

# Integrating Context-Free and Context-Dependent Attentional Mechanisms for Gestural Object Reference

Gunther Heidemann<sup>1</sup>, Robert Rae<sup>2</sup>, Holger Bekel<sup>1</sup>,  
Ingo Bax<sup>1</sup>, and Helge Ritter<sup>1</sup>

<sup>1</sup> Neuroinformatics Group, Faculty of Technology, Bielefeld University,  
Postfach 10 01 31, D-33501 Bielefeld, Germany

{gheidema,hbekel,ibax,helge}@techfak.uni-bielefeld.de  
[http://www.TechFak.Uni-Bielefeld.DE/ags/ni/index\\_d.html](http://www.TechFak.Uni-Bielefeld.DE/ags/ni/index_d.html)

<sup>2</sup> Now at *PerFact Innovation*, Lampingstr. 8,  
D-33615 Bielefeld, Germany

{robrae}@techfak.uni-bielefeld.de

**Abstract.** We present a vision system for human-machine interaction that relies on a small wearable camera which can be mounted to common glasses. The camera views the area in front of the user, especially the hands. To evaluate hand movements for pointing gestures to objects and to recognise object reference, an approach relying on the integration of bottom-up generated feature maps and top-down propagated recognition results is introduced. In this vision system, modules for context free focus of attention work in parallel to a recognition system for hand gestures. In contrast to other approaches, the fusion of the two branches is not on the symbolic but on the sub-symbolic level by use of attention maps. This method is plausible from a cognitive point of view and facilitates the integration of entirely different modalities.

## 1 Introduction

One of the major problems in human-machine interaction is to establish a common focus of attention. In current computer systems the mouse is used as an input device to select the windows to which keystrokes refer, which can be looked upon as a simple means to establish a common focus of attention. However, when human-machine interaction refers to real world objects or does not take place in front of a terminal, computer vision will be needed. In this case, *hand gestures* are most natural to guide attention of the machine.

The problem with hand gestures is that they are not precise according to the requirements of a machine. Humans do not point with high angular accuracy, instead, they rely (*i*) on the understanding of the dialog partner and (*ii*) on supplementing modalities like speech. Hence, it is not sufficient for visual evaluation of pointing gestures to calculate direction angles as accurately as possible.

In this contribution, we present a system which uses an *attention map* as a representation of focus of attention. The attention map allows integration of

entirely different modalities and thus facilitates solution of the above mentioned problems: (i) The machine needs a basic understanding of the scene. Only by this means the “continuous valued problem” of evaluating pointing directions (which might point everywhere) with possibly high accuracy can be transformed into a “discrete valued problem”. In the latter case, the machine analyses the scene for salient points or regions and thus restricts the possible pointing directions to a small subset of the whole angular range. In the system proposed here, several context-free attentional mechanisms (entropy, symmetry and edge-corner detection) are adopted to activate certain areas of the attention map and thus establish an “anticipation” of the system where the user might point to. (ii) The attention map allows the future integration of symbolic information from speech recognition systems. Hints like “right” or “above” can easily be expressed in terms of the manipulator maps outlined in sections 3 and 4.2.

An earlier version of the approach was successfully applied in human machine interaction [5]. It is related to the data driven component of the attention system introduced by Backer et al. [1], which combines several feature maps for gaze control of an active vision system. Similarly motivated architectures for focus of attention were proposed by Itti et al. [10] and Walther et al. [23]. A system for hand tracking and object reference that also allows the integration of modalities other than vision was proposed by Theis et al. [21] for the Cora robot system.

First we will describe the experimental setup and the image processing architecture, then the single context-free attentional features and their adaptive weighting, and finally the integration of the pointing direction recognition.

## 2 System description

### 2.1 Scenario: Gestural reference

The experimental setup is part of the VAMPIRE project (Visual Active Memory Processes and Interactive REtrieval) of the IST programme. The project aims at the development of an active memory and retrieval system in the context of



**Fig. 1.** Left: Miniature camera mounted to glasses. Middle: User points at an object (a button of the power supply unit on the table). Right: Setup used for evaluation, see section 5.

an Augmented Reality scenario. The user wears a head-mounted camera like in Fig. 1 and a display such that the system is able to build up a hierarchically structured memory of what the user sees. In future work, query results will be re-visualised in a virtual world or projected into the real world using the head-mounted display.

