EURASIP Journal on Applied Signal Processing 2005:14, 2375–2390

# Vision Systems with the Human in the Loop

**Christian Bauckhage**
*Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany*
*Email: cbauckha@techfak.uni-bielefeld.de*

**Marc Hanheide**
*Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany*
*Email: mhanheid@techfak.uni-bielefeld.de*

**Sebastian Wrede**
*Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany*
*Email: swrede@techfak.uni-bielefeld.de*

**Thomas Käster**
*Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany*
*Email: tkaester@techfak.uni-bielefeld.de*

**Michael Pfeiffer**
*Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany*
*Email: pfeiffer@techfak.uni-bielefeld.de*

**Gerhard Sagerer**
*Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany*
*Email: sagerer@techfak.uni-bielefeld.de*

*Received 31 December 2003; Revised 8 November 2004*

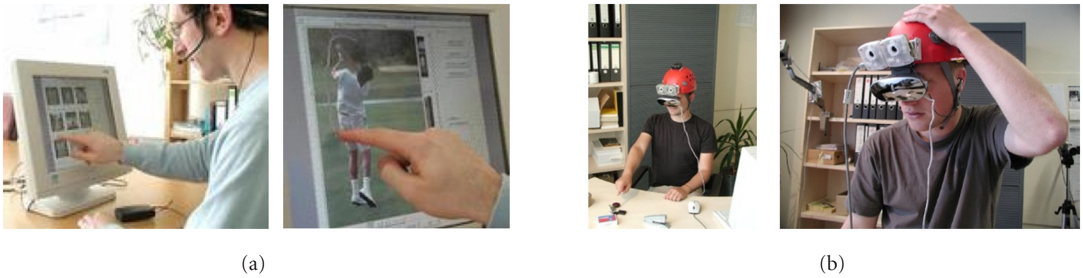
The emerging cognitive vision paradigm deals with vision systems that apply machine learning and automatic reasoning in order to learn from what they perceive. Cognitive vision systems can rate the relevance and consistency of newly acquired knowledge, they can adapt to their environment and thus will exhibit high robustness. This contribution presents vision systems that aim at flexibility and robustness. One is tailored for content-based image retrieval, the others are cognitive vision systems that constitute prototypes of *visual active memories* which evaluate, gather, and integrate contextual knowledge for visual analysis. All three systems are designed to interact with human users. After we will have discussed adaptive content-based image retrieval and object and action recognition in an office environment, the issue of assessing cognitive systems will be raised. Experiences from psychologically evaluated human-machine interactions will be reported and the promising potential of psychologically-based usability experiments will be stressed.

**Keywords and phrases:** cognitive vision, adaption, learning, contextual reasoning, architecture, evaluation.

## 1. INTRODUCTION

Currently, the computer vision community is witnessing the emergence of a new paradigm. Even though its roots at least date back to work by Crowley and Christensen [1] from the early 1990s, the idea of bringing together the achievements of 30 years of research in artificial intelligence, automatic perception, machine learning, and robotics was termed *cognitive computer vision* just recently (cf. [2]).

Rather than trying to tackle the philosophical, psychological, or biological subtleties of the question what characterises cognition, we will adopt Christensen's point of view and restrict ourselves to a limited notion of cognition. Following his argument, we will consider cognition as the generation of knowledge based on prior models, learning, reasoning, and perception [3]. In this sense, cognition is an



(a)

(b)

Figure 1: (a) Interactive content-based image retrieval using speech and haptics. (b) Head-mounted cameras and display for augmented reality visualisation of recognised objects and events in an office environment.

active process. Instead of just monitoring its surroundings, a cognitive vision system is able to communicate or interact with its environment. This underlines that the acquisition, storage, retrieval, and use of knowledge is no end in itself but guides the system's perception and (re)action. Simultaneously, the capabilities to perceive and act guide cognitive processes. Without perception and the possibility to manipulate or communicate perceived entities or events, knowledge cannot be acquired. Memory, however, is a limited resource. Besides mechanisms for learning, cognitive vision thus also implies attention control and a sense for relevance which comes along with the capability to forget irrelevant information. This requires flexible knowledge representation and techniques for top-down and bottom-up processing as well as functionalities for contextual reasoning and categorisation. Together with the biologically motivated principle of multiple computations [4], categorisation yields adaptability, flexibility, and robustness.

