Estimation of Pointing Poses for Visual Instructing Mobile Robots under Real World Conditions

Christian Martin* Frank-Florian Steege* Horst-Michael Groß*
*Neuroinformatics and Cognitive Robotics Lab
Ilmenau Technical University
Ilmenau, Germany

***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 gestures only. Because one objective of our work is the development of a low-cost system, we only used a monocular vision system. As first step, our approach employes a Background Substraction technique and a histogram equalization in a preprocessing step to work in environments with structured backgrounds and under variable lighting conditions. Furthermore, a Discriminant Analysis was used to find the best features for the pointing pose estimator. For the estimation process, different types of Neural function approximators were implemented and compared with each other. The approach presented in this paper has been also implemented on our mobile interaction robot HOROS to determine the performance of under real-world conditions. The best algorithm is able to estimate the target position in real-time on the robot. Furthermore, we compared the accuracy of our approach with humans performing the same estimation task, and achieved very comparable results.**

***Index Terms*— Human-Robot Interaction, Vision, Gesture Recognition**

I. INTRODUCTION AND STATE-OF-THE-ART

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. Todays robot systems mainly provide a keyboard, a touchscreen or other input devices for getting input from the user. More and more projects try to integrate a speech recognition onto the robot, but a robust speech recognition is still a hard problem. But besides this verbal communication also the non-verbal communication plays a very important role in a dialog between humans. To the best knowledge of the authors, only a few projects have already successfully integrated non-verbal communication aspects in an interactive dialog on their mobile robots. In the work presented in this paper, we show how a basic non-verbal communication (more precisely: the problem, of instructing a mobile robot by the use of pointing gestures/poses) can be realized on a mobile robot system.

In the field of mobile service robotics, the possibility to direct the robot to a certain position is an important part of the interaction. Gestures or poses (sometimes in combination with spoken commands) are a very intuitive way to instruct the robot without the use of certain input devices (e.g. a joystick). 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, creating a command alphabet for robot control.

Rogalla et al. [1], 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 Chohen [2] also use a skin color segmentation to track the hands and the head of a user. They use a Neural Network based approach to classify the trajectories recorded during the progress of the gesture and are able to recognise nine different robot instruction gestures like ”stop” or ”forward”.

Triesch and v.d. Malsburg [3] 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, that the specific commands of the command alphabet have to be known by the user. Another problem is, that the directing of the robot based on simple discrete commands is only possible in certain steps (for example ”drive forward”, ”drive to the left” and again ”drive forward” to direct a robot to a position 30° in front of the starting position), since typically only one of these commands can be executed at a time.

A much more intuitive and smoother way to direct the robot is through pointing directly at the target position on the ground. In [4, 5] for the first time we presented an approach, which allows to direct a mobile robot to a certain position by means of such pointing poses. The system presented was capable of estimating the target point of the pointing gesture 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 seconds was required for the estimation of a single target. These constraints 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 work is partially supported by TAB-Grant #2006-FE-0154 to H.-M. Gross and a HWP-Grant to A. Scheidig.

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.

This paper is organized as follows: After this introduction, Section II introduces our mobile robot platform HOROS, which was used again for extended investigations. After that, Section III explains, how the pointing poses can be estimated and how our entire system is designed. In Section IV the experiments and results will be presented. The papers ends with conclusions in Section V.

II. The Mobile Robot Horos

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 laser range-finder and a ring of sonar sensors (see Fig. 1). 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. Because one objective of our project is the development of a low-cost prototype of a mobile and interactive robot assistant, we are especially interested in vision technologies with a good price-performance ratio. Therefore, the two low-cost frontal cameras were utilized instead of a high-end stereo-vision system. This forces us to develop powerful and robust recognition algorithms allowing to compensate the deficits of the hardware. In this context, we were 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.

