Estimation of Pointing Poses on Monocular Images with Neural Techniques - An Experimental Comparison

Frank-Florian Steege, Christian Martin, and Horst-Michael Groß

Department of Neuroinformatics and Cognitive Robotics,
Ilmenau Technical University, Ilmenau, Germany
frank-florian.steege@stud.tu-ilmenau.de, christian.martin@tu-ilmenau.de
http://www.tu-ilmenau.de/neurob

Abstract. Poses and gestures are an important part of the nonverbal inter-human communication. In the last years many different methods for estimating poses and gestures in the field of Human-Machine-Interfaces were developed. In this paper for the first time we present an experimental comparison of several re-implemented Neural Network based approaches for a demanding visual instruction task on a mobile system. For the comparison we used several Neural Networks (Neural Gas, SOM, LLM, PSOM and MLP) and a k-Nearest-Neighbourhood classificator on a common data set of images, which we recorded on our mobile robot HOROS under real world conditions. For feature extraction we use Gaborjets and the features of a special histogram on the image. We also compare the results of the different approaches with the results of human subjects who estimated the target point of a pointing pose. The results obtained demonstrate that a cascade of MLPs is best suited to cope with the task and achieves results equal to human subjects.

1 Introduction and Motivation

In recent years the Human-Machine Interaction has reached a large importance. One of the most important and informative aspects of nonverbal inter-human communication are gestures and poses. In particular, pointing poses can simplify communication by linking speech to objects or locations in the environment in a well-defined way. Therefore, a lot of work has been done in recent years focusing on integrating pointing pose estimation into Human-Machine-Interfaces.

Numerous approaches, which can estimate the target of such a pointing pose have been developed in recent years. Our goal is to provide an approach, which can be used to estimate a pointing pose on a mobile robot by means of low-cost sensors. Therefore, in this paper we refer only to approaches using monocular images to capture the pose of the user. Second, approaches that do not use Neural Networks to estimate the target of the pointing pose like Haasch [1], who used an object-attention system and a skin color map or Nickel [2], who estimated the target by the use of a virtual line through the tracked hand and head of the user, are also not considered in this paper.

J. Marques de Sá et al. (Eds.): ICANN 2007, Part II, LNCS 4669, pp. 593–602, 2007.

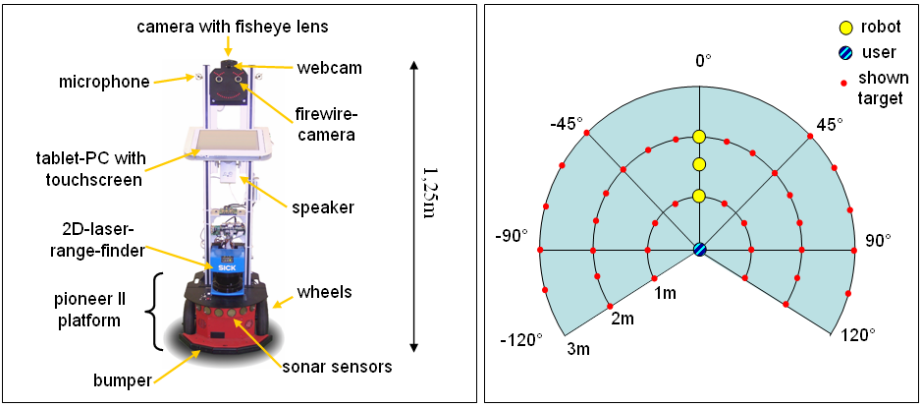

Fig. 1. (left) Our robot HOROS, used for experimental investigation of the pointing pose estimation is shown. The images for the estimation of the pointing target were taken with the firewire camera (located in the right eye). (right) The configuration used for recording the ground truth training and test data. The subject stood in front of the robot and pointed at one of the marked targets on the ground in a distance of 1 to 3 m from the subject. The distance of the robot to the subject varied between 1 m and 2 m.

However there are several approaches that utilize different Neural Networks to estimate the pointing pose. Nölker and Ritter [3] used Gaborfilters in combination with 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. Richarz et al. [4] recently also used Gaborfilters on monocular images and a cascade of Multi-Layer Perceptrons (MLP) as function-approximator to determine the target point of a pointing-pose on the ground. Takahashi [5] suggested to use a special kind of histogram features in combination with a SOM to estimate the pose of a person in an image. Finally, since the head pose is typically also important for a pointing pose, approaches estimating the head pose are also considered in this paper: Krüger and Sommer [6] utilized Gaborfilters and a LLM to estimate the head pose, while Stiefelhagen [7] presented a system that works on edge-filtered images and uses a MLP for head pose estimation.

