Multimodal Interaction in an Augmented Reality Scenario

Gunther Heidemann
Neuroinformatics Group
Bielefeld University
P.O. Box 10 01 31
D-33605 Bielefeld, Germany
gheidema@techfak.uni-bielefeld.de

Ingo Bax
Neuroinformatics Group
Bielefeld University
P.O. Box 10 01 31
D-33605 Bielefeld, Germany
ibax@techfak.uni-bielefeld.de

Holger Bekel
Neuroinformatics Group
Bielefeld University
P.O. Box 10 01 31
D-33605 Bielefeld, Germany
hbekel@techfak.uni-bielefeld.de

ABSTRACT

We describe an augmented reality system designed for online acquisition of visual knowledge and retrieval of memorized objects. The system relies on a head mounted camera and display, which allow the user to view the environment together with overlaid augmentations by the system. In this setup, communication by hand gestures and speech is mandatory as common input devices like mouse and keyboard are not available. Using gesture and speech, basically three types of tasks must be handled: (i) Communication with the system about the environment, in particular, directing attention towards objects and commanding the memorization of sample views; (ii) control of system operation, e.g. switching between display modes; and (iii) re-adaptation of the interface itself in case communication becomes unreliable due to changes in external factors, such as illumination conditions. We present an architecture to manage these tasks and describe and evaluate several of its key elements, including modules for pointing gesture recognition, menu control based on gesture and speech, and control strategies to cope with situations when vision becomes unreliable and has to be re-adapted by speech.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Algorithms, Human Factors, Reliability, Verification

Keywords

Mobile systems, Memory, Human-Machine-Interaction, Interfaces, Augmented Reality, Image Retrieval, Object Recognition

ICMI'04, October 13–15, 2004, State College, Pennsylvania, USA.

1. INTRODUCTION

For several years now, we are witnessing a dramatic decrease in the costs of sensors and computing or communication devices. As a consequence, the way we interact with computers, machines or even household applications is about to change: computing hardware is becoming ubiquitous or wearable, rooms are becoming smart, space is becoming the interface and interaction is becoming seamless and intelligent (cf. e.g. [7, 19, 24, 26]). Naturally, intelligent interaction calls for multimodality.

Since communication and interactions among humans are inherently multimodal, seamless interaction with machines calls for multimodality as well. Furthermore, as human interaction relies on human senses, multimodal interfaces should regard perceptual modalities which meet human needs, i.e. they should enable visual and acoustic interaction (cf. [18]). This, however, requires techniques for robust image and speech understanding. As discussed in the panel sessions at ICMI'03, robust recognition requires learning and adaption.

In this paper, we will present a prototype of a situated intelligent system with advanced interfaces for information retrieval. It is designed to recognize objects in an office environment, to store this kind of information and to make it available for its users if asked for. By means of interaction using gesture and speech recognition, the system can learn and extend its predefined knowledge about its surroundings. The prototype discussed in this paper is fully mobile. Wearing a head mounted device with cameras and an augmented reality display, the user perceives his environment as well as information generated by the system.

2. MOTIVATION AND RELATED WORK

In traditional intelligent interface research, model acquisition and recognition processes for interaction are decoupled in time and control: First, a set of handcrafted or learned object models is given to the system. Then, the models are used in order to accomplish the recognition tasks necessary for intelligent and seamless interaction. Especially when it comes to communicating about visual percepts, this approach has a couple of shortcomings that prevent the applicability of perceptive interfaces on a broader scale, which is needed in everyday environments like domestic or office settings. First of all, it is impossible to design a complete set of objects and activities. Secondly, human-machine interaction in dynamic environments will require to solve ad-hoc

tasks in contrast to a set of pre-specified tasks which were thought of when designing the system.

While there has been a lot of work concentrating on single aspects to overcome this shortcomings, e.g. generic object recognition [4], contextual object recognition [13], perception action cycle approaches [1], or one-shot object learning [17], there has been little work on systems that integrate different techniques and realize the complex but robust performance required in everyday environments. Early attempts in this area were reported by Hanson and Riseman [10]. Other approaches that do not include the possibility of learning new concepts include [6, 8]. A quite promising approach to learning grounded representations of the world by means of interaction has been reported by Roy [21]. Word semantics are learned from parallel speech and image data. The recently proposed *Cognitive Vision System* paradigm [5] even goes one step beyond for it considers integrated systems that are embedded in the world, interact with their environment to gather knowledge and articulate their knowledge by changing the state of the environment. Our system follows this paradigm: Working in an everyday office environment, the user wears a head-mounted device equipped with cameras and a display [22]. Information about recognized objects and results of user queries are visualized using augmented reality. Also, by means of displaying status messages and prompts the system can communicate with its user. This closes the perception-action cycle; asking for manipulations of the environment in order to study their effects can accomplish interactive object and event learning.

