There You Go! - Estimating Pointing Gestures In Monocular Images For Mobile Robot Instruction.

J. Richarz, C. Martin, A. Scheidig and H.M. Gross

Abstract— **In this paper, we present a neural architecture that is capable of estimating a target point from a pointing gesture, thus enabling a user to command a mobile robot to a specific position in his local surroundings by means of pointing. In this context, we were especially interested to determine whether it is possible to implement a target point estimator using only monocular images of low-cost webcams. The feature extraction is also quite straightforward: We use a gabor jet to extract the feature vector from the normalized camera images; and a cascade of Multi Layer Perceptron (MLP) Classifiers as estimator. The System was implemented and tested on our mobile robotic assistant HOROS. The results indicate that it is in fact possible to realize a pointing estimator using monocular image data, but further efforts are necessary to improve the accuracy and robustness of our approach.**

I. Introduction

In recent years, a lot of research has been done to develop mobile robots that can interact with - and be controlled by - non-instructed users, making them suitable for application in everyday life. To achieve this, it is essential to integrate man-machine-interfaces that are natural and intuitive to use. Gestures are a very important aspect of non-verbal interhuman communication. In particular, pointing gestures simplify communication by linking speech to objects in the environment in a well-defined way.

Therefore, many research projects focusing on integrating gesture recognition into man-machine-interfaces have emerged. However, most of this work concentrates on distinguishing different gestures, creating a "command alphabet" for robot control. Rogalla et al [1], for example, use the fourier descriptors of an extracted hand contour and a model database to classify different hand postures. Licsar and Sziranyi [2] use a similar approach to control an augmented-reality system. Triesch and v.d. Malsburg [3][4] detect and classify hand postures by using compound bunch graphs. They succeed in building a robust system that can cope with highly complex backgrounds. Alon et al. [5] simultaneously track and classify dynamic gestures with an algorithm called "Dynamic Space-Time Warping" (DSTW). Rahman and Ishikawa [6] use Eigenspaces to distinguish the trajectories of different dynamic gestures.

J. Richarz is now with:
Intelligent Systems Group, Robotics Research Institute,
University of Dortmund, Germany. (www.irf.de)
jan.richarz@udo.edu
C. Martin, A. Scheidig and H.M. Gross are with:
Department of Neuroinformatics and Cognitive Robotics,
Ilmenau Technical University, Germany.
This work was conducted at:
Department of Neuroinformatics and Cognitive Robotics,
Ilmenau Technical University.

There are only a few authors who actually tried to estimate a pointing direction out of a deictic gesture. For example, Jojic et al. [7] do so by detecting a person using dense disparity maps and color information. A simple Gaussian mixture model is fitted to the person and the pointing direction is determined from the largest principal component of the "arm-blob". Noelker and Ritter [8] use a Local Linear Map (LLM) classifier to detect 2D features in the images of two cameras. A Parametrized Self organizing Map (PSOM) then estimates the 3D coordinates of these features, making it possible to calculate a pointing direction. The approach is used to control a Virtual-Reality-System. However, the working conditions for their system are very restrictive. Nickel and Stiefelhagen [9] classify dynamic gestures by means of Hidden Markov Models (HMM) and estimate the pointing direction of a pointing gesture by calculating the connecting line between the center of the head and the hand. They also use a stereo camera system.

With our approach, we want to determine whether it is possible to implement a pointing direction estimator using only monocular images of low-cost cameras. Our goal is to implement that approach on our mobile robot platform HOROS and make it navigate to specified targets, thus enabling a user to control the robot only by means of pointing. To the best of our knowledge, there are no other low-cost oriented approaches that are comparable to the one presented here.

II. The mobile robot HOROS

Out target system is a mobile experimental robot acting as an intelligent agent in indoor environments. The robot HOROS (HOme RObot System) is based on a Pioneer2 platform by ActiveMedia Robotics (Fig. 1). It integrates an on-board PC (Pentium IV M, 1.6 GHz, 512MB) and is equipped with sonar sensors, a 180 ° Laser Scanner, simple microphones for sound input, and a total of three cameras: an omnidirectional 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. It further includes a tablet PC for touch-based interaction, speech recognition and speech generation. Because we are currently developing a low-cost prototype of a mobile and interactive shopping assistant, we are especially interested in vision technologies with a good price-performance ratio. Therefore, the two frontal cameras are inexpensive webcams.