All these approaches achieved more or less good results for their particular task, but can not be compared with each other, because they use different images captured in different environments and they use different combinations of methods for feature extraction as well as different Neural Networks for approximating the target point or the direction of the pose.

Therefore, for this paper we implemented and compared several selected neural approaches, all trained and tested with the same set of training and test data. In this way we give an overview of the suitability of the different approaches for the task of estimating a pointing pose on a monocular image. The referred approaches suggest different applications for the recognition of a pointing pose. In our comparison we choose an application where a user points at a target on

the ground which is similar to the application Richarz [4] suggested. We implemented this approach on our mobile robot HOROS (**HO**me **RO**bot **S**ystem, see Fig. 1 left), making it navigate to the specified targets, thus enabling a user to control the robot only by means of pointing.

The remainder of this paper is organized as follows: First, in Sect. 2 we give an overview of our test environment used to obtain the training and test data for our comparison. In Sect. 3, the preprocessing steps performed on every image and the methods for feature extraction we used in our approach are explained. In Sect. 4, we shortly describe the Neural Network techniques we compare in our approach. Section 5 describes the experimental investigations we conducted and compares the results of the different approaches. We conclude with a summary in Sect. 6 and give a perspective on possible improvements we plan to investigate in the near future.

2 Training-Data and Ground-Truth

We encoded the target points on the floor as (r, φ) coordinates in a user-centered polar coordinate system (see Fig. 1). This requires a transformation of the estimated target 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 m and a value range for φ from $-120°$ to $+120°$. The $0°$ direction is defined as 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. Figure 2 shows the configuration we chose for recording the training data and our robot HOROS which was used to record images of the subjects. 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 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. 2).

All captured images are labeled with the distance of the subject and the radius r and angle φ of the target, thus representing the ground truth used for

Fig. 2. Typical examples of images of subjects taken by the frontal camera of the robot in several demanding real world environments with background clutter. The left three images are from the trainig data, the right three images from the test data.

training and also for the comparison with humans as pointing pose estimators (see Section 5). This way, we collected a total of 2,340 images of 26 different interaction partners in demanding real world environments with background clutter. 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.

3 Image Preprocessing and Feature Extraction

Since the interaction partners standing in front of the camera can have different heights and distances, an algorithm had to be developed that can calculate a normalized region of interest (ROI), resulting in similar subimages for subsequent processing. We use an approach suggested by [4] to determine the ROI by using a combination of face-detection (based on the Viola & Jones Detector cascade [8]) and empirical factors. With the help of a multimodal tracker [9] 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. Then a histogram equalization is applied. The preprocessing operations used to capture and normalize the image are shown in Fig. 3. Since some of the approaches mentioned in Sect. 1 use a Background Subtraction ([5], [7]) while others do not ([3], [4] and [6]) we optional use a Background Subtraction to test its influence on the pose estimation result.

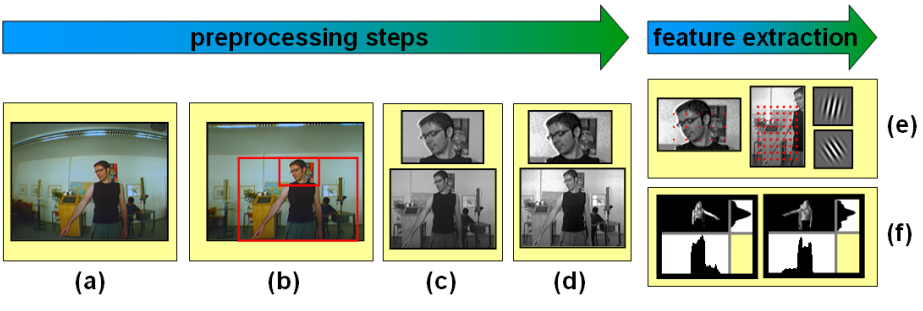

Fig. 3. Steps of preprocessing and feature extraction: the raw distorted image of the camera (a) is transformed into an undistorted image (b) and the face of the user is detected by means of [8]. Based on the height of the face in the picture and the distance of the user, two sections of the image are extracted and transformed into grayscale images (c). On these images a histogram equalization is used (d). Subsequently features are extracted in different ways. First, Gaborfilters placed at defined points of the image (marked as dots in (e)) were used. The second approach is to count how often pixel belonging to a pre-segmented user appear in every row and column of the image (f). A Background Subtraction can optionally be used between steps (d) and (e).