Christensen even argues that embodiment is a prerequisite for cognitive vision systems. Only the capability to interfere with the environment can close the so-called *perception action cycle*. However, even though there is considerable progress in the fields of mechatronics and robotics, machines that independently explore their environments are still in their infancy. In this contribution, we will thus argue that human-machine interaction can compensate embodiment. We will report results and experiences from two joint research projects on complex vision systems that make extensive use of the idea of the *human in the loop*.

First, we will present a system for interactive content-based image retrieval (CBIR). Although state-of-the-art retrieval systems adapt to the preferences of their users, the involved learning processes only occur on the feature level of vision and there is no real knowledge acquisition. Claiming CBIR as a subfield of cognitive vision would therefore mean to overstretch the idea. However, CBIR systems are a perfect example of the benefits of bringing together pure computer vision and human-machine interaction. The retrieval system introduced in Section 2 combines machine learning and adaption with intuitive multimodal interfaces for image retrieval. While working with the system, the user may use natural language or a touch screen facility to indicate interesting image content (see Figure 1a).

Then, we will introduce systems which follow the cognitive vision paradigm. They are being developed in a research project dedicated to architectures and computational models for *visual active memories* (VAMs). Visual active memories are systems which evaluate given facts or gather and integrate contextual knowledge for visual analysis. VAMs can learn new concepts and categories as well as new spatiotemporal relations. They can adapt to unknown situations and may be scaled to different domains. Furthermore, the project investigates techniques and interfaces for advanced *interactive retrieval*. As an example, Figure 1b shows impression from experiments with a prototype of a mobile VAM. Working in a natural office environment, the user wears a head-mounted device which is equipped with cameras and a display. Information about recognised objects and results of user queries are visualised using augmented reality (AR). Likewise, by displaying status messages and prompts into the user's field of view, the system can communicate with its user and thus close the perception-action cycle. Asking for manipulations of the environment in order to study their effects can accomplish interactive object and event learning.

The long-term perspective for interactive VAM research is to proceed towards *memory prosthetic devices*. The system in Figure 1b, for instance, can be seen as a first prototype of *memory spectacles* that may assist the memory challenged. But, of course, expecting assistive technology to answer questions like "Where did I put my keys"? requires vision systems that will operate in everyday environments. The VAM demonstrators presented in Section 3 are situated in unconstrained office environments. Applying multiple computations and contextual reasoning, the systems are able to identify different objects, actions, and activities. They can be operated using speech and gesture; they cope with varying illumination as well as cluttered video signals and have capabilities in interactive object learning.

With the advent of complex, interactive, and adaptive vision systems, the problem of system evaluation arises. Obviously, the evaluation of an interactive system must not be restricted to a snapshotted performance testing. Rather, it has to take into account that failures that appear at a certain stage of an interactive session might be corrected later on. Also, learning and adaption might improve the system performance over time. However, up to now, no commonly