Since HOROS' cameras have very different fields of view (due to different recognition tasks to be solved), it is impossible to use a combination as a kind of simple stereo

1-4244-0565-3/06/$20.00 ©2006 IEEE.

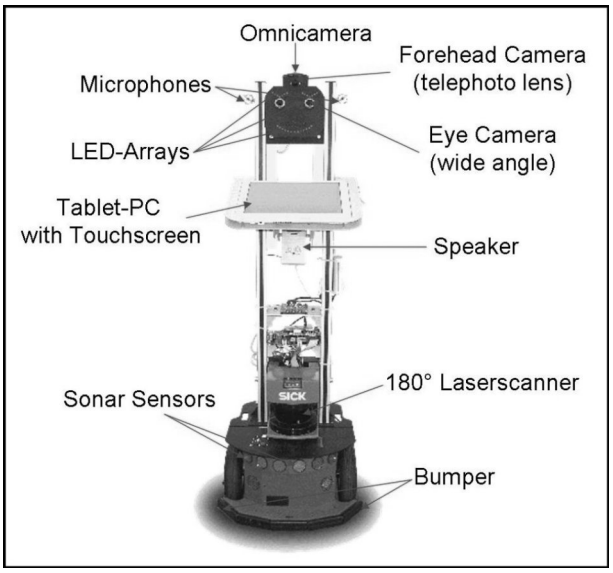

Fig. 1. Our mobile robotic assistant HOROS (HOme RObot System).

system. Furthermore, only the eye camera's field of view is large enough to record a pointing person. Fig. 2 shows an example for an image provided by this camera. The sonar and laser sensors enable us to determine the distance of a person standing in front of the robot with good accuracy up to a distance of approximately two meters. For greater distances, the reliability of the sensor measurements decreases rapidly.

III. System overview

A. Pointing Area and Ground Truth Data

We encode the target points as (r, ϕ) coordinates in a user-centered polar coordinate system. This requires a transformation of the target estimate into the robot's coordinate system (by simple trigonometry), but the estimation task becomes independent of the distance between user and robot. 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 meters and a value range for ϕ from -120 ° to 120 °. The 0 ° direction is defined as user-robot-axis, negative angles are

Fig. 2. An example for an image provided by the eye camera.

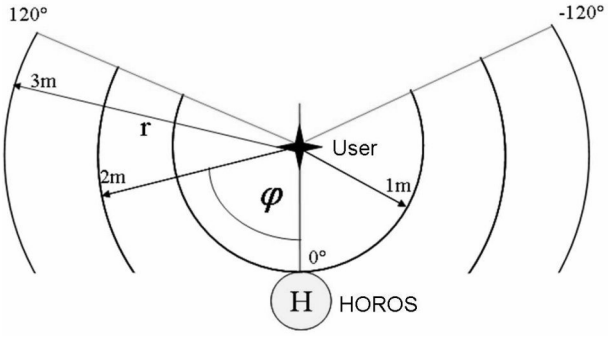

Fig. 3. The configuration used for recording the training data. Here, only one of the markers is shown for reasons of clarity.

on the user's left side. With respect to a predefined maximum user distance of 2 meters, this spans a valid pointing area of approximately 6 by 3 meters.

Fig. 3 shows the configuration we chose for recording the training data. There are three markers (distance 1, 1.5 and 2 meters from the robot) specifying different user positions. Around each marker, three concentric circles with radii of 1, 2 and 3 meters are drawn, 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 (see Fig. 2). Pointing was performed as a defined pose, with outstretched arm and the user fixating the target point. All captured images are labelled with distance, radius and angle, thus representing the ground truth used for training. This way, we collected a total of 900 images of ten different interaction partners. During preprocessing, the data were slightly variated to receive nine samples per training image, resulting in a training sample database of 8100 labelled images.

B. Preprocessing and feature extraction

Since the persons standing in front of the camera can have different height and distance, an algorithm had to be developed that can calculate a "normalized" region of interest (ROI), resulting in similar subimages for subsequent processing. We use a combination of head-shoulder-detection (using a Viola-Jones Detector cascade, see [10]), empiric factors and a distance measurement from a multimodal person tracker (see [11]) to determine the ROI (Fig. 4).