On the normalized image regions we extracted features for the approximation of the target position the user is pointing to. We therefore compared two possible methods. First, we used Gaborfilters of different orientations and frequencies bundled in Gaborjets that we located on several fixed points in the image sections. Gaborjets are also used in the approaches of [3], [4] and [6]. Second, we reimplemented the approach presented by [5]. Based on a background model, in this case, we could subtract the background from the image and count the number of pixels which belong to the user in every row and column. The several steps of preprocessing and feature extraction used in our comparison are shown in Fig. 3.

4 Used Techniques for Approximation of the Target

One objective of our approach is the experimental comparison of selected Neural Network based pointing pose estimators including a simple k-Nearest-Neighbour method well known as reference technique. In the following, the different methods used for comparison are presented:

k-Nearest-Neighbour Classification: 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 allows only classification and not an approximation between the labels of two or more neighbours. Therefore, we slightly modified the method in our approach in a SoftMax-manner where the label for the input $f_k(\mathbf{x})$ is determined as follows:

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

In this way, the labels l_i of the k nearest neighbours contribute to the output and are weighted by their Euclidian distance d_i to the input.

Neural Gas: A Neural Gas network (NG, [10]) approximates the distribution of the inputs in the feature space by a set of adapting reference vectors (neurons). The weights $\mathbf{w}_i$ 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 $D_{\mathbf{X}} = \{\|\mathbf{x} - \mathbf{w}_i\|, i = 1, \cdots, N\}$ associated with an input signal $\mathbf{x}$. Each time an input signal $\mathbf{x}$ is presented, the ordering of the elements of the set $D_{\mathbf{X}}$ 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, [11]). 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 $D_{\mathbf{X}}$. We modified the SOM so that every neuron has a learned label as we did with the NG above.

Local Linear Map: The Local Linear Map (LLM, [12]) 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 which each represent a pair of reference vectors $(\mathbf{w}_i^{in}, \mathbf{w}_i^{out})$ in the in- and output-space and an associated only locally valid linear mapping $\mathbf{A}_i$. The answer $\mathbf{y}_{bm}$ of the best-matching neuron of the LLM to an input $\mathbf{x}$ is calculated as follows:

$$\mathbf{y}_{bm} = \mathbf{w}_{bm}^{out} + \mathbf{A}_{bm} \left(\mathbf{x} - \mathbf{w}_{bm}^{in} \right) \tag{2}$$

The weights $\mathbf{w}_i^{in}, \mathbf{w}_i^{out}$ and the mapping matrix $\mathbf{A}_i$ have to be learned during the training process. See [12] for more details.

Parametrized Self-organizing Map: Like the LLM, a Parametrized Self-Organizing Map (PSOM, [13]) is also an extension of the SOM. While the LLM computes only a linear approximation of the output, a PSOM uses a set of non-linear basis manifolds to construct a mapping through the reference vectors $\mathbf{a}$. A basis function $H(\mathbf{s}, \mathbf{a})$ is associated with each reference vector $\mathbf{a}$. These basis functions realize a smooth interpolation of intermediate positions between the reference vectors. The interpolation is an iterative process starting at the best-matching reference vector. The topological order of the reference vectors has to be provided for the organization of the PSOM. In our approach we use a SOM to obtain this topological order.

Multi-layer Perceptron: For our comparison we used a cascade of several MLPs as decribed in [4]. 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 Results of Comparing Experimental Investigations

To have a simple reference for the quality of the estimation, 10 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 subject 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 estimation systems. 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. The results of the human based reference experiments are included in Fig. 4 and Fig. 5. The label *Human 2D* refers to the experiments on the computer screen and the label *Human 3D* refers to the results under real world conditions.

radius estimation

correct samples in % / mean error in m	k-NN	NG	SOM	LLM	PSOM	MLP
Gaborfilters	48,18 % / 0,314 m	33,85 % / 0,458 m	42,93 % / 0,443 m	54,45 % / 0,378 m	29,04 % / 0,507 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	58,05 % / 0,305 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	47,44 % / 0,455 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**	72,79 % / 0,267 m	**88,40 % / 0,138 m**
Histogram Features and BG Subtraction	64,63 % / 0,313 m	51,13 % / 0,399 m	45,86 % / 0,491 m	70,37 % / 0,357 m	26,76 % / 0,619 m	77,01 % / 0,205 m