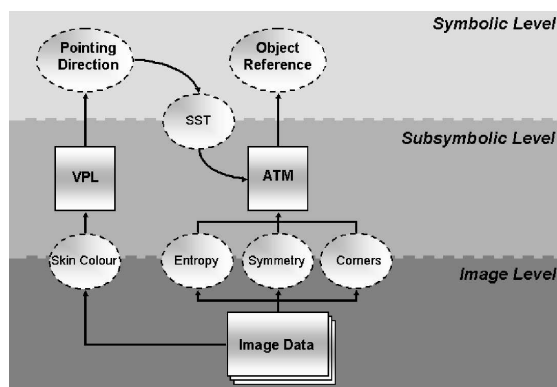
In the VAMPIRE scenario user-machine interaction is possible only using vision and speech, so recognition of gestural reference to memorised as well as *unknown* objects is a key ability. Therefore goal-oriented segmentation techniques are not feasible, instead context-free algorithms have to be used.

In this paper we present part of the attentional subsystem of the VAMPIRE project by which objects or other visual entities can be referenced using hand gestures. As a sample task, we chose the setup of Fig. 1, right, where the user points to different objects on a table. As long as the user changes pointing directions quickly, the system assumes that large object entities are indicated. When movements become slow, the attentional focus established by the detected pointing direction is narrowed down to facilitate reference to details (“virtual laser pointer”), so e.g. a button on a technical device can be selected.

## 2.2 Processing architecture

Figure 2 shows a system overview. From the image captured by the camera first three feature maps are calculated by different modules: Entropy, symmetry and edge-corner detection (section 3.1). In these maps, different image features stand out, for an example see Fig. 5. The attention map module (ATM) calculates a weighted sum of the basic feature maps using an adaptive weighting as described in section 3.2. Maxima of the attention map correspond to areas considered as “interesting” by the system and serve as a preselection of possible pointing targets.

Pointing directions are classified by the neural VPL classification module described in section 4.1 (left branch in Fig. 2). The classifier works on image patches found by a previous skin colour segmentation module and yields as a



**Fig. 2.** System architecture. In contrast to approaches which integrate pointing gesture information and object locations on the symbolic level [2], the pointing angle is down-propagated to the sub-symbolic level using a “symbol-signal-transformer” (SST) and integrated as a spatial weighting of the feature maps.

result (i) a classification whether the patch is an irrelevant object or a pointing hand and in the latter case (ii) an estimate of the 2D pointing angle.

To figure out which image part the user is actually pointing to, knowledge based approaches would process both the pointing angle and the positions of the maxima of the attention map on the symbolic level. A good survey of knowledge based image processing is given in [4].

In contrast, in our approach the pointing direction is transformed back to the sub-symbolic level using a so called “manipulator map”. The manipulator map serves as a multiplicative spatial weighting of the attention map to intensify attention maxima in the pointing direction while inhibiting others. Therefore, the manipulator map shows a cone of high values in the pointing direction, starting at the hand centre (Fig. 3, right). The cone is widened or narrowed depending on the context as described in section 4.2 and can be viewed as a “spotlight of attention”.

### 3 Integrating attention maps

#### 3.1 Generation of context-free feature maps

We use three different context-free methods to determine focus of attention: Local entropy, local symmetry and an edge-corner detector. These methods use entirely different principles to judge saliency and are best suited for different scales. We describe the algorithms ordered by scale “from coarse to fine”.

**Entropy map:** Judging saliency from local entropy relies on the assumption that semantically meaningful areas have also a high information content in the sense of information theory. This method was proposed by Kalinke and von Seelen [13] and has been integrated in a larger vision architecture in [12]. Calculation of an entropy map  $M_E$  is based on a grey value image, usually at low resolution:

$$M_E(x, y) = - \sum_q \tilde{H}(x, y, q) \cdot \log \tilde{H}(x, y, q), \quad \tilde{H}(x, y, q) = \frac{H(x, y, q)}{\sum_{q'} H(x, y, q')}, \quad (1)$$

where  $H$  denotes the histogram within a  $n_E \times n_E$ -window ( $n_E \geq 3$  and odd) around the pixel  $(x, y)$ :

