 630 images. This leaves a training set of 19 persons including 1710 samples.

4.2. Architecture of the pointing pose estimator

Based on the overall architecture (see Section 3.2 and Fig. 4) the pointing pose estimator uses the image of the frontal camera of our robot Horos.

For the pointing pose estimation process (see Fig. 6) a face detection system [20] is used to find the position of the head

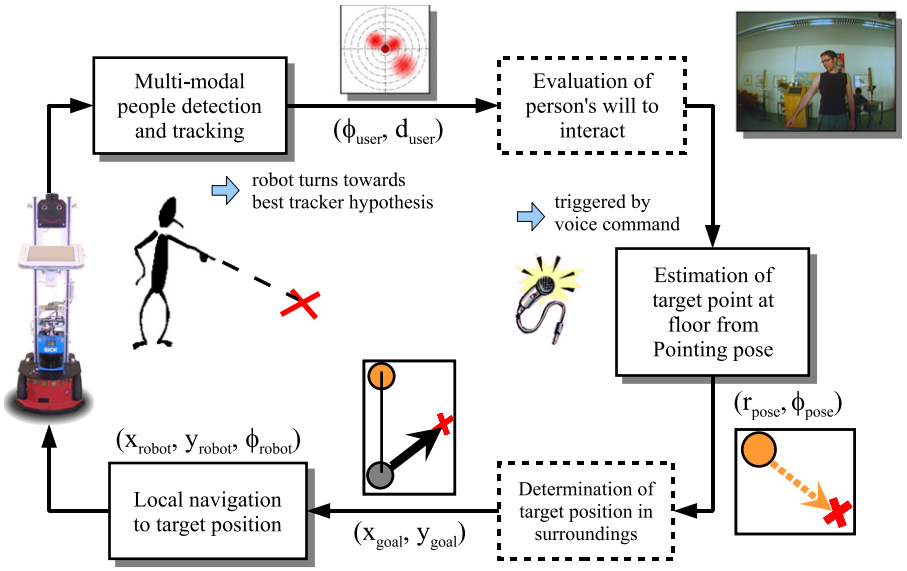


Fig. 4. Overall architecture of the developed on-board target position estimator. The system mainly consists of a multimodal person tracker, the estimator of the pointing pose, and the local navigation system.

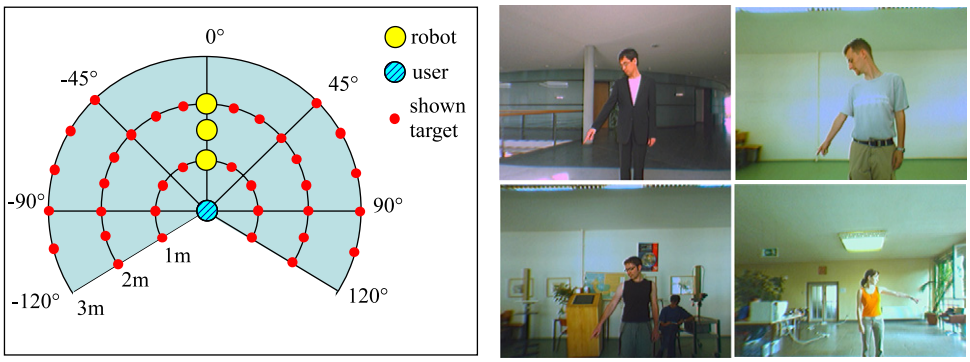


Fig. 5. The left image shows the configuration used for recording the ground truth training and test data: The subjects stood in front of the robot and pointed at one of the marked targets on the ground. The distance of the robot to the subject varied between 1 m and 2 m. The images on the right show typical examples of images of subjects taken by the monocular frontal camera of the robot in several demanding real-world environments with background clutter and different lighting conditions (in contrast to earlier approaches of us presented in [1,2]).

(x_{head}, y_{head}) of the user in the image. The output of the multimodal person tracker (see Fig. 4) is utilized to determine the direction ϕ_{user} and the distance d_{user} of the user to the robot. These data are processed to extract the primary region of interest (ROI) in the input image for the subsequent feature extraction.

The estimation of the radius r_{pose} and the angle ϕ_{pose} of the pointing pose is done in the user-centered polar coordinate system shown in Section 4.1.

The Gabor-filtered primary ROI is first fed into the "Left/Right-classifier". The result of this classifier enables one to extract the finer image ROIs of the head and the arm of the user. In the following stage the final pointing radius r_{pose} is estimated by the "Radius estimator". The estimation of the pointing angle ϕ_{pose} can be realized in two different versions: In one version, first a coarse angle is estimated by means of the Gabor-filtered ROIs and the output of the "Radius estimator". The result of the "Coarse angle classifier" is fed into a "Fine angle classifier", which estimates the final pointing angle ϕ_{pose}. In a simpler version (not shown in Fig. 6), the pointing pose angle ϕ_{pose} is estimated in one step by a single "Angle estimator".