In the following, we will describe the system "top down", starting with a brief description of the intended final functionality (section 3.1) and the required user interface (section 3.2), over a technical description of its components (section 4) to the evaluation of isolated functionalities (section 5).

3. SYSTEM DESCRIPTION

The prototype described here is being developed within the VAMPIRE project[1], which is aimed at investigating an active visual memory for interactive retrieval in an augmented reality scenario.

3.1 Augmented Reality Scenario

In the prototype scenario, a user wears a helmet with two cameras as artificial eyes, which view the area in front (Fig. 1). The user sees the camera images via a head mounted display. As an overlay to the scene, both the system output ("augmentations of reality") and tools such as buttons and menus for interaction are displayed. An inertial sensor allows to keep track of movements of the user [22].

The functionality of the system is that of a "personal assistant", who has knowledge about objects in the environment. Object knowledge is mainly acquired online by instructions of the user, who references objects by pointing gestures. Once the system is adapted to the environment, it can answer queries for information about objects by display of augmentations, or retrieve "lost" objects.

The project is not aimed at the development of an actually "wearable" system. Rather, the focus is on the development of an active memory and methods for multimodal interaction, therefore, no effort was spent on miniaturized

[1] `http://www.vampire-project.de`

Figure 1: User with head mounted cameras and display. The input is looped back into the display and enhanced by visual augmentations.

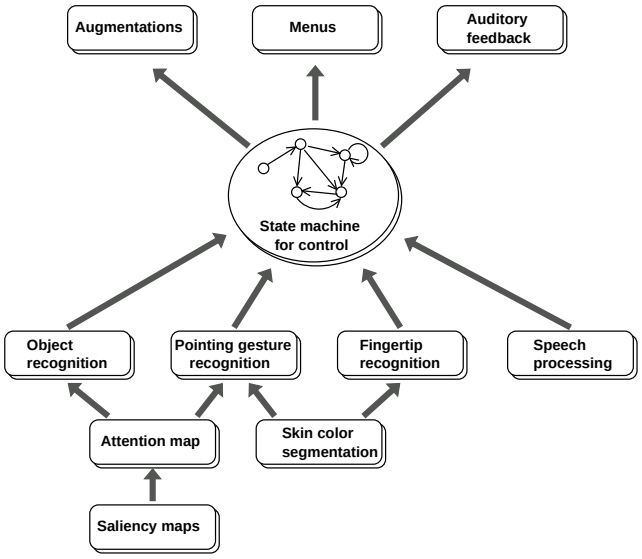

Figure 2: System architecture: Input modules running in parallel are evaluated by a central control module, realized as a state machine. State transitions reflect varying modes or tasks. The state machine is the "client" of the visual and auditory output servers, to which it directs its output.

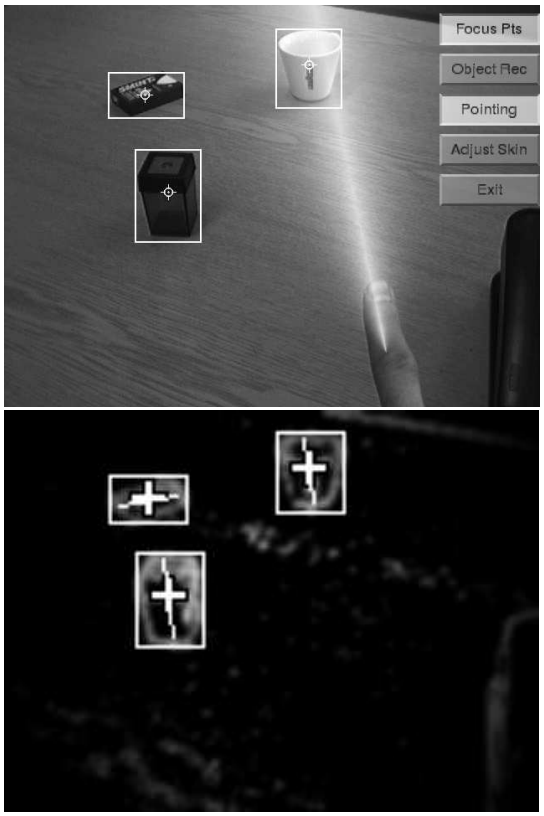

Figure 3: Top: The user has chosen to see augmentations of focus points and corresponding bounding boxes of objects, that were found by the attentional subsystem. The pointing recognition is switched on in the menu and the correct recognition of the finger and the pointing direction is visualized as a highlighting beam. Bottom: Example of a saliency map for the above scene.

hardware. Instead, the user wears a backpack with a laptop which performs frame grabbing, early image processing, visualization, audio in/output and communication tasks. It is connected to additional processing units via wireless LAN.