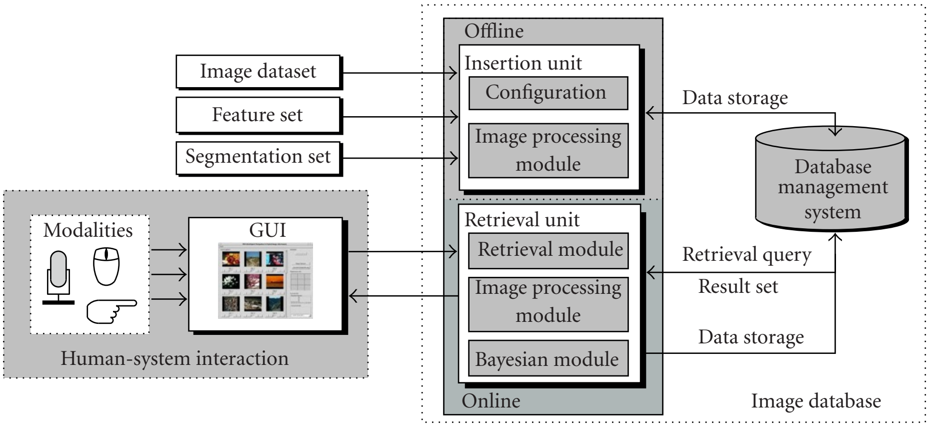


Figure 2: Components and conceptual architecture of the INDI system.

accepted evaluation framework that deals with these aspects has been established. In Section 4, we will point out that usability experiments provide a promising avenue to solve this problem. We will report on a study designed by psychologists we performed with naive users of our CBIR system. As we will see, this methodology can lead to surprising insight on how the human in the loop experiences his interaction with a cognitive system. Finally, a conclusion will close this contribution.

## 2. THE INDI SYSTEM

This section will present a system for content-based image retrieval (CBIR) that results from a project on Intelligent Navigation in Digital Image databases. Its characteristics are adaptability and multimodal interaction. Adaption to the peculiarities of a certain retrieval task is guided by user feedback and happens on the feature level of computer vision. Multimodal input devices are provided in order to facilitate intuitive handling. Figure 2 sketches the conceptual architecture of the INDI system. In the following, we will concentrate on the retrieval module displayed in the middle of the figure as well as on the user interface seen on the left.

### *2.1. A hierarchical CBIR approach*

Image retrieval usually starts with low-level feature extraction either from an entire image or from certain image regions. The INDI system considers the following features: local moments in the LUV colour space as introduced by Stricker and Dimai [5], fuzzy histograms of the hue channel of the HSV colour space, and edge co-occurrence histograms which according to Brandt and Oja [6] are local shape descriptors.

Since local image signatures increase the precision in CBIR, our system automatically extracts regions of interest. In an initial keypoint detection process, the most salient points in a colour image are identified using the generalised Harris keypoint detector [7]. Afterwards, they are clustered using support vector clustering [8]. Pixels within the resulting clusters represent regions of interest which allow the computation of meaningful signatures and can be referenced during a retrieval process.

Following the approach of Rui and Huang [9], we assume an image object $O_k$, that is, an image or parts of an image, to be characterised by several attributes: (i) a set of pixels; (ii) a set of feature classes, such as colour or texture; (iii) for each feature class $f_i$, there is a set of specific features. Examples of specific colour features could be histograms in different colour spaces or some sort of brightness information. All instances $j$ of specific features are stored as sets of feature vectors $R = \{\vec{r}_{ij} \in R_{ij}\}$.

Our system follows the common *query-by-example* approach and compute similarities between the database image objects $O_k$ and a query object $Q$. Using generalised Euclidian distances

$$m_{ij}(\vec{r}_{ij},\vec{q}_{ij}) = (\vec{r}_{ij} - W_{ij}\vec{q}_{ij})^T \mathbf{\Omega}_{ij}(\vec{r}_{ij} - W_{ij}\vec{q}_{ij}), \quad (1)$$

where $\vec{r}_{ij}$ and $\vec{q}_{ij}$ are the feature vectors of the image object and the query object, respectively, similarities are computed separately for each feature class.

Again for each feature class, the image objects $O_k$ are sorted yielding several ranked lists $L_{ij}$. Then, the ranks of the objects are linearly combined which produces an overall similarity ranking of the image objects $O_k, k = 1, \dots, n$, of the database with respect to the query object. Since the user of a content-based retrieval system will only want to see reasonable matches, only the $l$ most similar images (where $l \ll n$) will be selected from the database and displayed on the screen.