$$H(x, y, q) = \sum_{y'=y-\tilde{n}}^{y+\tilde{n}} \sum_{x'=x-\tilde{n}}^{x+\tilde{n}} \delta_{I(x', y'), q}, \quad (2)$$

with  $\tilde{n} = (n_E - 1)/2$ ,  $\delta$  the Kronecker symbol,  $I(x, y)$  the grey values, and  $q = 0 \dots 2^{Q_E} - 1$ .  $Q_E$  denotes the quantisation of the histogram which should be about  $2^{Q_E}/n_E^2 \approx 10 - 20$ . The crucial parameter in entropy calculation is the window size  $n_E$  in combination with the resolution of the intensity image. It determines the scale on which structures are evaluated. A window which is too small to capture object structure is mainly working as an edge detector. Here, we use windows large enough to direct attention to large objects (Fig. 5).

**Symmetry map:** The second saliency feature is local grey value symmetry as proposed by Reinfeld et al. [17]. While entropy serves for a primary detection of large objects regardless of structure, the symmetry map  $M_{Sym}$  yields a stronger focus to object details which are locally symmetric. The use of symmetry is cognitively motivated by psychophysical findings, see e.g. [3, 15]. For the calculation of  $M_{Sym}$  we use a more efficient version of the original algorithm [17], which can be outlined here only in short.  $M_{Sym}$  relies on the grey value derivatives  $I_x(p), I_y(p)$ , from which the gradient magnitude  $G_I(p) = \sqrt{I_x(p)^2 + I_y(p)^2}$  and direction  $\theta_I(p) = \arctan(I_y(p)/I_x(p))$  are calculated. The symmetry value  $M_{Sym}$  of a pixel  $p$  is a sum over all pixel pairs  $(p_i, p_j)$  within a circular surroundings around  $p$  of radius  $R$ :

$$M_{Sym}(p) = \sum_{(i,j) \in \Gamma(p)} PWF(i, j) \cdot GWF(i, j) \quad \text{with} \quad (3)$$

$$\Gamma(p) = \{(i, j) \mid (p_i + p_j)/2 = p \wedge \|p_i - p_j\| \leq R\}. \quad (4)$$

Gradient directions  $\gamma_i, \gamma_j$  at  $p_i$  and  $p_j$  are judged for the probability to be part of the contours of a symmetric object by the *Phase Weight Function PWF*

$$PWF(i, j) = [1 - \cos(\gamma_i + \gamma_j)] \cdot [1 - \cos(\gamma_i - \gamma_j)], \quad (5)$$

where  $\gamma_i, \gamma_j$  denote the angles between the line  $\overline{p_i p_j}$  connecting  $p_i$  and  $p_j$  and the gradients at  $p_i$  and  $p_j$ , respectively. The *Gradient Weight Function GWF* weights contributions of pixels  $(p_i, p_j)$  higher if they are both on edges because they might relate to object borders:

$$GWF(i, j) = \log(1 + G_I(p_i)) \cdot \log(1 + G_I(p_j)), \quad (6)$$

The logarithm attenuates the influence of very strong edges. Figure 5 shows an example of  $M_{Sym}$ , in which the symmetric buttons can be clearly detected.

**Edge and corner detection:** The third feature map is aimed to yield small, salient details of objects. Since small-scale saliency can hardly be detected from complex image structures, the local grey value gradients  $I_x, I_y$  have to be evaluated for corners and edges. As a detector we chose the method proposed by Harris and Stephens [6], which could be shown to be superior to others in [20]. It is based on an approximation of the auto-correlation function of the signal

$$A(p) = \begin{pmatrix} \langle I_x^2 \rangle_{W(p)} & \langle I_x I_y \rangle_{W(p)} \\ \langle I_x I_y \rangle_{W(p)} & \langle I_y^2 \rangle_{W(p)} \end{pmatrix}, \quad (7)$$