3.2 Multimodal Interface

A major challenge of the outlined system is that human-machine interaction has to take place without any traditional devices (mouse, keyboard), but relies entirely on hand gestures and speech input. Another complication is that there are basically three types of interaction:

1. Communication about the environment, i.e. reference to objects, asking for particular information, showing objects to the system,
2. Interaction with the system itself, e.g. switching into different modes or "escape" to a start off mode,
3. Re-adaptation of the input devices in case of erroneous recognition of user input, e.g., re-adaptation to skin color under different illumination conditions.

We will describe the interface first from the point of view of the user, the next section outlines the technical implementation.

Figure 4: Menu control by fingertip: The user carries out a "pressing" gesture on a button. The system analyzes the position and the movement trajectory to recognize "Selected" and "Pressed" events.

3.2.1 Menu based control

For communication with the system, on the right hand side of the image a semi transparent menu is overlaid. Fig. 4 shows the "Main" menu, which is displayed after initialization or after an "Escape" signal. There are three types of menu buttons: *Triggers*, which cause a certain action to be carried out, *checkbuttons*, which can be turned "on" or "off", and sets of exclusively coupled *Radiobuttons*. The latter work like checkbuttons, but pressing one button turns off another.

To ensure a reliable menu communication, all types of buttons must be first *selected*, then *pressed*. Selection can be carried out by speech, naming the label of the button, or by gesture. In this case, the user indicates a button with his finger. After selection by gesture, the button can be pressed either using speech ("Yes"), or by moving the finger inwards. If a button was selected by speech, it must also be "pressed" using speech. Menu operation is accompanied by visual and auditory feedback for selection and pressing. Of the various functionalities controlled by menus, the most essential ones will be discussed in the following.

3.2.2 Object reference

For interaction with the memory, the user can reference objects by pointing gestures. The system is sensitive to pointing only when the hand is held in the lower part of the visible area. A major problem of pointing gestures is the accurate detection of pointing direction. The difficulty is caused mainly by two factors: Firstly, a hand has many degrees of freedom, i.e. the same gesture may have many different appearances. Secondly, humans are not used to precise pointing, because vagueness of pointing is resolved by context knowledge — during discourse, pointing gestures mostly select between already known alternatives.

Here, the problem is solved by two complementing strategies: Pre-selection of salient areas by attentional mechanisms, and system feedback. Since an understanding of

natural scenes is still far beyond the capabilities of a machine, real contextual knowledge is not available. But instead, purely data driven methods can provide information about image regions which are "salient" or "interesting" in the sense of general, context free measures. As described in section 4.5, such measures are used to select only the most salient points of the input image, and offer only these locations to the user as possible pointing targets. In other words, the task of selecting an arbitrary point from a large 3D-volume is reduced to the task of selecting a 2D-position out of a limited set.

Salient locations are presented to the user as a set of overlaid markers, indicating possible pointing targets (Fig. 3). When a pointing gesture is carried out, the detected pointing direction is visualized as a system feedback (Fig. 3). So in case of errors, the user can e.g. make the pointing finger better visible to the system, or even adapt skin color detection if necessary.

3.2.3 Object learning / re-learning

The system has a built-in neural object recognition system which can be trained online. The complexity of the scene is so far restricted to a flat table in front of the user. For object learning, there are menu buttons to make either a novel object known to the system or to supply additional sample views of a known but so far incorrectly classified one (Fig. 5). In both cases, the user will be asked to place the object first in a certain reference position. When the first image has been captured, auditory feedback is given and the user is asked to turn the object to another pose, and so on. More views improve classification, but the decision how many views are captured is left to the user. In our experiments a reasonable number proved to be about 20 views. After view sampling, within about one second a new object classifier is available, which is, however, provisional and not yet fully trained to exploit the new object views. So, a running version is available at once, though still at reduced performance. In a parallel thread, a new classifier is trained thoroughly in the background within several minutes. When ready, it replaces the provisional classifier without requiring any action of the user.

3.2.4 Skin color adaptation

Skin color detection is crucial to both manual menu control and pointing recognition. Since a universal model of skin color does still not exist, detection may be influenced by variations of the lighting conditions. Consequently, as described in section 4.2, a flexible skin color model is used that can be easily adapted. In case of unsatisfactory results, the user can command the system — preferably by speech — to overlay the segmentation results (Fig. 6). Adaptation can be started by another command which makes a small frame appear. Samples of skin color can now be acquired, triggered again by speech input. Adapting the system to the new samples takes less than 0.5 seconds.