Human 3D	Human 2D
84,25 % / 0,080 m	75,00 % / 0,350 m

angle estimation

correct samples in % / mean error in °	k-NN	NG	SOM	LLM	PSOM	MLP
Gaborfilters	23,10 % / 23,00 °	13,91 % / 23,20 °	15,63 % / 23,61 °	21,61 % / 21,79 °	11,97 % / 26,34 °	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 °	18,35 % / 25,02 °	50,91 % / 17,23 °
Gaborfilters and Discriminant Analysis	29,41 % / 23,05 °	19,36 % / 22,23 °	20,70 % / 23,39 °	24,73 % / 23,77 °	16,28 % / 25,09 °	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 °	20,90 % / 23,76 °	57,28 % / 15,64 °
Histogram Features and BG Subtraction	35,48 % / 18,25 °	28,59 % / 17,35 °	23,91 % / 19,39 °	40,67 % / 15,52 °	15,54 % / 24,29 °	51,00 % / 15,75 °

Human 3D	Human 2D
74,66 % / 4,50 °	50,00 % / 13,76 °

Fig. 4. Results for the estimation of the radius (top) and the angle of the target position (bottom). For each method the percentage of the targets estimated correctly and the mean error is determined. At the right side, the results of the human viewers (2D on computer screen, 3D in reality) are given for comparison. Methods that achieve a result equal to that of the human viewers are marked with a shaded background.

The results of the several neural approaches for estimating the target position are shown in Fig. 4 and Fig. 5. As described in Sect. 2 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. 4. For the correct estimation of the target point, r as well as φ had to be estimated correctly. We defined the estimation result being correct if r differed less than 50 cm from the ground truth radius and φ differed less than 10° from the ground truth angle. Figure 5 shows the results for a correct estimation of both values.

Every of the six selected approaches was trained and tested on the same training data set. For each system, we used five 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 a Linear Discriminant Analysis (LDA) executed over all predefined Gaborfilter positions. Fourth, we combined Gaborfilter, Background Subtraction and utilized only the relevant features extracted by the Discriminant Analysis mentioned above. In the last setup, we did not apply the Gaborfilters but the column and row histograms of the presegmented persons in the images as proposed in [5].

target point estimation (correct radius ***and*** correct angle)							
correct samples in %	k-NN	NG	SOM	LLM	PSOM	MLP	Human 3D
Gaborfilters	11,12%	4,70%	6,70%	11,76%	3,47%	29,16%	62,90%
Gaborfilters and BG Subtraction (BGS)	22,28%	17,72%	15,34%	23,53%	10,65%	44,90%	Human 2D
Gaborfilters and Discriminant Analysis	17,69%	9,38%	11,66%	16,04%	7,72%	28,14%	37,50%
Gaborfilters, BGS and Discriminant Analysis	34,72%	22,66%	23,66%	31,74%	15,21%	50,63%	
Histogram Features and BG Subtraction	22,93%	14,61%	10,96%	28,61%	4,15%	39,27%	

Fig. 5. 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. Unlike Fig. 4, showing the separate results for the estimation of r and φ, here the results for the correct estimation of both values are shown. As in Fig. 4 the results of the human viewers (2D on computer screen, 3D in reality) are given for comparison.

computation time							
computation time in ms	k-NN	NG	SOM	LLM	PSOM	MLP	allowed max.
Gaborfilters	285 ms	72 ms	80 ms	63 ms	3308 ms	54 ms	80 ms
Gaborfilters and BG Subtraction (BGS)	300 ms	87 ms	95 ms	78 ms	3323 ms	63 ms	
Gaborfilters and Discriminant Analysis	137 ms	28 ms	33 ms	26 ms	1330 ms	21 ms	
Gaborfilters, BGS and Discriminant Analysis	129 ms	38 ms	42 ms	35 ms	995 ms	31 ms	
Histogram Features and BG Subtraction	224 ms	33 ms	40 ms	29 ms	1624 ms	21 ms	

Fig. 6. The computation times of the different methods. A method capable of running with a minimum of 12.5 images per second on our mobile robot has to process one image in less than 80 ms (Athlon 2800, SUSE Linux).

These results demonstrate, that a cascade of several MLPs as proposed in [4] 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. 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 humans on 2D screens, but it is not able to estimate the angle as good as humans in a real world setting. This is because

the estimation of the depth of a target in a monocular image is difficult for both, human and function approximators.