4.3. Image preprocessing and feature extraction

Since the interaction partners standing in front of the camera can have different body height and distance, an algorithm had to be developed that can calculate a normalized region of interest, resulting in similar subimages for subsequent processing. We use an approach suggested in [1,2] to determine the region of interest (ROI) by using a combination of face-detection (based on the Viola & Jones Detector cascade [20]) and some empirical factors. With the help of a multimodal tracker [1,2] implemented on our robot, the direction and the distance of the robot to the interacting person can be estimated. The cropped ROI is scaled to 160 × 100 pixels for the body and the arm and 160 × 120 pixels for the head of the user. Additionally, a histogram equalization is applied to improve the feature detection under different lighting conditions. The preprocessing steps used to capture and normalize the image are illustrated in Fig. 7. To reduce the effects of different backgrounds, in the improved version of our system, we used a simple background subtraction algorithm. For that, the difference image between the start command ("Horos") and the second command ("Go there!") is computed and post-processed with a closing algorithm and a search for connected regions [21] (see Fig. 8). The influence of the background subtraction on the pose estimation result was tested in comparison with our approach in [1,2] where no background subtraction was used (see Section 5). On the normalized image regions, features were extracted to approximate the pointing pose of the user. In our work, Gabor filters of different orientations and frequencies,

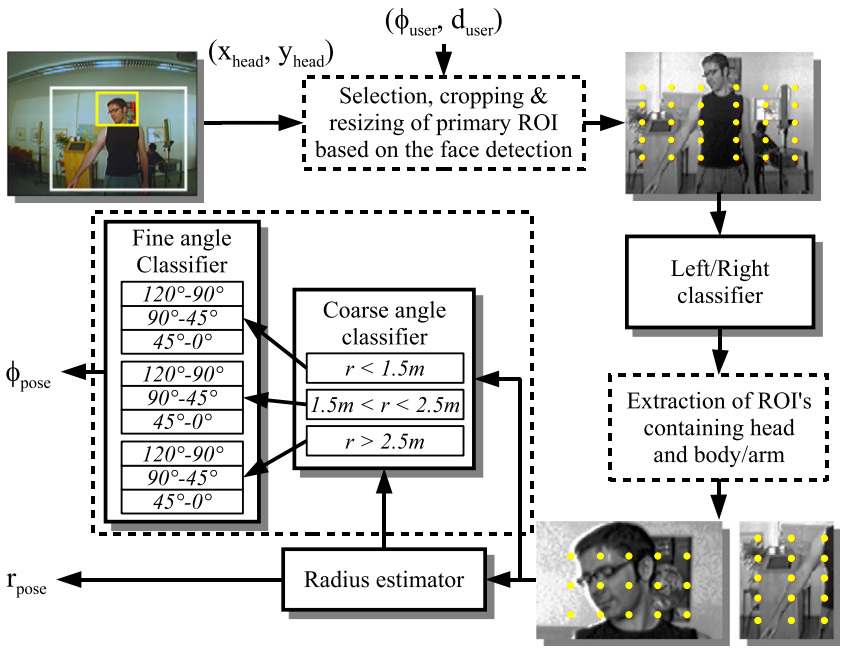


Fig. 6. Architecture overview of the Pointing Pose Estimator. Based on the face detection a primary ROI is extracted. The Gabor-filtered ROI is first fed in to "Left/right-classifier". The result of this classifier enables to extract the finer images ROIs of the head and the arm. In the following stage the final pointing radius r_{pose} is estimated by the "Radius estimator". The pointing angle ϕ_{pose} is estimated in two steps: First a "Coarse angle classifier" and second a "Fine angle classifier" is applied.

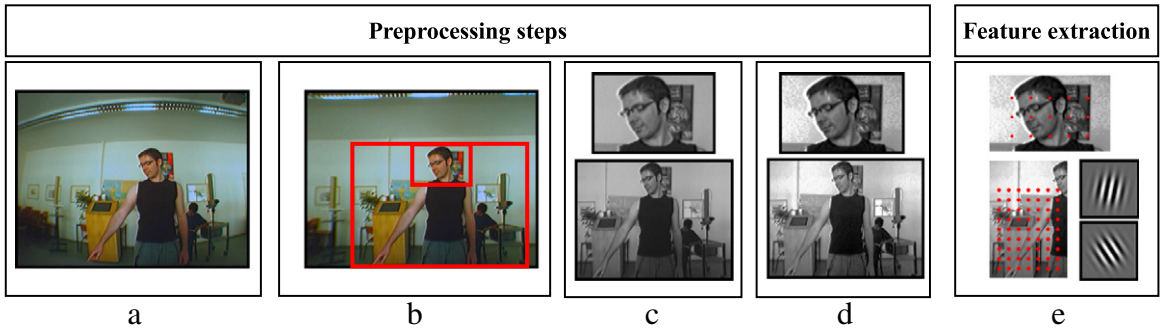


Fig. 7. Steps of preprocessing and feature extraction: the raw distorted image of the low cost camera in the robot's eye (a) is transformed into an undistorted image, and the face of the user is detected by means of [20] (b). Based on the height of the face in the picture and the distance of the user given by the person tracker, two sections of the image are captured and transformed into grayscale images (c). On these images a histogram equalization is applied (d). Subsequently, distributed features are extracted by Gabor filters placed at pre-defined points of the image (marked as red dots in (e)). A background subtraction (see Fig. 8) was optionally used between steps (d) and (e).

bundled in Gaborjets that are located on several fixed points in the selected ROIs, are used. The several steps of preprocessing and feature extraction applied in our comparison are summarized in Fig. 7.

A second feature extraction we used is the histogram of the user-silhouette as proposed by Takahashi and Tanigawa in [16]. This method also uses a background subtraction to separate the user from the background. Afterwards the algorithm counts the pixels, which belong to the silhouette of the user, for each line and column of the image. The number of pixels in each line and column is used as feature for the approximation of the target. Fig. 9 shows a sample histogram for a pointing pose from our dataset. We compare the results we achieved with this feature extraction to the results achieved with the Gabor filters in Section 5.