3.2.5 Activating fallback mode

Under very bad external conditions, both gesture and speech control may fail. In this case, the user can activate a fallback mode by covering the cameras with a hand for about a second. In this case, the system aborts all tasks, deactivates speech and returns to the main menu. Skin color detection, which is crucial for the gesture based menu control, is replaced by a simple thresholding mechanism: The user is expected to look at a white surface (e.g. a piece of paper), such that a finger will appear dark by comparison. As dark objects are now considered as fingers, the menu can now be controlled again safely and skin color adaptation be performed.

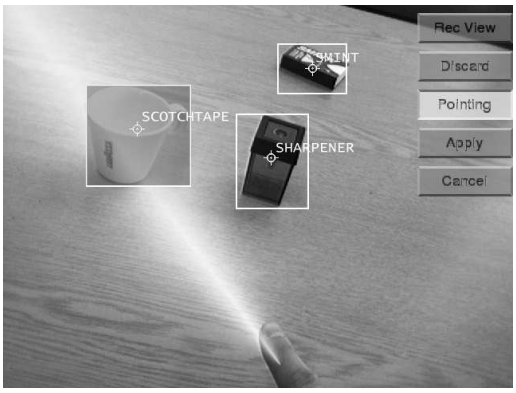

Figure 5: Online Learning Mode: The user points at an object, in this case the cup, which has been falsely classified to be "SCOTCHTAPE". The system highlights the object that is pointed at. By selecting "Rec View" from the menu, a shot of the object is added to the view database. When enough shots of the object have been taken, the user can initiate the classifier training by choosing "Apply" from the menu. Currently, labels have still to be typed by a keyboard, this will be replaced by speech recognition in future versions.

4. TECHNICAL DESCRIPTION OF MAIN MODULES

As a complete system description is beyond the scope of this paper, we sketch its main components.

4.1 System structure and control

The system consists of several independently running modules which are organized in a flat architecture. There are basically three types of components: *Input* modules for processing of vision and speech, *output* modules for display of augmentations and menus as well as auditory feedback, and a *control* module (Fig. 2).

The input modules are independent of each other and provide a continuous stream of processing results. The control module is realized as a finite state machine. Its state depends on the current task, e.g. acquisition of samples, detection of pointing gestures, or menu interaction. Depending on the state, the control module selectively evaluates the currently relevant data from the input modules, switches between states, and sends data to the output modules. The latter displays augmentations, give auditory feedback, and show appropriate menus, highlighting, etc. The output modules are in the role of servers, which act on requests of the client, i.e. the control module. For efficiency, the control module deactivates currently not needed input modules.

Since the focus of the paper is on input modalities, a detailed sketch of the control flow and output modules can

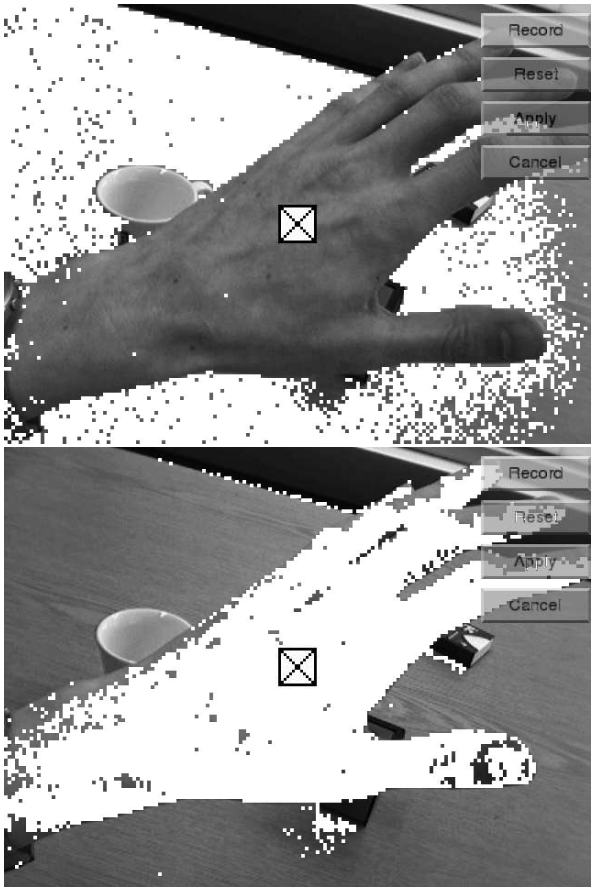

Figure 6: Skin color adaptation: Above: First, a visualization of the of the skin color segmentation is shown by marking all pixels detected as skin color white. Obviously , the segmentation performs badly since most white pixels are located on the wooden table and almost none on the hand. The user now moves the hand underneath the target to record skin color samples by choosing "Record" from the menu. Below: After recording several samples, parameters of the skin color segmentation have been adjusted automatically. Segmentation is satisfactory now.