In our experimental comparison, the LLM and the MLP deliver a better result than the SOM and the Neural Gas. We suppose this result is caused by the ability of the MLP and the LLM to better approximate the output function in regions with few examples. The cascade structure of the MLP approach as proposed in [4] makes it possible to estimate φ better than the other approaches. However, since r is estimated by a single MLP and the MLP-result for r is better than that of the other approaches, we believe that a cascade organization of the other Neural Networks would not lead to a better result than that achieved by the MLP cascade. The PSOM delivers a relatively bad result in comparison to the other approaches. This is based on the fact, that only few basis points could be used due to the very long computation time of the PSOM. Figure 6 finally shows the computation times of all methods. Except the k-NN and the PSOM, all methods are able to process more than 12.5 images per second at the robot's on board PC. The k-NN method needs a long running due to the many comparisons which are needed to get the best neighbours to given observations. The computation time of the PSOM is especially high because of the iterative gradient descent along the PSOM structure that is needed to get the best suited output.

6 Conclusion

We presented an experimental comparison of several re-implemented Neural Network based approaches for a demanding visual instruction task on a mobile system. Since our goal is to provide an approach, which copes with the task by means of low-cost sensors, we refered to approaches using monocular images. Of the relevant approaches a cascade of Multi-Layer Perceptrons proved to be best suited for this task. All methods profit from the use of a Background Subtraction and the information delivered by a Discriminant Analysis. The comparison of the different methods had shown, that the usage of Gaborfilters for feature extraction leads to better results than the histogram based features. The best system is able to estimate the radius r of the target point better than human subjects do, but there are still problems in estimating the angle φ of the target due to the use of monocular images. This problem could be reduced by means of a stereo camera, which delivers the lacking depth information. Possibly the angle of the estimated target might not be as important if an other application is chosen, for example, if the user is pointing at certain objects in the surroundings allowing a model-based pointing pose specification instead of a non-specific target point on the ground.

References

1. Haasch, A., Hofemann, N., Fritsch, J., Sagerer, G.: A Multi-Modal Object Attention System for a Mobile Robot. In: Int. Conf. on Intelligent Robots and Systems, pp. 1499–1504 (2005)

2. Nickel, K., Stiefelhagen, R.: Recognition of 3D-Pointing Gestures for Human-Robot-Interaction. In: European Conference on Computer Vision, pp. 28–38 (2004)
3. Nölker, C., Ritter, H.: Illumination Independent Recognition of Deictic Arm Postures. In: Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, Aachen, pp. 2006–2011 (1998)
4. Richarz, J., Martin, C., Scheidig, A., Gross, H-M.: There You Go! - Estimation Pointing Gestures in Monocular Images for Mobile Robot Instruction. In: Int. Symposium on Robot and Human Interactive Communication, pp. 546–551 (2006)
5. Takahashi, K., Tanigawa, T.: Remarks on Real-Time Human Posture Estimation from Silhouette Image using Neural Network. In: Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics: The Hague, pp. 370–375 (2004)
6. Krüger, V., Sommer, G.: Gabor Wavelet Networks for Efficient Head Pose Estimation. In: Image and Vision Computing 20(9-10), 665–672 (2002)
7. Stiefelhagen, R.: Estimating Head Pose with Neural Networks - Results on the Pointing04 ICPR Workshop Evaluation. In: Pointing04 ICPR, Cambridge, UK (2004)
8. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. Proc. Conf. of Computer Vision and Patter Recognition 1, 511–518 (2001)
9. Gross, H.-M., Richarz, J., Mller, St., Scheidig, A., Martin, Chr.: Probabilistic Multimodal People Tracker and Monocular Pointing Pose Estimator for Visual Instruction of Mobile Robot Assistants. In: Proc. IEEE World Congress on Computational Intelligence (WCCI 2006), pp. 8325–8333 (2006)
10. Martinetz, T., Schulten, K.: A Neural-Gas Network Learns Topologies. In: Proc. of the ICANN 1991. Helsinki, pp. 397–402 (1991)
11. Kohonen, T.: Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics 43, 59–69 (1982)
12. Ritter, H.: Learning with the Self-Organizing Map. In: Kohonen, T., et al. (eds.) Artifical Neural Networks, pp. 379–384. Elsevier Science Publishers, Amsterdam (1991)
13. Walther, J.A., Ritter, H.: Rapid Learning with Parametrized Self-Organizing Maps. Neurocomputing 12, 131–153 (1996)