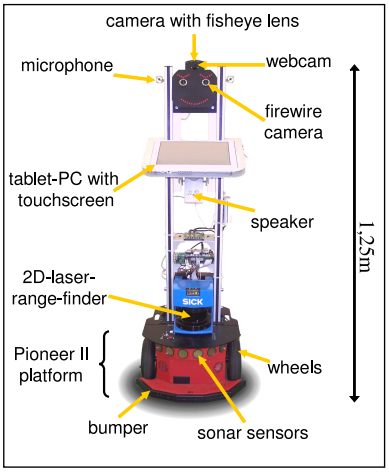

Fig. 1. Mobile service robot HOROS, used for experimental investigation of the pointing pose estimation. The images for the estimation of the pointing target were taken with the firewire camera (located in the left eye).

HOROS is controlled by a highly flexbile and extensible control architecture described in [6]. The approach described in this paper was implemented in this control architecture framework, which allows 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.

III. Estimation of Pointing Poses

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

A. Architecture for a pointing pose estimation system

To develop a system for an interactive mobile robot, which enables the robot to move to a referred point on the ground a complex architecture (see Fig. 2) is necessary.

First a *Pointing Pose Estimation Module* is needed, which can estimate the referrend point on the ground based on a sequence of monocular images. For this estimation process a face detection system [7] is used to find the position of the head (x_{head}, y_{head}) of the user in the image. Moreover, a multimodal person tracker [4, 5] is utilized to determine the direction ϕ_{user} and the distance d_{user} of the user to the robot. These data are processed to select the regions of interest (ROI) in the input image for the subsequent feature extraction. 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 III-B). With the tracking result from the person tracker and the estimated radius and angle, the referred goal point (x_{goal}, y_{goal}) on the ground can be computed in a local, robot-centered coordinate system. Given the current pose of the robot $(x_{robot}, y_{robot}, \phi_{robot})$, 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.

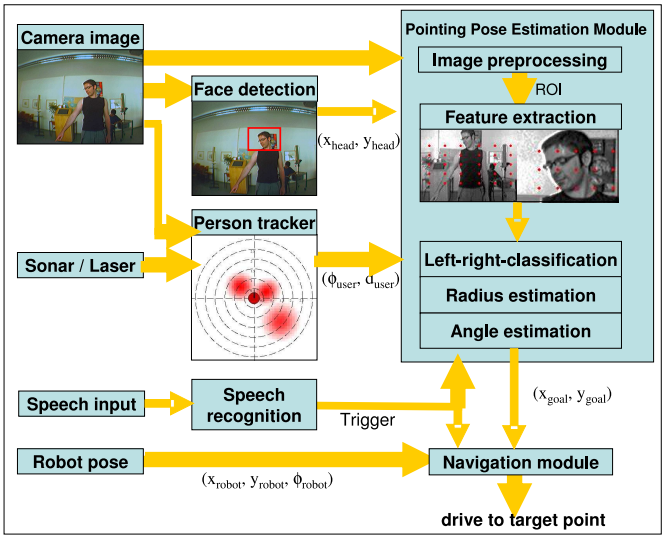

Fig. 2. Architecture of the proposed visual instructing system. The Pointing Pose Estimation Module (right) uses face detection and people tracking subsystems to estimate the pose on the ground, and a navigation module executes the movement of the robot to the estimated target point.

To embed this 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 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.

B. Training-Data and Ground-Truth

To develop the Pointing Pose Estimation Module, 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. 3) 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 3m 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 2m, this spans a valid pointing area of approximately 6 by 3m on the floor in front of the robot in which the indicated target points may lie. Figure 3 shows the configuration we chose for recording the training data. The subjects stood at distances of 1, 1.5 and 2m from the robot. Three concentric circles with radii of 1, 2 and 3m 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 an image was recorded each time. Pointing was performed as a defined pose, with outstretched arm and the user fixating the target point (see Fig. 3, 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 IV). This way, we collected a total of 2,340 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 1,710 samples.