4.4. Feature selection by discriminant analysis

The discriminant analysis [22,23] is a well-known technique to figure out the most relevant features in a feature space for the separation of two or more classes. In our approach, we used the discriminant analysis for two purposes: First, to achieve a higher robustness against cluttered backgrounds and, second to reduce the required computation time due to the reduced effort for feature extraction.

To determine the importance and the contribution of a single feature k on the estimation of a target position, the following simple feature selection was applied: First, the Gabor filter answers for the selected feature were computed at all samples of the training data set mentioned in Section 4.1. Every value was assigned to a certain class r which was defined through the target point the subject pointed to in the current sample. Then, for feature k the discriminant value $\sigma_{rs}^{(k)}$ between two arbitrary classes r and s was computed as follows:

$$\sigma_{rs}^{(k)} = \frac{\left(\overline{b_r^{(k)}} - \overline{b_{rs}^{(k)}}\right)^2 + \left(\overline{b_s^{(k)}} - \overline{b_{rs}^{(k)}}\right)^2}{\sum_{i\in r}\left(b_i^{(k)} - \overline{b_r^{(k)}}\right)^2 + \sum_{j\in s}\left(b_j^{(k)} - \overline{b_s^{(k)}}\right)^2}. \quad (1)$$

$b_i^{(k)}$ is the Gabor filter answer for the sample i belonging to the class r. $\overline{b_r^{(k)}}$ is the mean filter answer of all samples for the feature k in class r. $\overline{b_{rs}^{(k)}}$ is the mean filter answer of all samples assigned to a certain class r or s. The discriminant value $\sigma_{rs}^{(k)}$ gets a high

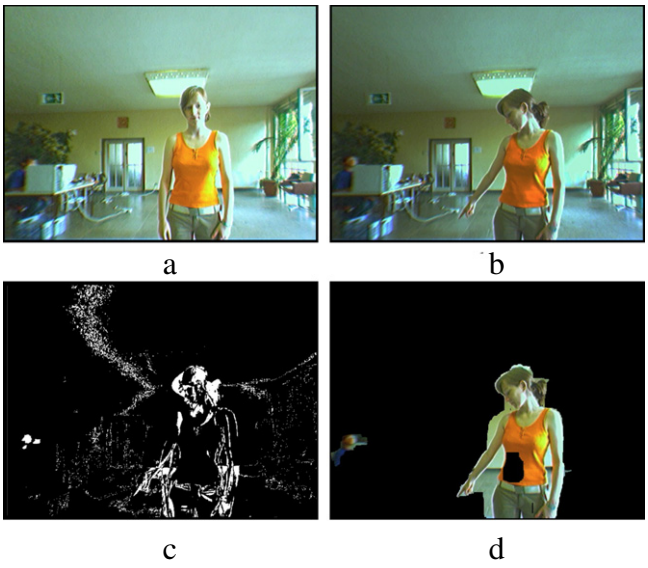


Fig. 8. The background subtraction employed in this approach. (a): by the use of a command word (for example the name of the robot Horos) the user triggers the capturing of a new background image. When the user is pointing at the target, the current image (b) is subtracted from the background image resulting in a difference image (c). With the help of a closing algorithm and the search for connected regions [21] the image is post-processed resulting in an image with the segmented user (d).

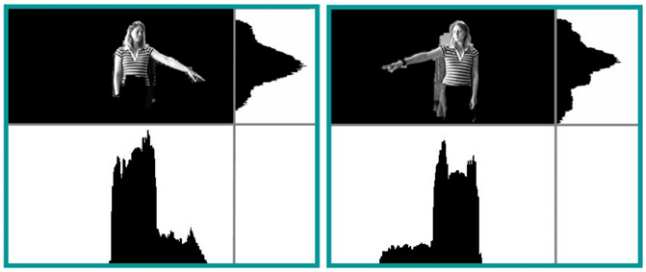


Fig. 9. Feature extraction as proposed by Takahashi and Tanigawa in [16]. The number of human-pixels in each line and column is used as feature for the approximation of the target.

value if the samples of each class have a little intra-class variance (the denominator) and if the different classes do not overlap (the inter-class variance given in the numerator). The results of Eq. (1), applied for all combinations of two classes r and s, were summed up resulting in a single discriminant value for the feature k. Fig. 10 shows the discriminant values for selected features. Gabor filters with high discriminant values directly correspond to the possible alignments of the pointing arm, while features with low values correspond to Gabor filter positions and/or orientations which are not associated with the appearance of a pointing arm but with objects or structures in the background of the picture (clutter). By extracting only those features showing high discriminant values and ignoring features with low discriminant values, we achieved higher robustness against cluttered background and a considerable faster computation since fewer Gabor filter features had to be determined.

4.5. Approximation of the target point

In [1,2] a cascade of several Multi-Layer Perceptrons (MLP) was used to estimate the target point from the extracted features as a regression task. Other techniques are also often used for the estimation of certain human poses, however, till now not on mobile robots but under predefined observation conditions in stationary