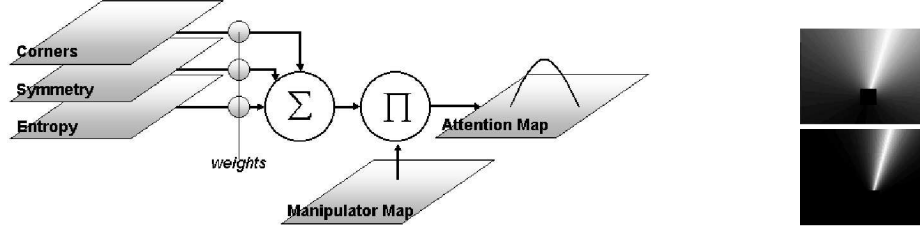
where  $\langle \cdot \rangle_{W(p)}$  denotes a weighted averaging over a window  $W(p)$  centred at  $p$ . The weight function inside the window is a Gaussian. Saliency of a point is high if both eigenvalues of  $A$  are large, however, to reduce the computational effort, the feature map is calculated from

$$M_{Harris}(p) = \det(A) - \alpha \cdot (\text{Trace}(A))^2. \quad (8)$$

Derivatives  $I_x, I_y$  are computed by  $5 \times 5$ -Sobel operators. The Gaussian weighting function for the components of  $A$  inside  $W$  has width  $\sigma = 2$ . As suggested in [20], a value of 0.06 is used for the constant  $\alpha$ .

### 3.2 Adaptive integration algorithm

The adaptive integration of the various maps, in this case consisting of three feature maps  $M_i(x, y)$ ,  $i = 1, 2, 3$ , and one manipulator map  $M_m(x, y)$ , takes place in the “ATM” module as illustrated in Fig. 3.



**Fig. 3.** Left: Processing flow of the ATM-module (central box in Fig. 2). The attention map is generated from an (adaptively) weighted superposition of the feature maps. The manipulator map, which allows the coupling of information from other modules like the pointing direction recognition, is multiplied to the attention map. Right: Examples of manipulator maps (“spotlight of attention”). A wide cone is used as long as the user wants to indicate large objects, a narrow one for precise pointing to details.

The output  $C(x, y)$  is calculated by a weighted summation over the input feature maps and a product of contributing manipulator maps:

$$C(x, y) = \sum_{i=1}^N \theta(w_i * M_i(x, y)) * \prod_{m=1}^l M_m(x, y), \quad (9)$$

with  $\theta(\cdot)$  as a threshold function. The maximum of the output attention map  $C(\cdot, \cdot)$  determines the *point of common attention* of man and machine, which can be used for further processing.

To equalise contributions of all saliency features to the attention map, we calculate the contributions  $S_i$  as a sum over all pixels of each map  $M_i$ . To reach approximate equalisation of the  $S_i$ , the map weights  $w_i$  are adapted by iterating

$$w_i(t+1) = w_i(t) + \epsilon(w_i^s(t) - w_i(t)), \quad 0 < \epsilon \leq 1, \quad (10)$$

with the following target weights  $w_i^s$ :

$$w_i^s = \frac{1}{n^2} \cdot \frac{\sum_{k=1}^n S_k}{S_i} \quad \text{with} \quad S_i = \frac{\sum_{(x,y)} (M_i(x, y) + \gamma)}{\xi_i}. \quad (11)$$

$\gamma$  enforces a limit for weight growing. The parameters  $\xi_i$  can be used if certain saliency features should a priori be weighted higher. In section 4.2 we make use of this possibility to give entropy higher weight for large-scale selection of objects and low weight when object details are pointed at.

## 4 Pointing gesture evaluation

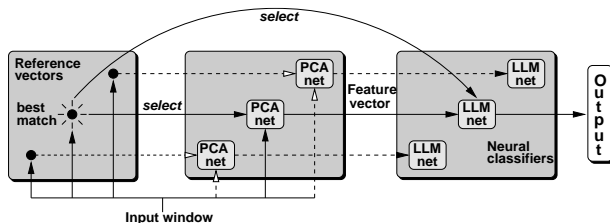
In section 3.1 the contributing context-free feature maps to the attention map  $C(x, y)$  were described. To guide attention to objects or object substructures selected by the user, the pointing direction of the hand has to be evaluated. In a first step, a skin colour segmentation yields a candidate region for the hand. This ROI is subsequently classified by a neural system for two informations: (i) classification into *pointing hand* and *other object* and (ii) recognition of the pointing direction if applicable. We will describe this subsystem first (section 4.1). Since the output of the classifier is on the symbolic level, it has to be “translated” back to the sub-symbolic level as described in section 4.2.