C. 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 [4, 5] to determine the region of interest (ROI) by using a combination of face-detection (based on the Viola & Jones Detector cascade [7]) and some empirical factors. With the help of a multimodal tracker [4, 5] 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

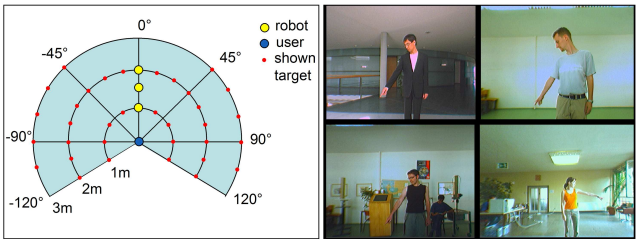

Fig. 3. The left image shows the configuration used for recording the ground truth training and test data: The subjects stood in front of the robot and pointed at one of the marked targets on the ground. The distance of the robot to the subject varied between 1m and 2m. The images on the right show typical examples of images of subjects taken by the 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 [4, 5]).

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 operations used to capture and normalize the image are illustrated in Fig. 4. 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 [8] (see Fig. 5). The influence of the Background Substraction on the pose estimation result was tested in comparsion with our approach in [4, 5] where no Background Subtraction was used (see Section IV). On the normalised image regions, features were extracted to approximate the pointing pose of the user. In our work, Gaborfilters of different orientations and frequencies, 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 shown in Fig. 4.

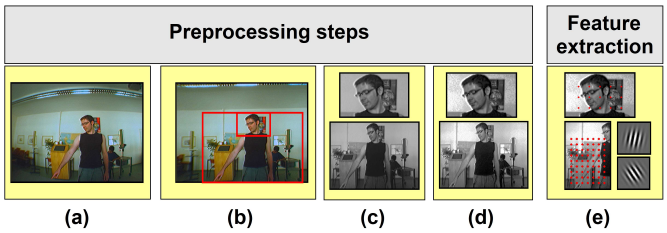

Fig. 4. Steps of preprocessing and feature extraction: the raw distorted image of the lowcost camera in the robot's eye (a) is transformed into an undistorted image, and the face of the user is detected by means of [7] (b). Based on the height of the face in the picture and the distance of the user given 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 Gaborfilters placed at pre-defined points of the image (marked as red dots in (e)). A Background Subtraction (see Fig. 5) was optionally used between steps (d) and (e).

D. Discriminant Analysis

The Discriminat Analysis [9, 10] is a well-known technique to figure out the most relevant features in a feature space for

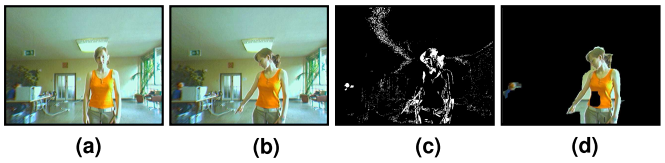

Fig. 5. The Background Subtraction used in our approach. (a): by the use of a command word (for example the name of our robot HOROS) the user triggers the capture of a new background image. When the user is pointing at the target, the current image (b) is subtracted from the background image resulting in a difference image (c). With the help of a closing algorithm and the search for connected regions [8] the image is post-processed resulting in an image with the segmented user (d).

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 computation time to estimate the target position based on the describing features.

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 Gaborfilter answers for the selected feature was computed at all samples of the training data set mentioned in Section III-B. Every value was assigned to a certain class r which was defined through the target point the subject refered to in the current sample. Then, for feature k the discriminant value $\sigma_{rs}^{(k)}$ between the classes r and s was computed as follows:

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

$b_i^{(k)}$ is the Gaborfilter 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 value if the samples of each class have a little intra-class variance (the term bellow the fraction stroke) and if the different classes do not overlap (the inter-class variance given above the fraction stroke). The results of equation (1) - applied for all combinations of two classes r and s - were summed up resulting in a single discriminant value for the feature k. Figure 6 shows the discriminant values for selected features. Gaborfilters with high dicriminant values directly correspond to the possible alignments of the pointing arm, while features with low values correspond to Gaborfilter 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 dicriminant values and ignoring features with low dicriminant values, we achieved higher robustness against cluttered background and a considerable faster computation since less Gaborfilter features had to be determined.