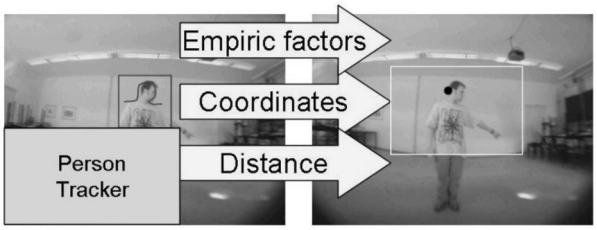

Fig. 4. Determining the ROI in the camera image. A combination of empiric factors, head-shoulder-detector and measurement of the person-camera distance from a multimodal tracker is used to achieve a normalization.

Fig. 5. Examples for extracted ROI with the described normalization algorithm. Top: Extracted image regions for different subjects all performing the same pose. Subjects' heights variate from 1.65 to 2 meters. Bottom: Extracted image regions for different distances person-camera, distance ranging from 1 to 2 meters.

The head-shoulder detector provides a starting point and implicitly includes the user's height into the calculation. For different people, the maximum distance between the center of detection and the tip of the pointing arm in both x and y direction were determined before. These distances were divided by the y-coordinate of the head-shoulder-detection, yielding a factor specifying the size of the ROI. Thus, the size of the extracted image region implicitly depends on the user's height. Assuming that the ratio between height and arm length is approximately the same for most humans, this results in an extracted image region that is very similar in most cases. Finally, the distance estimation from the tracking system allows a simple scaling of the respective ROI. Typical ROIs captured this way are shown in Fig. 5.

The found ROI is scaled to 81x81 pixels, and then an illumination correction and histogram equalization is applied. After this, the preprocessed subimage is Gabor-filtered (4 frequencies, 8 orientations) using an equidistant grid to extract a pose-describing feature vector as input for the first stage of our pointing estimator. For later stages, the ROI is modified again to create two subimages, one of them containing the pointing arm, the other the head (Fig. 6, see section III-C for details). By doing this, we integrate the head pose as additional information.

C. Architecture of the Classifier Cascade

Informal experiments showed that it is not possible to tackle the function approximation problem with a single MLP network estimating both radius and angle in one step. It also became clear, that while the radius estimation works quite well, it was very difficult to robustly estimate the angle. Therefore we decided to use a cascade of classifiers. Fig. 7 gives an overview over the architecture of the developed target point estimator cascade.

Fig. 6. Examples for extracted image data using two image samples, one containing the pointing arm, the other the head. By using these samples as input data, the head orientation is integrated as additional information.

After extracting the ROI, a left/right classifier first determines whether the person is pointing to the left or to the right. Knowing this, that half of the input image that does not contain the pointing arm can be discarded. This way the "finer" ROI shown in Fig. 6 is extracted. If the person is pointing to the left, the image is simply flipped. This allows us to use the same classifier for both directions.

In the following cascade stage, the value of the pointing radius r is estimated. Since the estimation of ϕ is much less accurate and prone to errors, this estimation is done later in the cascade, so it can be given as much supporting and simplifying information as possible. To that purpose, three coarse radius classes were defined. For each of this classes, there is a specialized classifier assigning the sample to a coarse angle class. Finally, within the respective coarse class, a finer estimation of ϕ is determined, leading to the final target estimation [r, ϕ]. The system contains a total of 14 MLP classifiers (left/right, radius, 3x coarse angle, 9x fine angle), but due to the hierarchical architecture, only four of these classifiers have to be activated during one pass. Calculating all four nets takes less than 100 milliseconds per image on the robot's onboard computer.

IV. Experimental results

A. Estimation Results of Human Viewers

In order to get a reference value for the recognition performance of our estimator, separate experiments determined how accurate a human viewer could estimate the referred target point from a monocular image. The images from the training and test data sets were presented to test viewers in random order using the graphical user interface shown in Fig. 8. The valid area for the pointing targets is specified by a circle segment. Subjects were told beforehand that the targets can only lie inside this area. The circle segment is skewed perspectively to create a 3D impression and adapted in size according to the distance between the person in the image and the camera. Each subject could adjust the perspective freely so that it occured most natural. The test viewers could mark a guessed target point by clicking with the mouse