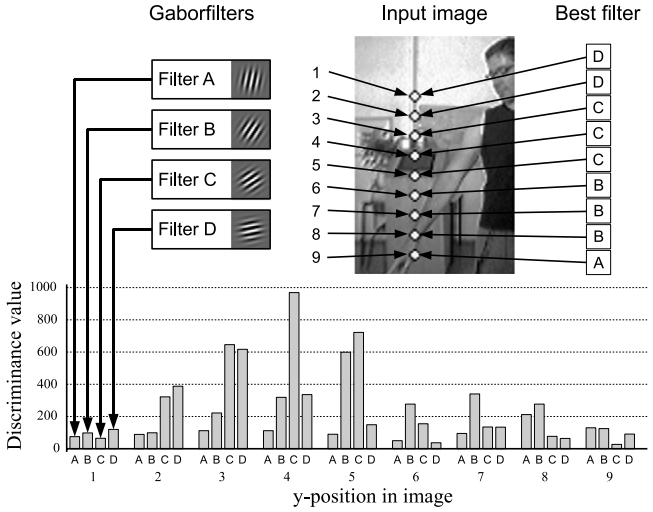


Fig. 10. Determination of important features with the help of a discriminant analysis: the bar chart shows the discriminant values of the Gabor filter features (A to D, shown top left) for each of the fixed filter points in one vertical line in the image. Top right for the respective position, the filter with the highest discriminant value is displayed. Obviously, filters with high discriminant values directly correspond to the possible orientations of the pointing arm of the subjects.

scenarios. Nölker et al. [14] used a Local Linear Map (LLM) and a Parameterized Self-Organizing Map (PSOM) to estimate the target of a pointing pose on a screen the user is pointing to. In [18] Gabor filters and a LLM are utilized to estimate the head pose, while Stiefelhagen [17] presented a stationary system that works on edge-filtered images and uses a MLP for head pose estimation. To give an overview of the suitability of different approaches for the task of estimating a pointing pose from a monocular image, we implemented and compared several relevant approaches, which all were trained and tested with the same sets of training and test data (see Section 4.1). Therefore, for evaluation of the different approaches, all obtained results can be directly compared with each other. In the following paragraphs the different approaches used for comparison are presented briefly:

k-Nearest-Neighbor classification: The k-Nearest-Neighbor method (k-NN) is based on the comparison of features of a new input with features of a set of known examples from the training data. A distance measure is used to find the k nearest neighbors to the input in the feature space. The label that appears most often at the k neighbors is mapped on the new input. This method only allows classification but not a regression, e.g. by approximation between the labels of two or more neighbors. Therefore, we slightly modified the method in our approach. Now the label for the input $f_k(\mathbf{x})$ is determined as follows:

$$f_k(\mathbf{x}) = \sum_i l_i \cdot \left(\frac{1/d_i}{\sum_j 1/d_j} \right). \tag{2}$$

This way the labels l_i of the k nearest neighbors contribute to the output and are weighted with their Euclidian distance d_i to the input $\mathbf{x}$. For example: if an input has the same distance to a sample with the label 45° and a sample with the label 60° the output will be 52.5°. This way the k-NN method can be used for the approximation of targets and not only for classification.

Neural Gas: A Neural Gas network (NG, [24]) approximates the distribution of the input data in the feature space by a set of adapting reference vectors (neurons). The reference vectors $\mathbf{w}_i$ of the neurons are adapted independently of any topological arrangement of the neurons within the neural net. Instead, the adaptation steps are affected by the topological arrangement of the

receptive fields within the input space, which is implicitly given by the set of distortions $D_x = \{\|\mathbf{x} - \mathbf{w}_i\|, i = 1, \ldots, N\}$ associated with an input signal x. Each time an input signal $\mathbf{x}$ is presented, the ordering of the elements of the set D_x determines the adjustment of the synaptic weights $\mathbf{w}_i$. In our approach, each neuron also has a label l_i which is adapted to the label of the input signal, like the k-NN approach described above. After the training of the Neural Gas, the output (the estimated target point (r, φ)) is computed by weighting the labels l_i of the best-matching and the k subsequent neurons with the Euclidian distance d_i to the input x (see Eq. (2)).

Self-Organizing Map: An approach very similar to the NG is the well known Self-Organizing Map (SOM, [25]). The SOM differs from the NG in the fact that the neurons of the SOM are connected in a fixed topological structure. The neighbors of the best-matching neuron are determined by their relation in this structure and not by their order in the set D_x. We modified the SOM so that every neuron also has a learned label (similar to LVQ). The estimated target point (r, φ) is computed in the same way as in the Neural Gas approach, with the exception, that in the case of the SOM the best-matching neuron and the local neighbors in the topological structure are used.

Local Linear Map: The Local Linear Map (LLM, [26]) is an extension of the Self-Organizing Map. The LLM overcomes the discrete nature of the SOM by providing a way to approximate values for positions between the nodes. A LLM consists of n nodes representing a pair of reference vectors $(\mathbf{w}_i^{\text{in}}, \mathbf{w}_i^{\text{out}})$ in the in- and output-space and an associated linear mapping $\mathbf{A}_i$ which is only locally valid. The answer $\mathbf{y}_{\text{bm}}$ of the best-matching neuron of the LLM to an input $\mathbf{x}$ is calculated as follows:

$$\mathbf{y}_{\text{bm}} = \mathbf{w}_{\text{bm}}^{\text{out}} + \mathbf{A}_{\text{bm}} \left(\mathbf{x} - \mathbf{w}_{\text{bm}}^{\text{in}}\right). \tag{3}$$

The weights and the mapping matrix are also learned during the training process. For the estimation of the target (r, φ), we used two separate LLMs:

$$r_{\text{bm}} = \mathbf{w}_{\text{bm}}^{r,\text{out}} + \mathbf{A}_{\text{bm}}^{r} \left(\mathbf{x} - \mathbf{w}_{\text{bm}}^{r,\text{in}}\right) \tag{4}$$

$$\varphi_{\text{bm}} = \mathbf{w}_{\text{bm}}^{\varphi,\text{out}} + \mathbf{A}_{\text{bm}}^{\varphi} \left(\mathbf{x} - \mathbf{w}_{\text{bm}}^{\varphi,\text{in}}\right). \tag{5}$$

Multi-Layer Perceptron: For our experimental comparison, we also used a cascade of several MLPs as described in [1,2]. The different MLPs are trained with a standard backpropagation algorithm. The (r, φ) coordinates of the target point are estimated by separate MLPs. The radius r is estimated by a single MLP while φ is determined by a cascade of MLPs which first estimate a coarse angle φ' and second the final angle φ depending on r and φ'.