E. Approximation of the Target Point

In [4, 5] a cascade of several Multi-Layer Perceptrons (MLP) was used to estimate the target point from the extracted

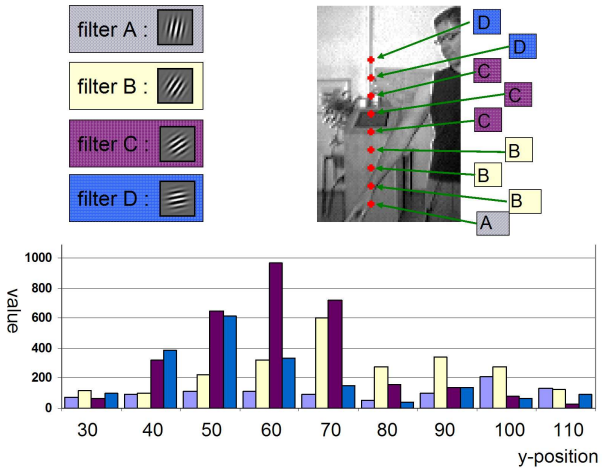

Fig. 6. Determination of important features with the help of a Discriminant Analysis: the bar chart shows the discriminant values of the Gaborfilter features (A to D, shown top left) for each of the fixed filterpoints in one vertical line in the image. Top right, the filter with the highest discriminant value at the certain position is displayed. Obviously, filters with high dicriminant values directly correspond to the possible orientations of the pointing arm of the subjects.

features. However 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 scenarios. Nölker et al. [11] used a Local Linear Map (LLM) and a Parametrized Self-Organizing Map (PSOM) to estimate the target of a pointing pose on a screen the user is pointing to. In [12] Gaborfilters and a LLM are utilized to estimate the head pose, while Stiefelhagen [13] 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 on 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 Sec. III-B). 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 roughly:

k-Nearest-Neighbour Classifi cation: The k-Nearest-Neighbour 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 neighbours to the input in the feature space. The label that appears most often at the k neighbours is mapped on the new input. This method only allows classification and not an approximation between the labels of two or more neighbours. Therefore, we slightly modified the method in our approach. Now the label for the input $f_k(\mathbf{x})$ is determined as follows:

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

This way the labels l_i of the k nearest neighbours contribute to the output and are weighted with their Euclidian distance

d_i to the input $\mathbf{x}$.

Neural Gas: A Neural Gas network (NG, [14]) 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 $D_x = \{\|\mathbf{x} - \mathbf{w}_i\|, i = 1, \cdots, N\}$ 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.

Self-Organizing Map: An approach very similar to the NG is the well known Self-Organizing Map (SOM, [15]). The SOM differs from the NG in the fact that the neurons of the SOM are connected in a fixed topological structure. The neighbours 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).

Local Linear Map: The Local Linear Map (LLM, [16]) 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 $(\mathbf{w}_i^{in}, \mathbf{w}_i^{out})$ in the in- and output-space and an associated linear mapping $\mathbf{A}_i$ which is only locally valid. The answer $\mathbf{y}_{bm}$ of the best-matching neuron of the LLM to an input $\mathbf{x}$ is calculated as follows:

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

The weights and the mapping matrix are also learned during the training process.

Multi-Layer Perceptron: For our experimental comparison, we also used a cascade of several MLPs as decribed in [4, 5]. 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'.

IV. 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 III-B. These tests were used to indicate which function approximator is the best for the problem of estimating the target point of a pointing pose. 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 a simple 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 estimated the target. Thus, the subjects were estimating the target on the images having the same conditions as the different function approximators. 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 subject 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 Fig. 7. The label *Human (srceen)* 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.