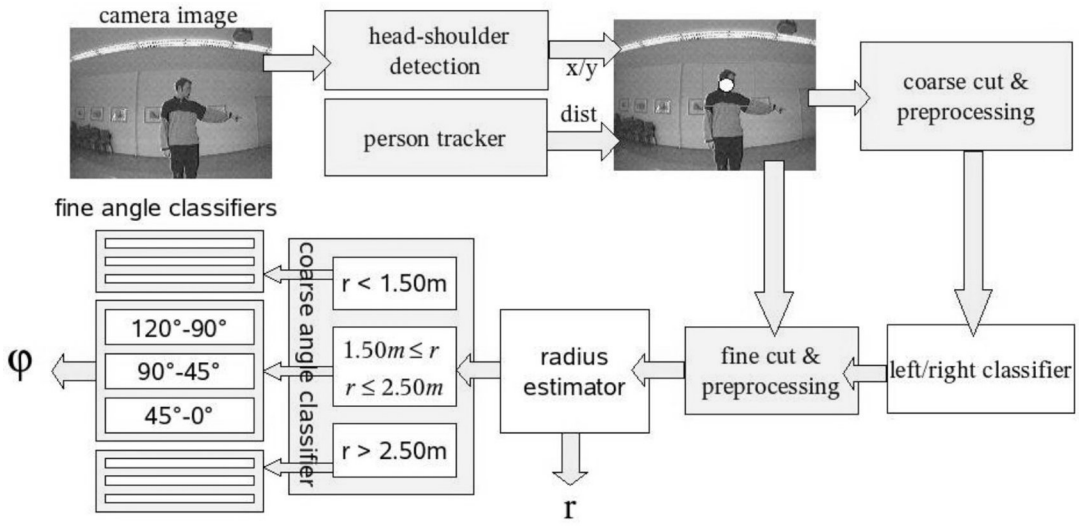

Fig. 7. System overview of the target point estimator cascade. The gabor-filtered subimage is first fed into a left/right - classifier. The result of this classifier enables it to extract the finer image ROI shown in Fig. 6. In the following stage, the radius r is estimated and three radius classes are generated artificially. For each class, a coarse angle estimator is trained, yielding a classification into one of three angle classes. The last stage yields the final angle estimate ϕ.

pointer on the interface. The found coordinates were then transformed according to the given perspective and person distance, yielding the estimated target coordinates r and ϕ. The estimates were then compared with the known image label.

These comparing experiments were performed with eight test viewers, resulting in 885 target estimates altogether. The achieved estimation accuracy is shown in Fig. 9. On the top, the mean values and standard deviations of the angle estimate are shown over the correct angle. Obviously, perfect estimates would lie on a straight line depicted by the dotted line in the image. The mean values of the estimates deviate from this ideal case, forming a symmetric sigmoidal function. It is noticeable that angle value estimates between (+/-) 45 ° and 90 ° are persistently too large in magnitude. Additionally, the standard deviations (depicted by the vertical lines) for these angles are significantly higher. So, it seems to be quite difficult for a human viewer to precisely estimate ϕ from the monocular images in this area.

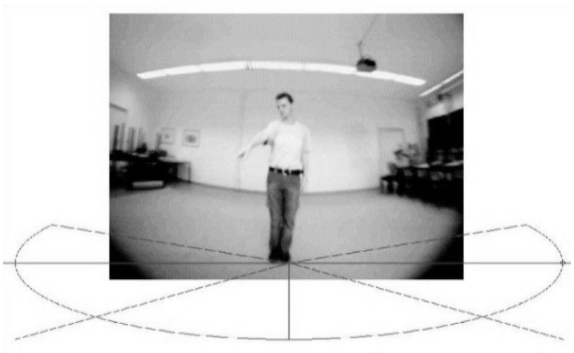

Fig. 8. A screenshot of the test program for human viewers. The valid target area is shown by the circle segment.

At the bottom of Fig. 9, the average errors for the estimates of r and ϕ over the correct angle are shown. For the radius