5. Experiments and results

We divided the experiments into two groups. At first, we tested the different function approximators with the test data, which were recorded with the subjects described in Section 4.1. These tests were used to indicate which function approximator is best suited for the problem of estimating the target point of a pointing pose given the same feature extraction and preprocessing techniques. Second, we tested the capability of the estimation system on the robot with the best function approximator. This way, we can measure, how much the estimation error of the pose estimator on the test data is increased by real-world influences, like the odometry error of the robot or the detection error of the face detector.

To have the best possible reference for the quality of the estimation, 10 human subjects were asked to estimate the target point of a pointing pose on the floor. At first, the subjects had to estimate the target on a computer screen where the images of the training data set were displayed. The subjects had to click on the screen at the point where they assumed the target to be. Thus, the subjects were estimating the target in the images having the same conditions as the different technical approaches. Second, we determined the estimation result the subjects achieved under real-world circumstances. Here, each subject had to point at a target on the ground, and a second one had to estimate the target. At first the recognizing person used both of their eyes to estimate the target, later we blindfolded one of the eyes, and the person estimated the target again under monocular conditions. The results of the human based reference experiments are illustrated in Table 1 (right). The label *Human (screen)* refers to the experiments on the computer screen and the labels *Human* (2 *eyes*) and *Human* (1 *eye*) refer to the results under real-world conditions.

The results of the several technical approaches for estimating the target position are also shown in Table 1. As described in Section 4.1, for each pointing pose of the ground truth data set, the target radius r and angle φ of the pointing pose was recorded. The separate results for the estimation of r and φ are shown in Table 1(a) and 1(b). For the correct estimation of the target point, r as well as φ had to be estimated correctly. We defined the estimation result to be correct if r differed less than 50 cm from the ground truth radius and φ differed less than 10° from the ground truth angle. Table 1(c) shows the results for a correct estimation of both values.

Each of the five selected approaches described in Section 4.5 was trained on the same training data set and tested on the same test data set. For each system, we used five different feature extraction strategies: first only Gabor filters were utilized, second we combined Gabor filters with an additional background subtraction to reduce the effects of the different cluttered backgrounds in the images. Third, we used only those Gabor filters that had a high discriminant value extracted by means of the discriminant analysis executed over all predefined Gabor filter positions (see Section 4.4). Fourth, we combined Gabor filter, background subtraction and utilized only the relevant features extracted by the discriminant analysis. Fifth we used the histogram features generated by the silhouette of the person as described in Section 4.3. The obtained results show, that the background subtraction and also the discriminant analysis significantly improve the classification results under real-world conditions. The histogram features of the silhouette of the person in most cases produce better results then Gabor filter-features without background subtraction. But, if a background subtraction is applied, then the Gabor filters produce better results. The best results are achieved with a combination of Gabor filters, background subtraction and discriminant analysis.

The results also demonstrate, that a cascade of several MLPs as proposed in [1,2] is best suited to estimate the target position of a user's pointing pose on monocular images. A background subtraction and the information delivered by a discriminant analysis can be used to significantly improve the results for all different classifier systems. The usage of these two algorithms, combined with the histogram equalization in the preprocessing step, now also allows one to handle background clutter and different lighting conditions, which was not possible in our previous work. The best system is capable of estimating the radius r as good as humans with their binocular vision system in a real-world environment and even better than humans estimating the target on a 2D computer screen. The estimation of φ does not reach comparably good values. The system is able to reach a result equal to that of humans on 2D screens or humans with one eye blindfolded, but it is not able to estimate the angle as well as humans in a real-world setting using both eyes. This can be explained, because the estimation of the depth of a target in a monocular image is difficult for both, human and function

Table 1
The results for the estimation of the target point of the pointing pose. The target point is determined by the radius r and the angle φ. Table (a) and (b) show the separate results for the estimation of r and φ. For each method the percentage of the targets estimated correctly and the mean error is determined. Table (c) shows the results for the correct estimation of both values r **and** φ. The results of the human viewers (on computer screen, and in reality (with both eyes "Human (2 eyes)" and with one eye blindfolded "Human (1 eye)")) are given for comparison. Methods that achieve a result comparable to that of the human viewers are marked with a shaded background with different colors. GF = "Gabor filters", BGS = "Background Subtraction", DA = "Discriminant Analysis", HoS = "Histogram of Silhouette".