### 4.1 The neural VPL classification system

The classifier is a trainable system based on neural networks which performs a mapping  $\mathbf{x} \rightarrow \mathbf{y}, \mathbf{x} \in \mathbb{R}^D, \mathbf{y} \in \mathbb{R}^N$ . In this case, the input dimension  $D$  is the number of pixels of the skin-segmented windows. The window vector  $\mathbf{x}$  is mapped to a three-dimensional output  $\mathbf{y} \in \mathbb{R}^3$ : Two of the output channels denote the class, one the pointing angle. The class is coded in the first two components in the form  $(1, 0)$  for “pointing hand” and  $(0, 1)$  for “other object”, the third component is the continuous valued pointing angle. Classification of unknown windows  $\mathbf{x}$  is carried out by taking the class  $k$  of the channel with maximal output:  $k = \arg \max_{i=1,2}(\mathbf{y}_i(\mathbf{x}))$ . Only in case of a “pointing hand” the angle  $\mathbf{y}_3$  is relevant.

Training is performed with hand labelled sample windows of the cropped pointing hand plus objects assigned to a rejection class. The rejection class contains other objects which are part of the scenario, e.g. the objects the user points at or parts of the background. In addition, hand postures other than pointing gestures are part of the rejection class, e.g. a fist. So the rejection class is not “universal” but reflects the scenario — a totally unknown object might be mistaken for a pointing hand.

The VPL classifier combines visual feature extraction and classification. It consists of three processing stages which perform a local principal component analysis (PCA) for dimension reduction followed by a classification by neural networks, see Fig. 4. Local PCA [22] can be viewed as a nonlinear extension of simple, global PCA [11]. “VPL” stands for the three stages: **V**ector quantisation, **P**CA and **L**LM-network. The vector quantisation is carried out on the raw



**Fig. 4.** The VPL classifier performs a local PCA for feature extraction and a subsequent neural classification.

image windows to provide a first data partitioning with  $N_V$  reference vectors  $\mathbf{r}_i \in \mathbb{R}^M, i = 1 \dots N_V$ . For vector quantisation we use the Activity Equalisation Algorithm proposed in [8].

To each reference vector  $\mathbf{r}_i$  a single layer feed forward network for the successive calculation of the principal components (PCs) as proposed by Sanger [19] is attached which projects the input  $\mathbf{x}$  to the  $N_P < D$  PCs with the largest eigenvalues:  $\mathbf{x} \rightarrow \mathbf{p}_l(\mathbf{x}) \in \mathbb{R}^{N_P}, l = 1 \dots N_V$ . To each of the  $N_V$  different PCA-nets one “expert” neural classifier is attached which is of the Local Linear Map – type (LLM network), see e.g. [18] for details. It performs the final mapping  $\mathbf{p}_l(\mathbf{x}) \rightarrow \mathbf{y} \in \mathbb{R}^N$ . The LLM network is related to the self-organising map [14] and the GRBF-approach [16]. It can be trained to approximate a nonlinear function by a set of locally valid linear mappings.

The three processing stages are trained successively, first vector quantisation and PCA-nets (unsupervised), finally the LLM nets (supervised). For classification of an input  $\mathbf{x}$  first the best match reference vector  $\mathbf{r}_{n(\mathbf{x})}$  is found, then  $\mathbf{x}$  is mapped to  $\mathbf{p}_{n(\mathbf{x})}(\mathbf{x})$  by the attached PCA-net and finally  $\mathbf{p}_{n(\mathbf{x})}(\mathbf{x})$  is mapped to  $\mathbf{y}$ :  $\mathbf{p}_{n(\mathbf{x})}(\mathbf{x}) \rightarrow \mathbf{y}$ .

The major advantage of the VPL classifier is its ability to form many highly specific feature detectors (the  $N_V \cdot N_P$  local PCs), but needing to apply only  $N_V + N_P - 1$  filter operations per classification. It has been successfully applied to several vision tasks (e.g. [7]), for an application in robotics see [9]. Especially, it could be shown that classification performance and generalisation properties are well-behaved when the main parameters are changed, which are  $N_V, N_P$  and the number of nodes in the LLM nets  $N_L$ .