(a) correct estimation of radius

radius estimation

correct samples in % / mean error in m	k-NN	NG	SOM	LLM	MLP
Gaborfilters	48,18 % / 0,314 m	33,85 % / 0,458 m	42,93 % / 0,443 m	54,45 % / 0,378 m	70,46 % / 0,235 m
Gaborfilters and BG Subtraction (BGS)	64,84 % / 0,246 m	65,16 % / 0,286 m	65,31 % / 0,244 m	77,74 % / 0,280 m	**88,21 % / 0,134 m**
Gaborfilters and Discriminant Analysis	60,17 % / 0,292 m	48,49 % / 0,323 m	56,34 % / 0,326 m	64,90 % / 0,338 m	74,41 % / 0,216 m
Gaborfilters, BGS and Discriminant Analysis	**82,81 % / 0,124 m**	74,16 % / 0,208 m	79,27 % / 0,186 m	**84,24 % / 0,226 m**	**88,40 % / 0,138 m**

Human (2 eyes)	84,25 % / 0,080 m
Human (1 eye)	75,2 % / 0,099 m
Human (screen)	75,00 % / 0,350 m

(b) correct estimation of angle

angle estimation

correct samples in % / mean error in °	k-NN	NG	SOM	LLM	MLP
Gaborfilters	23,10 % / 23,00 °	13,91 % / 23,20 °	15,63 % / 23,61 °	21,61 % / 21,79 °	41,39 % / 18,51 °
Gaborfilters and BG Subtraction (BGS)	34,37 % / 20,29 °	27,72 % / 21,36 °	23,50 % / 20,91 °	30,28 % / 18,76 °	50,91 % / 17,23 °
Gaborfilters and Discriminant Analysis	29,41 % / 23,05 °	19,36 % / 22,23 °	20,70 % / 23,39 °	24,73 % / 23,77 °	37,82 % / 20,99 °
Gaborfilters, BGS and Discriminant Analysis	41,93 % / 17,46 °	30,55 % / 20,54 °	29,85 % / 20,96 °	37,68 % / 19,55 °	57,28 % / 15,64 °

Human (2 eyes)	74,66 % / 4,50 °
Human (1 eye)	56,19 % / 7,35 °
Human (screen)	50,00 % / 13,76 °

(c) combined estimation

target point estimation (correct radius ***and*** correct angle)

correct samples in %	k-NN	NG	SOM	LLM	MLP
Gaborfilters	11,12%	4,70%	6,70%	11,76%	29,16%
Gaborfilters and BG Subtraction (BGS)	22,28%	17,72%	15,34%	23,53%	44,90%
Gaborfilters and Discriminant Analysis	17,69%	9,38%	11,66%	16,04%	28,14%
Gaborfilters, BGS and Discriminant Analysis	34,72%	22,66%	23,66%	31,74%	50,63%

Human (2 eyes)	62,90 %
Human (1 eye)	40,75 %
Human (screen)	37,50 %

Fig. 7. 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. Fig. (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. Fig. (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.

The results of the several approaches for estimating the target position are shown in Fig. 7. As described in Sect. III-B

the ground truth data is a tuple (r, φ) with the target radius r and the target angle φ. The separate results for the estimation of r and φ are shown in Fig. 7(a) and Fig. 7(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 50cm from the ground truth radius and φ differed less than 10° from the ground truth angle. Figure 7(c) shows the results for a correct estimation of both values.

Every of the five selected approaches was trained on the same training data set and tested on the same test data set. For each system, we used four different feature extraction strategies: first only Gaborfilters were utilized, second we combined Gaborfilters with an additional Background Subtraction to reduce the effects of the different cluttered backgrounds in the images. Third, we used only those Gaborfilters that had a high discriminant value extracted by means of the Discriminant Analysis executed over all predefined Gaborfilter positions (see Section III-D). Fourth, we combined Gaborfilter, Background Subtraction and utilized only the relevant features extracted by the Discriminant Analysis.

The results demonstrate, that a cascade of several MLPs as proposed in [4, 5] 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 improve the results for all different classifier systems. The usage of this two algorithms, combined with the histogram equalization in the preprocessing step, now also allows to handle background clutter and different lighting conditions, which was not able in our previous work. The best system is capable of estimating r as good as humans with their binocular vision system in a real-world environment and even better than humans estimating the target on 2D screens. The estimation of φ does not reach equally good values. The system is able to reach a result equally to that of humans on 2D screens or humans with one eye blindfolded, but it is not able to estimate the angle as good 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 approximators.