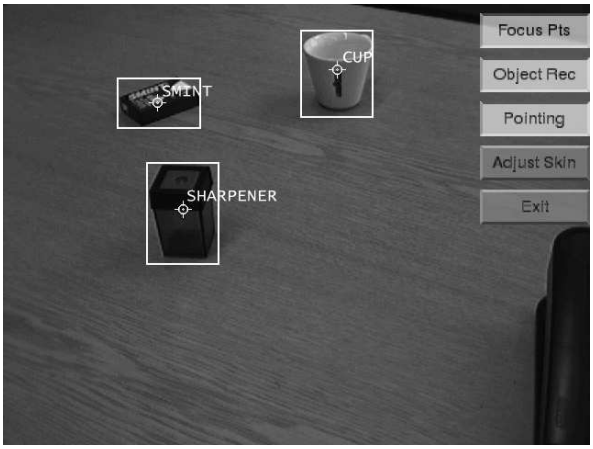

Figure 7: Output of the object recognition module.

not be given. In the following, the input modalities will be described and evaluated.

4.2 Skin color detection

Skin color segmentation serves the selection of candidate regions for fingers and hands. Though the skin color module plays the role of an attentional subsystem, it is implemented as a processing branch independent from the attention module (section 4.5), because, firstly, hands or fingers are not of interest as "learnable" objects, and secondly, the correct detection of hands is crucial for interaction, so attention should never be distracted by other salient regions.

Caused by the expeditious changes of the illumination conditions, online adaption of the skin color segmentation module became mandatory. As an online trainable skin color classifier we apply the model of Stoerring [2], who verified that the overall skin color distribution is a shell-shaped area in the r-g color space that is called the *skin locus*. The color space is determined by $r = R/(R+G+B)$ and $g = G/(R+G+B)$.

The skin locus can be well fitted by two parabolae in the two dimensional color space [23]. When started, the system first applies pre-defined standard parameters for the parabolae. If skin segmentation is unsatisfactory, the user can activate an adaptation mode. In this case, the color distribution is derived from the user's hand, held in a highlighted frame of the displayed image. The center of mass and the standard deviation of the color distribution are estimated, then the parameters of the two quadratic functions are adapted to enclose the new color distribution. This is done by setting the r-coordinates of the angular points to the center of mass and the g-coordinates to two times the distance of the standard deviation.

The user is asked to hold his or her hand into the highlighted frame several times to gather samples. When the newly fitted skin locus is available, the new segmentation results are displayed. The user can now decide whether to apply the new segmentation, gather still more samples, or use the old segmentation.

4.3 Trainable object recognition

The module for object recognition facilitates a fast and easy to adapt classification of image patches which are candidate regions for objects. It is based on a neural architecture called "VPL" described in detail in earlier work, see [12] and references therein. In short, the VPL-classifier consists of three processing stages: The first two layers perform feature extraction by means of local principal component analysis (PCA), which is implemented in the separate steps of clustering the raw input data by vector quantization ("V-layer") and a subsequent local PCA ("P-layer"). This relatively simple implementation of local PCA facilitates a fast training and avoids the need to search for suitable training parameters. The third layer consists of several neural classifiers of the local linear map type ("L-layer"), which map the extracted features to any desired output vector. For object recognition, the output vector has one component for each of the objects, which indicates the probability that a certain object is presented. The final classification is the object with maximum probability.

The VPL-classifier is particularly well suited for the present tasks, because it can be trained in two ways. When novel training views were acquired online and should be incorpo-

rated fast, the first two stages, which perform feature extraction, are left unchanged — it is assumed that the so far available features cover appearance of the new object at least to some degree. Only the neural classifier is trained anew, in case of a novel object, its output dimensionality is increased. In a parallel thread, a full training of all three processing stages – which is much slower – can be performed offline.

4.4 Pointing gesture recognition

For pointing gesture recognition, a second instance of the VPL operates on the candidate regions supplied from skin color segmentation. Two tasks have to be solved: Making a decision whether the skin colored region is a pointing hand at all, and, if so, detection of the pointing direction. So the VPL has two qualitatively different output channels: a binary one indicates presence of a pointing hand, and a continuous channel is the pointing angle (which is irrelevant in case the first is "false"). The classifier is trained offline from labelled sample images, an online training for pointing gestures is planned but not yet implemented.

To obtain object reference, the symbolic output (angle) of this module is transformed back to a subsymbolic representation as described below.