### *2.2. Adaption from relevance feedback*

Iterative improvement during content-based image retrieval requires relating the user's high-level conception to low-level visual features. This is realised by means of relevance feedback. The user can rate objects in the current result list using scores $V \in \{2, 1, 0, -1, -2\}$ which represent ratings from *highly relevant* to *highly nonrelevant*.

Figure 3: Exemplary query images taken from a database of 1250 images from 10 different domains.

Preserving the information of previous search steps is accomplished by adapting the feature weights $W_{ij}$. Weights of features that allow the distinction of relevant and non-relevant images and thus allow to characterise the user's intention are increased, others are decreased:

$$W'_{ij} = W_{ij} + \epsilon \sum_{k=1}^{l} V(O_k) \cdot \lambda(\rho(O_k, L_{ij})). \qquad (2)$$

Here, $V(O_k)$ is the score of image object $O_k$ assigned by the user. $\rho$ represents the rank of image object $O_k$ in the feature-dependent, ascendingly ordered result list $L_{ij}$. $\lambda$ is a continuous descending function and $\epsilon$ is a learning rate.

Adopting another idea by Rui and Huang [10], the dissimilarity measures are refined as well. The matrix $\mathbf{\Omega}_{ij}$ is adapted using the covariances of the feature vectors of to the image objects rated to be *relevant* or *highly relevant*. Finally, a query vector adaption is applied where the query vectors in the feature spaces $R_{ij}$ are slowly moved towards feature vectors of *relevant* and *highly relevant* image objects [11, 12].

### 2.3. Evaluation of the CBIR components

The adaptability of the INDI system was evaluated in different query tasks which were formulated as category searches like "show me images resembling *Q*." Independence of the image domain was ensured by testing different categories, namely, *autoracing*, *flowers*, and *golfing* examples of which can be seen in Figure 3.

Following the usual custom in information retrieval, a *precision* value was applied to evaluate efficiency and effectiveness. For the *s*th step of an interactive query, it is defined as

$$\text{precision}(s, t) = \frac{N_{s,t}}{t}, \qquad (3)$$

where $N_{s,t}$ represents the number of correct category images retrieved in session $s$ within the first $t = 1, \ldots, l$ retrieved images.

The adaptivity of our system is illustrated in Figure 4. It shows the evolution of the precision values for $l = 27$ returned images over sequences of six interactive retrieval steps.

### 2.4. User interface

In order to enable easy and intuitive handling, the INDI system provides different modalities for interaction. Except for the mouse, there also is a touch screen facility. Both input devices enable the selection of images or image regions. They can be used to rate displayed database content and to initiate further selections from the database. Furthermore, a speech recognition component developed by Fink and colleagues was integrated whose core component is a statistical speech recogniser based on hidden Markov models [13].

Often, it is natural to use several input modalities simultaneously. For instance, users may point to the screen saying things like "this image." Therefore, a hierarchical event handling module was developed that can fuse asynchronous input events from different sources [14].