## 4.2 Translation from symbolic to sub-symbolic level

Skin colour segmentation and the VPL classifier yield the position of the hand  $(x_H, y_H)$  and the pointing direction  $\alpha$ , respectively. Both these (symbolic) informations are translated to a manipulator map  $M_m$  and thus back to the sub-symbolic level. The manipulator map shows a “Gaussian cone” of width  $\sigma_c$  which determines the effective angle of beam spread

$$M_m(x, y) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(\arctan(\frac{y-y_H}{x-x_H}) - \alpha)^2}{\sigma_c^2}\right), \quad (12)$$

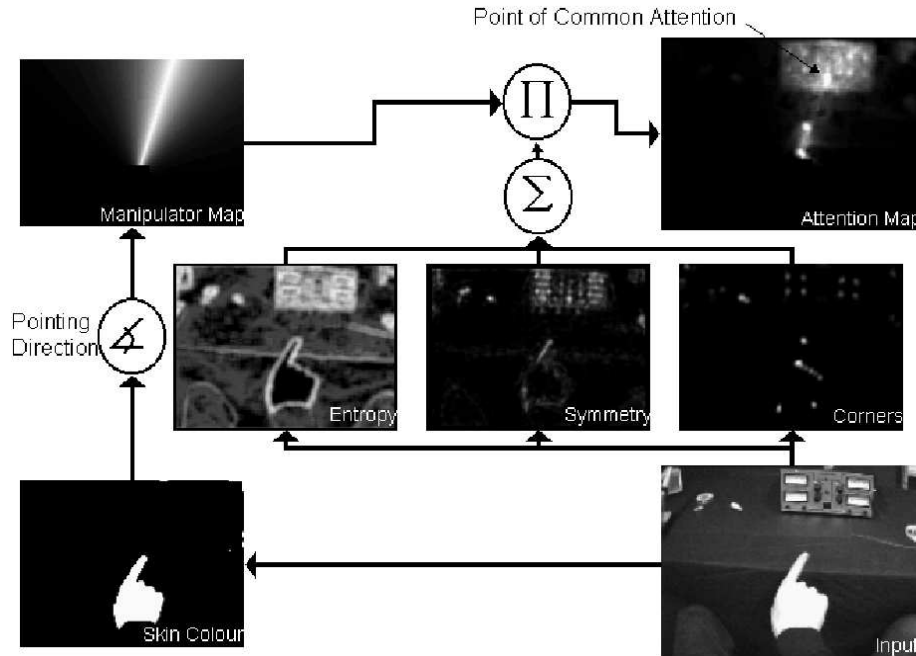
here in the form for the first quadrant for simplicity, see Fig. 3. The cone gives higher weight in the attention map to image regions in the pointing direction and thus “strengthens” salient points in this area.

To facilitate selection of objects on differing scales,  $\sigma_c$  is adjusted online according to the user behaviour. The pointing angles  $\alpha$  and hand positions  $(x_H, y_H)$  are recorded over the last six frames. If they show large variance, it is assumed that the user moves the hand on large scale to select a big object, so also a large  $\sigma_c$  is chosen. In contrast,  $\sigma_c$  is reduced for small variance to establish a “virtual laser pointer” since it is assumed that the user tries to select a detail.

As an additional assistance for coarse / fine selection, the a priori weights  $\xi_i$  of (11) are changed such that the large scale entropy map  $M_E$  dominates for large pointing variance whereas the symmetry map  $M_{Sym}$  and the corner saliency  $M_{Harris}$  are weighted higher for detail selection.

## 5 Results

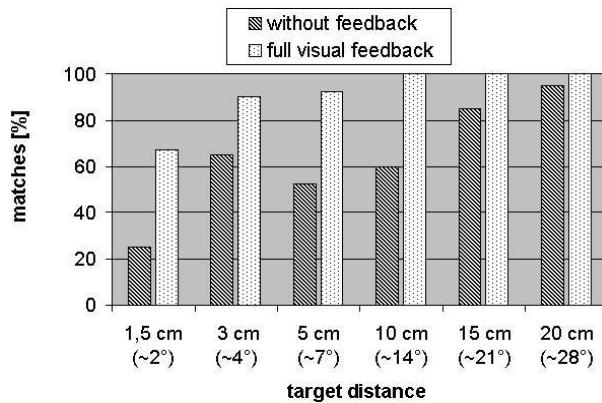
Figure 5 shows a typical result together with intermediate processing stages. The user has just slowed down the pointing movement, so the manipulator map shows a cone of medium width in the pointing direction, starting at the approximate position of the hand centre. Still, weighting of the entropy map is considerable, so the entire object (the power supply unit) stands out in the feature map, but as the user now holds still the smaller-scale features begin to shine through, especially the symmetry of the referenced button.