	k-NN	NG	SOM	LLM	MLP	
(a) Correct estimation of radius; Correct samples in %; Mean error in (m)						
GF	48.2%	33.9%	42.9%	54.4%	70.5%	Human (2 eyes)
	0.31 m	0.46 m	0.44 m	0.38 m	0.24 m	84.3%
GF + BGS	64.8%	65.1%	65.3%	77.7%	**88.2%**	0.08 m
	0.25 m	0.29 m	0.24 m	0.28 m	0.13 m	Human (1 eye)
GF + DA	60.2%	48.5%	56.3%	64.9%	74.4%	75.2%
	0.29 m	0.32 m	0.33 m	0.34 m	0.22 m	0.10 m
GF, BGS + DA	**82.8%**	74.2%	79.3%	**84.2%**	**88.4%**	Human (screen)
	0.12 m	0.21 m	0.19 m	0.23 m	0.14 m	75.0%
HoS + BGS	64.6%	51.1%	45.9%	70.4%	77.0%	0.35 m
	0.31 m	0.40 m	0.49 m	0.36 m	0.20 m	
(b) Correct estimation of angle; Correct samples in %; Mean error in °						
GF	23.1%	13.9%	15.6%	21.6%	41.39%	Human (2 eyes)
	23.0°	23.2°	23.6°	21.8°	18.5°	74.7%
GF + BGS	34.4%	27.7%	23.5%	30.3%	50.9%	4.5°
	20.3°	21.4°	20.91°	18.8°	17.2°	Human (1 eye)
GF + DA	29.4%	19.4%	20.7%	24.7%	37.8%	56.2%
	23.1°	22.2°	23.4°	23.8°	21.0°	7.4°
GF, BGS + DA	41.9%	30.6%	29.9%	37.7%	57.3%	Human (screen)
	17.5°	20.5°	21.0°	19.6°	15.6°	50.0%
HoS + BGS	35.5%	28.6%	23.9%	40.7%	51.0%	13.7°
	18.3°	17.4°	19.4°	15.5°	13.8°	
(c) Combined estimation; Correct samples in %						
GF	11.1%	4.7%	6.7%	11.8%	29.2%	Human (2 eyes)
GF + BGS	22.3%	17.7%	15.3%	23.5%	44.9%	62.9%
GF + DA	17.7%	9.4%	11.6%	16.0%	28.1%	Human (1 eye)
GF, BGS + DA	34.7%	22.7%	23.7%	31.7%	50.6%	40.8%
HoS + BGS	22.9%	14.6%	11.0%	28.6%	39.3%	Human 37.5%

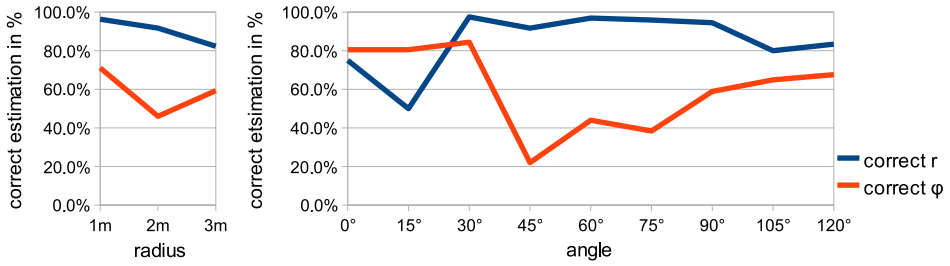


Fig. 11. Relation between the correct estimation of r and φ with the best approximator and the r and φ shown by the subject.

approximators. Fig. 11 shows the correlation between the correct estimation of r and φ the subject was pointing to and vice versa for the best system. The best results for the estimation of the radius are obtained if the subject is pointing to a target within an angle of 30° to 90°. The best results for the estimation of the angle are obtained within an angle of 0° to 30° and within 105° to 120°. This distribution of the estimation error is very similar to the error-distribution human subjects achieve if they estimate the target on a 2D screen or with one eye blindfolded.

The implemented Pointing Pose Estimator is able to run in real-time. The total computation time (on an Athlon XP 2800 CPU) with background subtraction and discriminant analysis was 38 ms for the NG, 42 ms for the SOM, 35 ms of the LLM and 31 ms of the MLP cascade. The k-NN classifier requires 129 ms.

For every approximator we tested different configurations with different numbers of neurons and hidden-layers. In Table 2 the configurations that produced the best results are listed for each approximator.

After selecting the MLP as the best function approximator for our task and testing it under real world conditions, we made additional inquiries about the influence of certain factors on the task of pointing pose recognition. At first, we tested the influence

Fig. 12. (a) A person pointing at a specific location in a static background situation. (b) The same image with background subtraction applied. (c) The same person pointing at the ground with a second person in the background waving with the hand and walking around. (d) The same image with background subtraction applied.

Table 2
Architecture used by the different approximators. 69-20-10-2 means a MLP wih 69 input neurons, two layers of hidden neurons with 20 and 10 neurons and one output layer with 2 neurons.

Approximator	Target	Architecture
MLP	Left–right	69-20-10-2
MLP	Radius	204-50-20-1
MLP	Coarse angle	204-60-30-3
MLP	Fine angle	204-40-20-1
Neural-Gas	Left–right	100
Neural-Gas	Radius, angle	2000
SOM	Left–right	400×1
SOM	Radius, angle	25×25
LLM	Left–right	400×1
LLM	Radius, angle	15×15

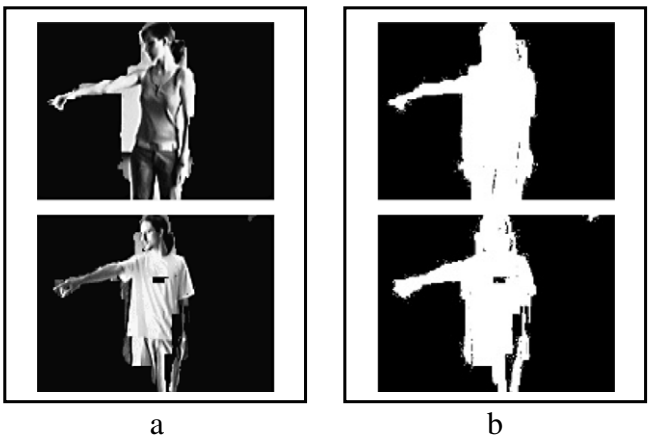


Fig. 13. Left: Input for the pointing pose estimation system with texture. Right: Input for the pointing pose estimation system without texture information.

of movements in the background of the pointing person. Therefore, we executed two experiments: in the first experiment, a subject pointed at 90 targets on the ground without any movements in the background of the subject. In the second experiment, the same subject pointed at the same targets, but a second person was moving and waving in the background of the subject (see Fig. 12).