4.5 Attention module and object reference

On the signal level, relevant or "salient" image regions may be indicated by a variety of features. Our approach therefore exploits several complementary methods, the most important of which are gray value entropy, local symmetry and edge-corner detection. A *local entropy* map yields high saliency value for image windows which have a high informational content in the sense of information theory [14]. A *symmetry map* attracts attention to objects or details which are locally symmetric [20]. The use of symmetry is cognitively motivated by eyetracking experiments [16]. The third feature map concentrates on *edges and corners* as small salient details of objects. Here we use the standard detector proposed in [11].

The output of each of these algorithms is a "saliency map", which assigns a saliency value to each pixel (Fig. 3). The different maps are integrated by weighted summation to obtain one final saliency map M. The highest maxima of M are displayed in the original image by markers, which indicate possible pointing targets (Fig 3). The selection of a particular target is implemented as an overlay of M with a "manipulator map", i.e. a cone-shaped activation in the direction of pointing, which highlights the nearest maximum of M. A detailed description of the underlying saliency algorithms, the integration to a single map M, and the selection of pointing targets can be found in [12].

4.6 Menu control

Menu buttons are activated either by speech input, or with the fingertip (Fig. 4). As this functionality is the fundamental user control mechanism, it has, apart from skin color segmentation, its own specialized processing branch. When a menu is being displayed, candidate skin color regions which are within the region of menu buttons are evaluated by template matching with prototypic fingertip shapes. The matching module operates on multiple scales to facilitate recognition of the fingertip at various distances. To avoid erroneous input, the fingertip must be found on the same button and at the same scale for a minimum number of frames before a button is selected. To trigger a "press" event, a movement towards the center must be detected.

Simultaneous detection of a fingertip and a pointing hand is possible, since the user might use both hands.

4.7 Speech processing

Speech input is processed by the ESMERALDA system, which supplies an integrated environment for statistical model estimation [9], which is applied here for speech recognition. ESMERALDA employs an incremental recognizer, which uses in turn vector quantization for feature extraction and, subsequently, Hidden Markov Models to estimate the mixture densities and n-gram language models. ESMERALDA is able to recognize languages based on a context-free grammar. It was evaluated using the VERBMOBIL appointment scheduling domain. The achieved word error rate of 20.1% was among the best results for this benchmark test [9]. Due to its classification robustness, it was successfully applied in object recognition systems [25] and in robotics [3].

5. EVALUATION

The evaluation of vision systems that operate in natural environments is challenging and by itself a major topic of research ([15]). The main problem is the difficulty in creating an evaluation setup which is capable of exhaustively covering all degrees of freedom the system input is exposed to. To cope with this problem, evaluation must be restricted (a) to testing the most relevant components and (b) to a defined set of "use cases". This limited, but possibly prototypic applications should allow to judge the output of the integrated system. In the following, we define a small set of such use cases for fingertip and pointing reference recognition, which are the most important subsystems for interaction.

5.1 Menu control

The experimental setup for fingertip recognition is the following: A "generic" menu is overlaid in the head mounted display, buttons are labeled by numbers. A subject is then asked by the supervisor to first select and then press a particular button number. The number of the button is chosen randomly. We count how often the subject manages to press the correct button using his or her finger within 5 seconds time, i.e. how often the system recognizes a "pressed-event" on the correct button first within the time limit. The experiments were carried out in an office environment, so background is varying and cluttered.

The evaluation of the menu control consists of two parts. First, the accuracy of the menu control is tested under varying parameters:

- To test the spatial resolution of the fingertip recognition, the number of buttons, is varied from 3 to 11. The buttons are equidistantly arranged in the display.
- Four different illumination conditions were applied in order to test robustness of the skin color segmentation: *Natural daylight, artificial light, source of light behind the back of the person, and frontal lighting.*

Results of the experiment are shown in Table 1. The test persons could first get used to the system behavior over 20 items without performance evaluation. Success rates were than counted over 160 items, illumination was changed every