**Fig. 5.** Processing results for a pointing gesture towards an object. From the input image (bottom right) skin colour is segmented (bottom left), the VPL classifier calculates the angle which is transformed to a manipulator map (top left). The manipulator cone “illuminates” the object, the maxima of the feature maps stand out.

To evaluate the system performance we choose a setup which is based on a “generic” pointing task that can be easily reproduced: A proband has to point

on a row of six white circles on a black board (Fig. 1, right). The diameter of each circle is 12 mm for distances between the circles from 3 to 20 cm (measured from the centre of the circles). To test performance for pointing to details, in an additional experiment circles of diameter 6 mm with a distance of 1.5 cm were used. The distance between the hand and the board is approximately 40 cm, so angular resolutions from  $2^\circ$  to  $28^\circ$  could be tested for targets of about  $0.9^\circ$  or  $1.7^\circ$  angular range, respectively. A supervisor gives the command to point at one of the circles by telling a randomly chosen circle number. We use only the inner four circles to avoid border effects. A match is counted if the system outputs a focus point on the correct circle within three seconds. The results of the experiment under two conditions are shown in Fig. 6. The left and right bars show the result of the condition *without feedback* and *full visual feedback* respectively. Under the condition *full visual feedback* the proband sees the calculated focus points on a screen while pointing. Here the match percentage is quite high even for small target distances, whereas the values decrease substantially under the *without feedback* condition at small distances. Surprisingly the number of matches with a target distance of 3 cm is higher than at 5 and 10 cm. The reason for this behaviour is that the probands tend to carry out more translatory movements of the hand and less changes of the pointing angle for smaller target distances, which leads in some cases for yet unknown reasons to more stable results.



**Fig. 6.** The chart shows the results of the evaluation experiment. The values are averaged for three probands with 20 items each. On the x-axis the distances between the centres of the targets and the corresponding angles for a pointing distance of about 40 cm are shown.

The major result achieved in this test scenario is that system performance can be significantly increased by giving feedback because (i) the user is enabled to adjust single pointing gestures to a target and (ii) the user can adapt himself or herself to the system behaviour. This way the achievable *effective resolution* can be improved, because it does not solely rely on the accuracy of the pointing gesture recognition any more. Since the *full visual feedback* as used in the test scenario is rather unrealistic in natural settings, future work will focus on developing other feedback methods like auditory feedback or visual feedback using a head mounted display.

Still, the system has several limitations: The hand has to be completely visible, otherwise the centre of the skin-segmented blob shifts position so that the VPL classifier gets an unknown input. A “beep” is used as auditory feedback to the user if the hand is too close to the border. Another restriction is that the saliency operators do not yield maxima on all of the objects or not on the desired locations, in particular, edges of strong contrast indicating object boundaries are sometimes weighted higher than object centres.

## 6 Conclusion and Acknowledgement

We have presented a system for visual detection of object reference by hand gestures as a component of a mobile human-machine interface. Feature maps based on different context-free attentional mechanisms were integrated as adaptively weighted components of an attention map. Areas of high “interestingness” in the attention map serve to establish anticipations what the user might be pointing at. A neural classifier gives an estimate of the pointing direction which is integrated using a “manipulator cone” into the attention map.

The functionality of the presented system is not limited to the current scenario. Since arbitrary other saliency features like colour or movement can be integrated, the bottom-up focus of attention can be directed to a wide variety of objects. Even more important is the possibility to transform cues from other modules top-down to the sub-symbolic level. One of the first steps will be the integration of speech-driven cues to generate spatial large scale anticipations.