In the first experiment our system estimated the correct radius with a rate of 86.4% and the angle with a rate of 54.6%. In the second experiment with a second person moving in the background the system reached a rate of 86.0% for the radius and a even slightly better rate of 57.3% for the angle. So, a second person moving and waving in the background of the subject does not reduce the estimation rates of the system. Next we conducted experiments about the information the system is getting from the texture of the body and face of the pointing person. We removed the texture of the subjects from the training and test data set leaving only a black and white silhouette of the subject after the background estimation (see Fig. 13) and trained and tested the system with the same algorithm as explained in Section 4. The system reached an estimation rate of only 43.6% for the radius of the target and 17.5% for the angle of the target, if only the silhouette of the subject was used. This is much worse than the results of the system when the texture of the subject was visible and shows that the system requires the texture information to estimate the correct target.

We also investigated, how much influence the position of the arm and the orientation of the head of the subject has on the estimation result. Therefore, we generated two additional training and test data sets. In the first set, we placed the Gaborjets only on several fixed points in the region of the pointing arm of the subjects. In the second set, we placed the Gaborjets only in the region of the head of the subjects (see Fig. 14). We trained and tested our system as described in Section 4 and compared the achieved results with those we got if using information from both regions (head and arm of the subjects). In case using only

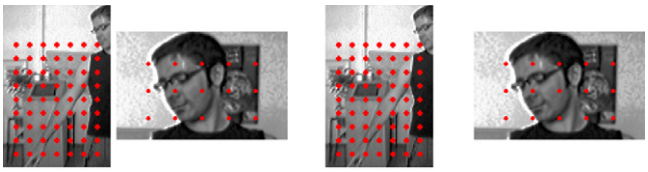


Fig. 14. Three different ways for the placement of the Gaborjets: (a) Gaborjets are placed in the region, of the head and the arm (b) Gaborjets only placed in the arm region, and. (c) Gaborjets only placed in the head region.

Gaborjets from the arm, the estimation rate for the radius was reduced to 87.9% (−0.5) and the estimation rate of the angle was reduced to 46.3% (−11.0). If using only Gaborjets from the head, the estimation rate fell to 34.1% (−54.3) for the radius and 10.0% (−47.3) for the angle of the target. As expected, this illustrates, that the estimation of the target point is not possible only with information from the head of the subject, but the information of the head helps to improve the estimation of the angle of the target point by nearly 10% compared to results were only information from the pointing arm is considered.

After selecting the MLP cascade (with background subtraction and discriminant analysis) as the best function approximator (based on our experiments described above), we tested the whole system under real-world conditions with our mobile robot Horos. Under such conditions, small errors of the face-tracking system, the speech recognition module, the person tracker, the navigator and the odometry of the robot are integrated and reinforce the error of the Pointing Pose Estimator. Under real-world conditions, the robot reached the selected target in 45.1% of the tests, which is an additional error of 5.5% compared to the results achieved in the offline experiments. The correct radius of the target was estimated in 86.3%, and the correct angle of the target in 47.1% of the tests. The results of these real-world experiments confirm the results of our experiments on the test data (see Table 1) with an additional error of 4%–6%.

6. Conclusion

In this paper, we presented an extension and additional experimental studies to our earlier approach in pointing pose estimation introduced in [1,2]. We compared different function approximators and architectures based on the same data set under the same conditions. Extensive experiments have shown, that the MLP-based approximator leads to the best estimation result. The major problems of the preceding approach—bad results in environments with structured background and a computation time which exceeds real-time requirements, could be solved. The realized approach is able to estimate a pointing position on the ground given only by monocular images with an accuracy equal to human observers. Moreover, it now works in real-time. This enables the user to command a mobile robot into a target position only by means of pointing poses. We also have shown, that our approach easily can be integrated in a complex robot control architecture, such as that presented in [19].

References

[1] H.-M. Gross, J. Richarz, S. Müller, A. Scheidig, C. Martin, Probabilistic multi-modal people tracker and monocular pointing pose estimator for visual instruction of mobile robot assistants, in: Proc. of the IEEE World Congress on Computational Intelligence, WCCI, 2006, pp. 8325–8333.

[2] J. Richarz, A. Scheidig, C. Martin, S. Müller, H.-M. Gross, A monocular pointing pose estimator for gestural instruction of a mobile robot, International Journal of Advanced Robotic Systems 4 (1) (2007) 139–150.

[3] O. Rogalla, M. Ehrenmann, R. Zöllner, R. Becher, R. Dillmann, Using gesture and speech control for commanding a robot assistant, in: Proc. of the 11th IEEE Int. Workshop on Robot and Human Interactive Communication, 2002, ROMAN, 2002, pp. 454–459.

[4] V. Paquin, P. Cohen, A vision-based gestural guidance interface for mobile robotic platforms, in: Proc. of the Workshop on HCI, Computer Vision in Human-Computer Interaction, ECCV, in: LNCS, vol. 3058, Springer, 2004, pp. 39–47.

[5] J. Triesch, C. von der Malsburg, A system for person-independent hand posture recognition against complex backgrounds, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (12) (2001) 1449–1453.