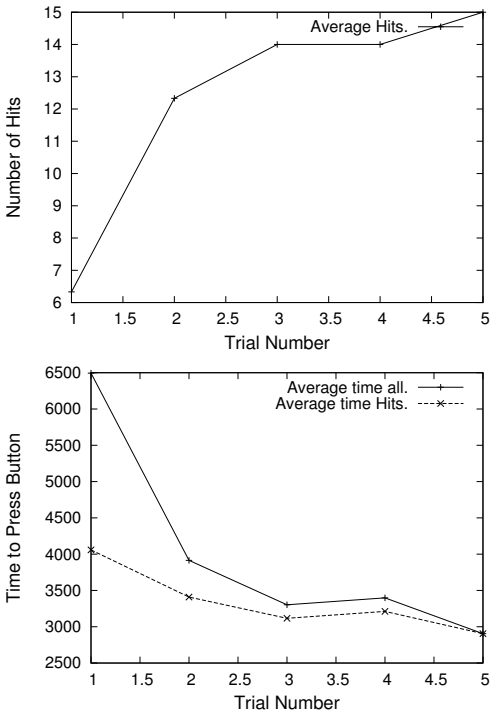

Figure 8: Above: Average hit rate of the second menu control evaluation. Below: Average time the subjects need to select and press a button of the second menu control evaluation experiment. The upper graph depicts the time including errors, whereas the lower graph presents just the averaged time in case of a correct selection.

Table 1: Hit rates for the menu control evaluation.

Number of buttons	3	5	7	9	11
Matches	98.6	97.3	90.6	74.6	57.3

40 items. Results were averaged over five subjects and the different illumination conditions. Table 1 shows that menu control is efficient and robust for up to seven buttons, for nine buttons, control is still possible but requires more than five seconds. In this case, sub-menus are more advisable.

Due to changing illumination conditions and background, the accuracy of the menu control does not only depend on the correctness of the fingertip recognizer but on the adaption capabilities of the user. Therefore, to test how fast a subject gets used to the system behavior, in a second experiment the above described setup is used with fixed illumination conditions and with a fixed number of buttons (here 7). Five subjects, completely unexperienced to the system, are asked to select and then press buttons, as mentioned above. One trial consists out of 15 items each. Fig. 8(above) shows the fast increase of the number of hits whereas Fig. 8(below) depicts the fast decrease of the average time the subject needs to select and press the button. Already after the first trial all subject reached a hit rate of about 80 percent. After 5 trials nearly all subjects achieved a hundred percent hit rate.

Table 2: Results of the pointing gesture evaluation in percent.

Distance	1.5 cm	3 cm	5 cm	10 cm	15 cm	20 cm
Matches	26.6	75	81.6	93.3	100	100

5.2 Pointing Gesture Recognition

The primary aim of the pointing gesture evaluation setup we chose is testing the spatial resolution and efficiency of this subsystem using a generic pointing task: A subject is asked to point at a row of six white circles placed on a black paperboard on the desk. The diameter of each circle is 12 mm, distances between the circles vary from 3 to 20 cm (measured from the center 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° to 28° could be tested for targets of about 0.9° or 1.7° angular range, respectively. A supervisor gives the command to point at one of the circles by telling a randomly chosen circle number. Only the inner four circles were used to avoid border effects. A match is counted if the system recognizes a reference to the correct circle within three seconds time. As the subject is able to see the visual feedback during the test, i.e. which spot is currently referenced, in the head mounted display, he or she is able to adapt his or her movements to the system behavior.

Results of the experiment are shown in Tab. 2. The values are averaged for five subjects with 15 items each. The match percentage is quite high even for small target distances. Only at point distances of 1.5 cm, the values decrease substantially. This effect is probably caused by the accuracy of the detected skin color patch of the finger. Due to slightly changing illumination conditions, the center of the skin color blob and the direction of the skin color patch leads to jumping points of attention. The major result achieved in this test scenario is that the user is enabled to adjust single pointing gestures to a target and that the user can adapt himself or herself to the system behavior. This way the achievable *effective resolution* is improved by repetitions of the test runs, because it does not solely rely on the accuracy of the pointing gesture recognition any more.

6. DISCUSSION AND OUTLOOK

We have presented a human-machine interface using hand gesture and speech in the context of an augmented reality scenario. So far, generic functionalities were implemented: For interaction with the system, menu based control, object reference, and object learning were described. An additional functionality is the adaptation of the interface itself by skin color training. The latter demonstrates the need of multimodality particularly well: In case vision does not work well, speech can not only take over control but also trigger re-adaptation of vision.

In future work, the system will be equipped with an active memory which can give feedback on the "standard of knowledge" of the system. So the system will be enabled to ask the user for additional object views, to compare similar objects and carry out retrieval tasks. Interaction will be enhanced by an adaptation mode for speech, where speech parameters like amplification can be controlled, and indi-

vidual speech samples of the user be recorded. To improve gesture recognition, online training of user-specific pointing poses will be implemented similar to the online acquisition of object samples.