Still, precision and user independence are big problems in the area of gesture recognition. A major advantage of the new approach is that it does not require high recognition accuracy. This is achieved by the systems anticipation that only salient image points will be selected. In addition, the system offers the possibility to integrate feedback to the user, so in future work we hope to compensate for shortcomings of gesture recognition by using the flexibility of humans to adapt to the machine. In a further advanced version also the user will be enabled to give feedback (“I mean more to the left”) in order to adapt the gesture recognition module to the individual user characteristics online.

This work was supported within the project VAMPIRE (Visual Active Memory Processes and Interactive REtrieval) which is part of the IST programme (IST-2001-34401).

## References

1. G. Backer, B. Mertsching, and M. Bollmann. Data- and Model-Driven Gaze Control for an Active-Vision System. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12):1415–1429, 2001.
2. C. Bauckhage, G. A. Fink, J. Fritsch, F. Kummert, F. Lömker, G. Sagerer, and S. Wachsmuth. An Integrated System for Cooperative Man-Machine Interaction. In *IEEE Int. l Symp. on Comp. Intelligence in Robotics and Automation*, Banff, Canada, 2001.

3. V. Bruce and M. Morgan. Violations of Symmetry and Repetition in Visual Patterns. *Psychological Review*, 61:183–193, 1954.
4. D. Crevier and R. Lepage. Knowledge-based image understanding systems: A survey. *Computer Vision and Image Understanding*, 67(2):161–185, 1997.
5. M. Fislage, R. Rae, and H. Ritter. Using visual attention to recognize human pointing gestures in assembly tasks. In *7th IEEE Int'l Conf. Comp. Vision*, 1999.
6. C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conf.*, pages 147–151, 1988.
7. G. Heidemann, D. Lücke, and H. Ritter. A System for Various Visual Classification Tasks Based on Neural Networks. In A. Sanfeliu et al., editor, *Proc. 15th Int'l Conf. on Pattern Recognition ICPR 2000, Barcelona*, volume I, pages 9–12, 2000.
8. G. Heidemann and H. Ritter. Efficient Vector Quantization Using the WTA-rule with Activity Equalization. *Neural Processing Letters*, 13(1):17–30, 2001.
9. G. Heidemann and H. Ritter. Visual Checking of Grasping Positions of a Three-Fingered Robot Hand. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proc. ICANN 2001*, pages 891–898. Springer-Verlag, 2001.
10. L. Itti, C. Koch, and E. Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
11. I. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 1986.
12. T. Kalinke and U. Handmann. Fusion of Texture and Contour Based Methods for Object Recognition. In *IEEE Conf. on Intelligent Transportation Systems 1997*, Stuttgart, 1997.
13. T. Kalinke and W. v. Seelen. Entropie als Maß des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeitssteuerung. In B. Jähne et al., editor, *Mustererkennung 1996*. Springer, Heidelberg, 1996.
14. T. Kohonen. Self-organization and associative memory. In *Springer Series in Information Sciences 8*. Springer-Verlag Heidelberg, 1984.
15. P. J. Locher and C. F. Nodine. Symmetry Catches the Eye. In A. Levy-Schoen and J. K. O'Reagan, editors, *Eye Movements: From Physiology to Cognition*, pages 353–361. Elsevier Science Publishers B. V. (North Holland), 1987.
16. J. Moody and C. Darken. Learning with localized receptive fields. In *Proc. of the 1988 Connectionist Models Summer School*, pages 133–143. Morgan Kaufman Publishers, San Mateo, CA, 1988.
17. D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-Free Attentional Operators: The Generalized Symmetry Transform. *Int'l J. of Computer Vision*, 14:119–130, 1995.
18. H. J. Ritter, T. M. Martinez, and K. J. Schulten. *Neuronale Netze*. Addison-Wesley, München, 1992.
19. T. D. Sanger. Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network. *Neural Networks*, 2:459–473, 1989.
20. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of Interest Point Detectors. *Int'l J. of Computer Vision*, 37(2):151–172, 2000.
21. C. Theis, I. Iossifidis, and A. Steinhage. Image processing methods for interactive robot control. In *Proc. IEEE Roman International Workshop on Robot-Human Interactive Communication*, Bordeaux and Paris, France, 2001.
22. M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.
23. D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch. Attentional Selection for Object Recognition – a Gentle Way. In *Proc. 2nd Workshop on Biologically Motivated Computer Vision (BMCV'02)*, Tübingen, Germany, 2002.