[6] N. Hofemann, J. Fritsch, G. Sagerer, Recognition of deictic gestures with context, in: Proceedings of the 26th DAGM Symposium on Pattern Recognition, vol. 3175, Springer, 2004, pp. 334–341.

[7] M. Bennewitz, T. Axenbeck, S. Behnke, W. Burgard, Robust recognition of complex gestures for natural human–robot interaction, in: Proc. of the Workshop on Interactive Robot Learning at Robotics: Science and Systems Conference, RSS, 2008.

[8] Z. Li, N. Hofemann, J. Fritsch, G. Sagerer, Hierarchical modeling and recognition of manipulative gesture, in: Proceedings of the IEEE International Conference on Computer Vision, Workshop on Modeling People and Human Interaction, IEEE Computer Society, 2005.

[9] K. Nickel, R. Stiefelhagen, Visual recognition of pointing gestures for human–robot interaction, Image and Vision Computing 25 (12) (2007) 1875–1884.

[10] A.D. Wilson, A.F. Bobick, Parametric hidden Markov models for gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 884–900.

[11] E. Hosoya, H. Sato, M. Kitabata, I. Harada, H. Nojima, A. Onozawa, Arm-pointer: 3D pointing interface for real-world interaction, in: ECCV Workshop on HCI, in: Lecture Notes in Computer Science, vol. 3058, Springer, 2004.

[12] N. Jojic, B. Brumitt, B. Meyers, S. Harris, T. Huang, Detection and estimation of pointing gestures in dense disparity maps, in: The fourth International Conference on Automatic Face- and Gesture-Recognition, 2000, pp. 468–475.

[13] Y. Hung, Y. Yang, Y. Chen, I. Hsieh, C. Fuh, Free-hand pointer by use of an active stereo vision system, in: ICPR'98: Proceedings of the 14th International Conference on Pattern Recognition, vol. 2, 1998, pp. 1244–1246.

[14] C. Nölker, H. Ritter, Illumination independent recognition of deictic arm postures, in: Proc. of the 24th Annual Conference of the IEEE Industrial Electronics Society, 1998, pp. 2006–2011.

[15] T. Urano, T. Matsui, T. Nakata, H. Mizoguchi, Human pose recognition by memory-based hierarchical feature matching, in: SMC (7), IEEE, 2004, pp. 6412–6416.

[16] K. Takahashi, S. Sugakawa, Remarks on human posture classification using self-organizing map, in: SMC (3), IEEE, 2004, pp. 2623–2628.

[17] R. Stiefelhagen, Estimating head pose with neural networks—Results on the pointing04 ICPR workshop evaluation data, in: Proc. of the Pointing 04 ICPR Workshop, 2004.

[18] V. Krüger, G. Sommer, Gabor wavelet networks for efficient head pose estimation, Image and Vision Computing 20 (9–10) (2002) 665–672.

[19] C. Martin, A. Scheidig, T. Wilhelm, C. Schröter, H.-J. Böhme, H.-M. Gross, A new control architecture for mobile interaction robots, in: Proc. of the 2nd European Conference on Mobile Robots, ECMR 2005, Stampalibri, 2005, pp. 224–229.

[20] P.A. Viola, M.J. Jones, Rapid object detection using a boosted cascade of simple features, in: Proc. of the Conf. on Computer Vision and Patter Recognition, 2001, pp. 511–518.

[21] B.K.P. Horn, Robot Vision, MIT press, 1986.

[22] P.A. Lachenbruch, Discriminant Analysis, Hafner, 1975.

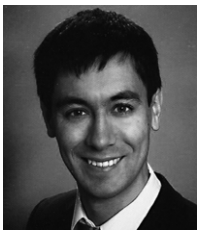
[23] G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, Wiley-Interscience, 2004.

[24] T. Martinetz, K. Schulten, A Neural-Gas network learns topologies, in: Proc. of the ICANN 1991, 1991, pp. 397–402.

[25] T. Kohonen, Self-organized formation of topologically correct feature maps, Biological Cybernetics 43 (1982) 59–69.

[26] H. Ritter, in: T. Kohonen, et al. (Eds.), Learning with the Self-Organizing Map, in: Artifical Neural Networks, Elsevier Science, 1991.

Christian Martin is of the founders of the company MetraLabs GmbH, where he is the Head of Software Development. He has been a Ph.D. student at the Department of Neuroinformatics and Cognitive Robotics at the Ilmenau Technical University since 2004. He received his Diploma degree in Computer Science from the Ilmenau Technical University in 2003. His Ph.D. research is concerned with multi-modal human–robot interaction, especially the development of modelling concepts for human–robot interaction and autonomous mobile robot control architectures.



Frank-Florian Steege has been a Ph.D. student at the Department of Neuroinformatics and Cognitive Robotics at the Ilmenau Technical University since 2008. He received his Diploma degree in Computer Science from the Ilmenau Technical University in 2007. His Ph.D. research is concerned with Neural Information Processing and Adaptive Control as well as Image Processing and Feature Selection Techniques.

Horst-Michael Gross is full professor of Neuroinformatics and head of the Neuroinformatics and Cognitive Robotics Lab at the Ilmenau University of Technology. He received his doctoral degree in Computer Science and Neural Computing in 1989 from Ilmenau University. From 1998–2005 he was Dean of the Faculty of Computer Science and Automation of this University. Among his current research interests are Cognitive Robotics (Autonomous Robots, Human–Robot Interaction, Probabilistic Robotics), Neural Information Processing, Computer and Robot Vision, and Reinforcement Learning.