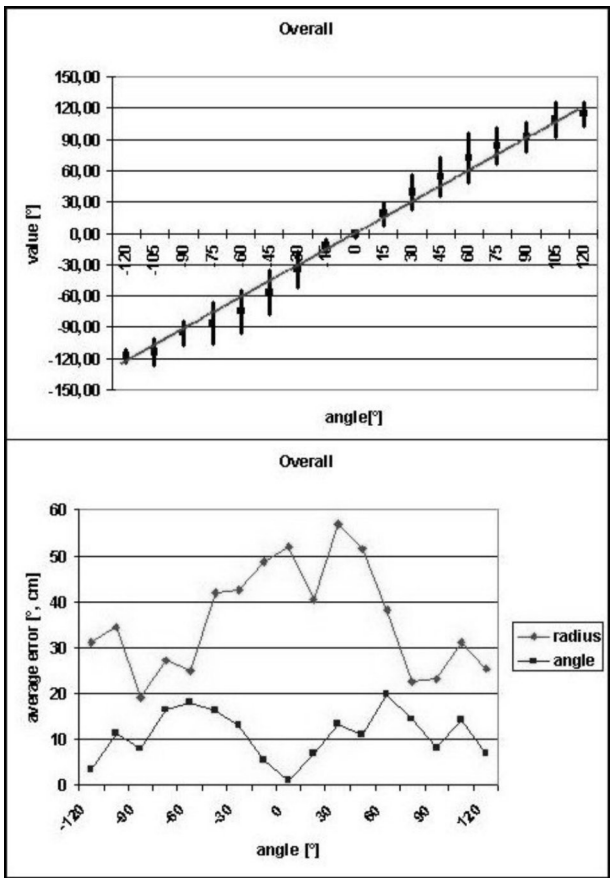

Fig. 9. Estimate results of human viewers. Top: Mean values and standard deviations of the angle estimate over the correct angle. Bottom: Average errors of the radius and angle estimates.

r, the errors are significantly higher for small angle values compared to large angle values. The errors for ϕ behave inversely, being small for small angle values, then getting bigger with increasing angle value. This trends can be easily explained geometrically. For angles greater than 90°, the errors decrease again. This is due to the fact that pointing to a target behind one's position results in a significant change of body pose: The shoulder is turned backwards, which is clearly visible in the images.

Overall, in 50.06% of all cases, the human viewers estimated ϕ correctly within a tolerance of 10°. For r, 76.27% of all trials were within a tolerance of 50 cm. These results give a hint for evaluating the following results of our neural estimator, keeping in mind that the presented data - the distorted monocular images - are very unfamiliar for a human.

B. Results of the Neural Estimator Cascade

In the following experiments, the face position was labelled manually in the test images. This eliminates the influences of errors in the automatic head-shoulder-detection and therefore shows the capabilities and properties of the developed ROI extraction algorithm and the neural estimator cascade. Fig. 10 shows the classification results of each cascade stage for three test subjects, with 90 test cases covering the whole valid pointing area per subject. The right subject (P2) is taken from the training data set. All results mentioned in the following passages refer to the two remaining subjects not included in the training data.

The left/right classifier stage yields classification rates of almost 100% for all subjects. This is especially important since further processing of the input image depends on the results of this stage, and misclassifications will lead to a totally erroneous target estimate. The radius estimator stage shows a good overall performance, with 84.4% and 96.7% of the samples within the 50 cm tolerance and classified into the correct radius class. Compared to this, the angle estimator stages perform poorly: While the performance of the coarse angle classifier stage is very good for all subjects, the fine angle estimate is not, with 66.7/80% of samples within the 10° tolerance, respectively. These results show that the angle estimate is the major problem, limiting the performance and accuracy of the developed pointing direction estimator.

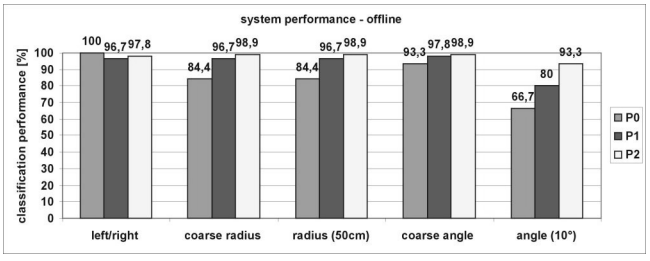

Fig. 10. Classification results of the different stages of the estimator cascade for three test subjects. Left to right: left/right classifier, radius classifier (radius classes), radius estimate (tolerance 50cm), coarse angle classifier, angle estimate (tolerance 10°)