7. ADDITIONAL AUTHORS

Additional authors: Christian Bauckhage (Applied Computer Science Group, email:*cbauckha@techfak.uni-bielefeld.de*), Sven Wachsmuth (Applied Computer Science Group, email: *swachsmu@techfak.uni-bielefeld.de*), Gernot Fink (Applied Computer Science Group, email:*gernot@techfak.uni-bielefeld.de*), Axel Pinz (Institute of Electrical Measurement and Measurement Signal Processing, Graz University of Technology, email:*axel.pinz@tugraz.at*), Helge Ritter (Neuroinformatics Group, email:*helge@techfak.uni-bielefeld.de*) and Gerhard Sagerer(Applied Computer Science Group, email: *sagerer@techfak.uni-bielefeld.de*) .

8. REFERENCES

[1] Y. Aloimonos. Active vision revisited. In Y. Aloimonos, editor, *Active Perception*, pages 1–18. Lawrence Efibaum, Hillsdale, 1993.

[2] H. J. Andersen, M. Stoerring, and E. Granum. Physics-based modelling of human skin colour under mixed illuminants. *Robotics and Autonomous Systems*, 35(3-4):131–142, 2001.

[3] 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. Comput. Intelligence in Robotics and Automation*, pages 328–333, Banff, Canada, 2001.

[4] R. Bergevin and M. D. Levine. Generic Object Recognition: Building and Matching Coarse Descriptions from Line Drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(1):19–36, 1993.

[5] H. I. Christensen. Cognitive (vision) systems. *ERCIM News*, pages 17–18, April April, 2003.

[6] J. L. Crowley and H. I. Christensen, editors. *Vision as Process*. Springer, 1995.

[7] J. L. Crowley, J. Ciutaz, and F. Bérard. Things that see. *Communications of the ACM*, 43(3):54–64, 2000.

[8] B. A. Draper, G. Kutlu, E. M. Riseman, and A. R. Hanson. ISR3: Communication and Data Storage for an Unmanned Ground Vehicle. In *Proc. ICPR*, volume I, pages 833–836, 1994.

[9] G. A. Fink. Developing HMM-based recognizers with ESMERALDA. In V. Matoušek, P. Mautnerand J. Ocelíková, and P. Sojka, editors, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 229–234, Berlin Heidelberg, 1999. Springer.

[10] A. R. Hanson and E. M. Riseman. VISIONS: A Computer System for Interpreting Scenes. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*. Academic Press, 1978.

[11] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conf.*, pages 147–151, 1988.

[12] G. Heidemann, R. Rae, H. Bekel, I. Bax, and H. Ritter. Integrating context-free and context-dependent attentional mechanisms for gestural object reference. In *Proc. Int'l Conf. Cognitive Vision Systems*, pages 22–33, Graz, Austria, 2003.

[13] A. Hoogs, J. Rittscher, G. Stein, and J. Schmiederer. Video Content Annotation Using Visual analysis and a Large Semantic Knowledgebase. In *Proc. CVPR 2003*, volume 2, pages 327–334, 2003.

[14] T. Kalinke and W. von Seelen. Entropie als Maß des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeitssteuerung. In B. Jähne, P. Geißler, H. Haußecker, and F. Hering, editors, *Mustererkennung 1996*, pages 627–634. Springer Verlag Heidelberg, 1996.

[15] G. Lindegaard. *Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems*. Chapman & Hall, 1994.

[16] 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.

[17] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. ICCV*, pages 1150–1157, 1999.

[18] S. Oviatt and P. Cohen. Multimodal Interfaces That Process What Comes Naturally. *Communications of the ACM*, 43(3):45–53, 2000.

[19] A. Pentland. Perceptual intelligence. *Communications of the ACM*, 43(3):35–44, 2000.

[20] 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.

[21] D. Roy. Learning visually grounded words and syntax of natural spoken language. *Evolution of Communication*, 4(1):33–56, 2000.

[22] H. Siegl, and A. Pinz. A Mobile AR Kit as a Human Computer Interface for Cognitive Vision. In *Proc. WIAMIS'04*, Lisbon, 2004.

[23] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. Skin detection in video under changing illumination conditions. In *Proc. CVPR 2000*, volume 1, pages 839–842, 2000.

[24] T. E. Starner. Wearable computers: No longer science fiction. *IEEE Pervasive Computing*, 1(1):86–88, 2002.

[25] S. Wachsmuth, G. A. Fink, F. Kummert, and G. Sagerer. Using speech in visual object recognition. In G. Sommer, N. Krüger, and C. Perwass, editors, *Mustererkennung 2000, 22. DAGM-Symposium Kiel*, Informatik Aktuell, pages 428–435. Springer, 2000.

[26] C. Wisneski, H. Ishii, A. Dahley, M. Gobert, S. Brave, B. Ullmer, and P. Yarin. Ambient displays: Turning architectural space into an interface between people and digital information. In *Proc. Int'l Workshop on Cooperative Buildings*, pages 22–32, 1998.