The implemented Pointing Pose Estimation Module is able to run in real-time. The total computation time (on an Athlon XP 2800 CPU) with Background Substraction and Discriminant Analysis was 38ms for the NG, 42ms for the SOM, 35ms of the LLM and 31ms of the MLP cascade. The k-NN classifier requires 129ms and is therefore not suitable for real-time processing.

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 odometrie of the robot are integrated and reinforce the error of the Pointing Pose Estimation Module. 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 test data set. 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 Figure 7) with an additional error of 4-6% due to the errors in the input data for the Pointing Pose Estimation Module.

V. CONCLUSION

In this paper we presented an extension to our earlier approach introduced in [4, 5]. The major problems of the old approach - bad results in environment with structured background and a computation time which exceeds real-time requirements, could be solved. Extensive experiments with different function approximators have shown, that the MLP-based approximator leads to the best estimation result. The realized approach is able to estimate a pointing position on the ground given only by monocular images with an accuracy equal to humans. Moreover, it works now in real-time. This enables the user to control a mobile robot system into a target position only by means of pointing gestures. We also have shown, that our approach easily can be integrated in a complex robot control architecture.

REFERENCES

[1] O. Rogalla, M. Ehrenmann, R. Zöllner, R. Becher, and R. Dillmann. Using Gesture and Speech Control for Commanding a Robot Assistant. In *In Proc. of the 11th IEEE Int. Workshop on Robot and Human Interactive Communication, 2002 (ROMAN)*, pages 454–459, 2002.

[2] V. Paquin and P. Cohen. A Vision-Based Gestural Guidance Interface for Mobile Robotic Platforms. In *In Proc. of the Workshop on HCI, Computer Vision in Human-Computer Interaction, ECCV*, volume 3058 of *LNCS*, pages 39–47, Prague, Czech Republic, 2004. Springer.

[3] J. Triesch and C. von der Malsburg. A System for Person-Independent Hand Posture Recognition against Complex Backgrounds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(12):1449–1453, 2001.

[4] H.-M. Gross, J. Richarz, S. Müller, A. Scheidig, and C. Martin. Probabilistic Multi-modal People Tracker and Monocular Pointing Pose Estimator for Visual Instruction of Mobile Robot Assistants. In *In Proc. of the IEEE World Congress on Computational Intelligence (WCCI)*, pages 8325–8333, Vancouver, Canada, 2006.

[5] J. Richarz, A. Scheidig, C. Martin, S. Müller, and H.-M. Gross. A Monocular Pointing Pose Estimator for Gestural Instruction of a Mobile Robot. *Int. Journal of Advanced Robotic Systems*, 4(1):139–150, 2007.

[6] C. Martin, A. Scheidig, T. Wilhelm, C. Schröter, H.-J. Böhme, and H.-M. Gross. A new Control Architecture for Mobile Interaction Robots. In *Proc. of the 2nd European Conference on Mobile Robots (ECMR 2005)*, pages 224–229, Ancona, Italy, 2005. stampalibri.

[7] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *In Proc. of the Conf. on Computer Vision and Patter Recognition*, pages 511–518, Munich, Germany, 2001.

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

[9] P. A. Lachenbruch. *Discriminant analysis*. Hafner, 1975.

[10] G. J. McLachlan. *Discriminant analysis and statistical pattern recognition*. Wiley-Interscience, 2004.

[11] C. Nölker and H. Ritter. Illumination Independent Recognition of Deictic Arm Postures. In *In Proc. of the 24th Annual Conference of the IEEE Industrial Electronics Society*, pages 2006–2011, 1998.

[12] V. Krüger and G. Sommer. Gabor wavelet networks for effi cient head pose estimation. *IVC*, 20(9-10):665–672, August 2002.

[13] R. Stiefelhagen. Estimating Head Pose with Neural Networks - Results on the Pointing04 ICPR Workshop Evaluation Data. In *In Proc. of the Pointing 04 ICPR Workshop*, Cambridge, UK, August 2004.

[14] T. Martinetz and K. Schulten. A Neural-Gas Network Learns Topologies. In *In Proc. of the the ICANN 1991*, pages 397–402, 1991.

[15] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

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