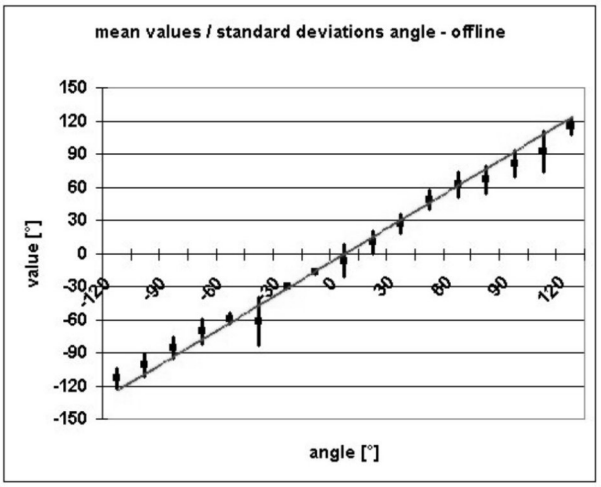

Fig. 11. Mean values and standard deviations of the angle estimate over the correct angle determined on hand-labelled data. The ideal case is depicted by the dotted line.

For comparison with Fig. 9, the diagram for the mean values and standard deviations of the angle estimate is given in Fig. 11. The results are close to the optimal straight line with small standard deviations for most angles. However, the effects of errors in early stages of the classifier cascade become visible, resulting in larger errors and deviations for certain angles. Fig. 12 shows the average errors of the angle and radius estimates. The behavior of the angle estimate is quite similar to that observed in Fig. 9. The radius estimate behaves almost inversely to that observed before, apart from the large errors for 0°.

Looking on Fig. 10 again, we see that the neural estimator achieved a classification rate of 66.7% and 80% respectively for the fine angle estimate with a tolerance of 10°, and 84.4% and 96.7% for the radius estimate with a tolerance of 50 cm. This is significantly better than the results achieved by human viewers (50.06 / 76.27%). But of course, the latter

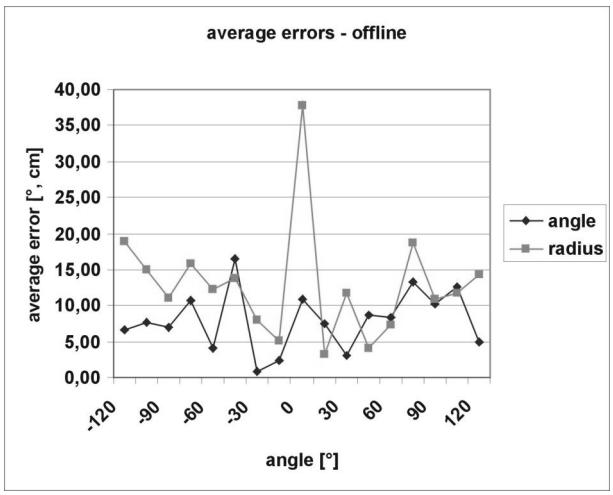

Fig. 12. Average errors for radius and angle estimate over the correct angle.

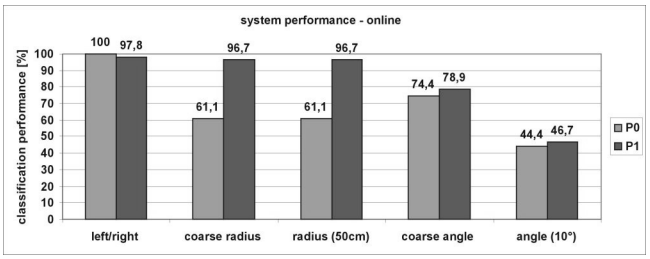

Fig. 13. Online classification results of the different stages of the estimator cascade for 2 test subjects. Left to right: left/right classifier, radius classifier (radius classes), radius estimate (tolerance 50cm), coarse angle classifier, angle estimate (tolerance 10 ˚)

are more reliable in the sense that they don't produce outliers and large errors.

When interpreting this results, we have to keep in mind that they were achieved with an "ideal" head detection. Fig. 13 shows the classification performance of the classifier stages for the two test persons when the mentioned Viola-Jones head-shoulder-detector is used. Obviously, the performance is significantly worse. This shows that of all possible error sources, the head-shoulder detection is the most crucial: Deviations of a few pixels from the "ideal" position may already lead to great errors in the final target estimate since the extracted ROI is then deviating strongly from the training data.

To determine the overall precision of the presented system online on HOROS, a random target approach experiment was conducted: The operator pointed to randomly selected targets in his local surroundings, and the robot navigated to the positions estimated by the neural cascade. In this experiment, all errors resulting from the head-shoulder detection, the neural estimator, and the robot's navigation system are superimposed. From a total of 72 trials, 6 (8.3%) were totally erroneous outliers. The remaining trials yielded an average position error of 59 cm. 28 (38.9%) were within 50 cm, 31 (43.1%) within 1 meter, and 7 (9.7%) more than 1 meter from the target point.

V. Conclusion and future work

In this paper, we presented a neural classifier cascade for estimating a referred target point out of a pointing pose. Although we only use monocular image data of relatively poor quality, the system works in principle, achieving an accuracy better than that of a human viewer on the same data. One goal of our project was to determine whether it is possible to implement a target point estimator using only monocular image data. Our results suggest that this is possible, however, with restrictions with respect to the accuracy needed for real world scenarios. There are several starting points for improvements to our system that are to be investigated in the near future:

First, the used feature extraction routine (Gabor filtering using an equidistant grid) is very simple. Several more sophisticated methods for feature extraction and representation are imaginable that may lead to better results. For example, a foreground extraction routine could be applied, segmenting the pointing person from the background and thus limiting disturbing background influences. Second, the MLP classifiers in our cascade could be replaced by other function approximators. On first view, an MLP seems well suited for tackling the problem, but it is possible that the input data distribution shows characteristics that can be better represented with another type of neural function approximator. Third, the available input data might be improved in quality or extended integrating additional sensory cues. This seems the most promising field because we believe that the input data is our main problem, limiting the achievable accuracy. Fourth, further efforts are necessary to improve the accuracy of the head-shoulder detection preceding the target estimation. For example, the existing Viola-Jones cascade could be used to limit the search area for a subsequent more specialized detector.

So far, we only evaluated the performance of our target point estimator on single images of the final pointing pose. An interesting question is whether the dynamic movement of the pointing arm to the final pose contains additional information that could be exploited to enhance the precision of the estimator. In a first experiment, we utilized a Kalman filtering algorithm with a very simple system model. The results suggest that this could indeed improve the estimator performance, especially the accuracy of the angle estimate. However, further investigations are required on this topic.

References

[1] Rogalla, O., Ehrenmann, M., Zoellner, R., Becher, R., and Dillmann, R., "Using gesture and speech control for commanding a robot assistant", in Proc. IEEE Int. Workshop on Robot and Human Interactive Communication 2002 (ROMAN2002), pages 454–459.

[2] Licsar, A. and Sziranyi, T., "Hand gesture recognition in camera-projector system", in Int. Workshop on Human-Computer Interaction, Lecture Notes in Computer Science, volume 3058, pages 81–91.

[3] Triesch, J. and v.d. Malsburg, C., "A system for person-independent hand posture recognition against complex backgrounds", in IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 23 No.12, pages 1148–1153.

[4] Triesch, J. and v.d. Malsburg, C., "Classification of hand postures against complex backgrounds using elastic graph matching", in Image and Vision Computing, volume 20, pages 937–943.

[5] Alon, J., Athitsos, V., Yuan, Q., and Sclaroff, S, "Simultaneous localization and recognition of dynamic hand gestures", in Proc. IEEE Workshop on Motion and Video Computing 2005, pages 254–260.

[6] Rahman, M. M. and Ishikawa, S., "Appearance based representation and recognition of human motions", in Proc. Int. IEEE Conference on Robotics and Automation 2003, pages 1410–1415.

[7] Jojic, N., Brumitt, B., Meyers, B., Harris, S., and Huang, T., "Detection and estimation of pointing gestures in dense disparity maps", in Proc. Int. Conference on Automatic Face and Gesture Recognition 2000, pages 468–475.

[8] Noelker, C. and Ritter, H., "Illumination independent recognition of deictic arm postures", in Proc. 24th annual conference of the IEEE Industrial Electronics Society 1998, pages 2006–2011.

[9] Nickel, K. and Stiefelhagen, R., "Real-time recognition of 3d pointing gestures for human-robot-interaction", in DAGM-Symposium 2003, pages 557–565.

[10] Viola, P. and Jones, M., "Rapid object detection using a boosted cascade of simple features", in Proc. Conference of Computer Vision an Pattern Recognition 2001, volume 1, pages 511–518.

[11] Martin, C., Schaffernicht, E., Scheidig, A., and Gross, H.-M., "Sensor fusion using a probabilistic aggregation scheme for people detection and tracking", in Proc. 2nd European Conference on Mobile Robots 2005 (ECMR2005), Ancona, Italy, pages 176–181.