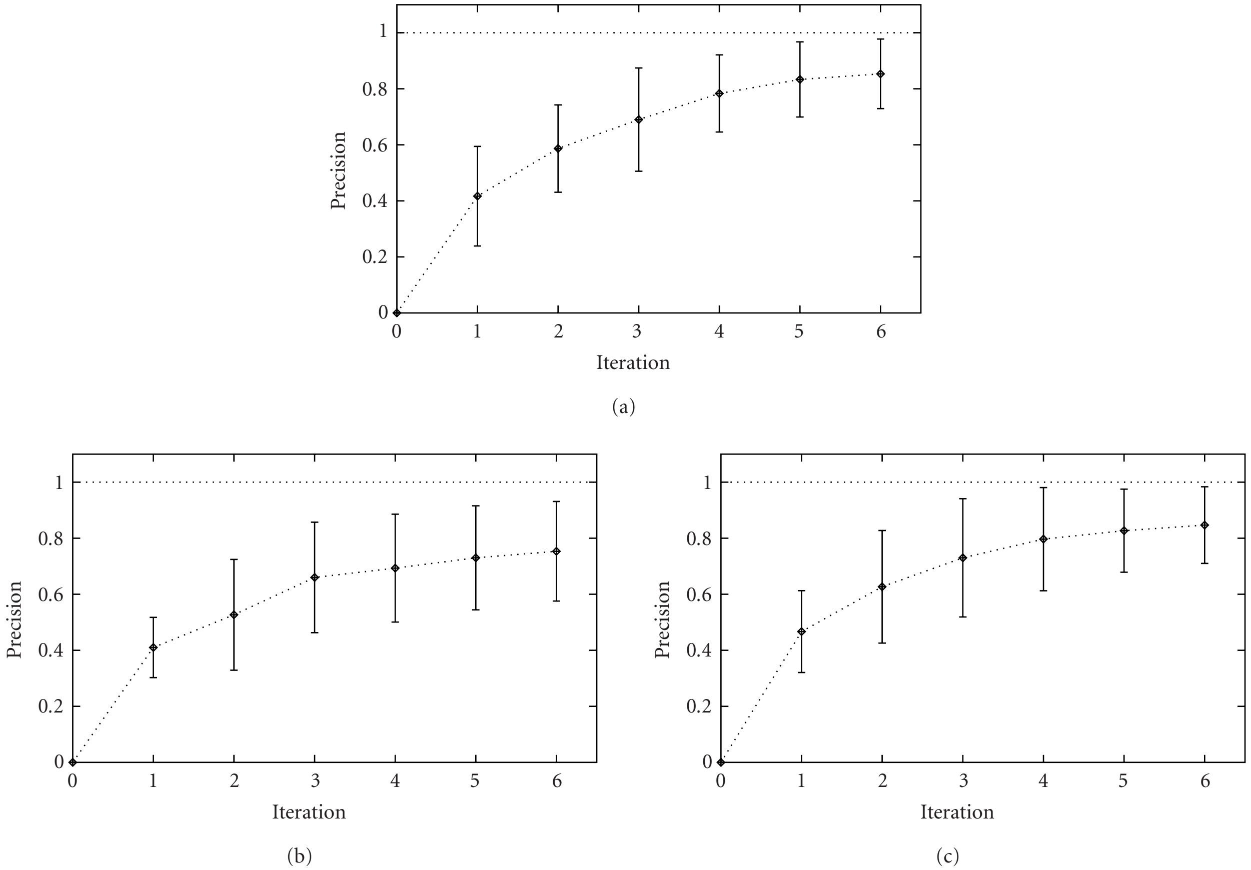
Given all these input devices, the system must be able to relate verbally uttered commands to currently selected images or image regions in order to comprehend the user's intentions. However, fusing results from speech and vision processing suffers from uncertainties like erroneous recognition or partial or unspecific descriptions. Consequently, we treat the task of speech and image integration as a *probabilistic decoding process* which is modelled using Bayesian networks (cf., e.g., [15]).

Adopting algorithms developed by Wachsmuth [16], each region description recognised in an utterance and each region detected in an image, are represented as separate subnetworks. Matches between attributes obtained from speech recognition and those derived from image processing can be found by means of the relations in the network. After the relaxation of such a network, regions intended by the user will have the highest joint probability of being part of the image and also being referred to in an utterance [11].

## 3. VAMPIRE SYSTEMS

In this section, we will describe how the concept of human-machine interaction for computer vision can be extended to higher cognitive levels. While the previous section demonstrated how interaction can trigger adaption on the mere feature level of vision, this section will introduce *cognitive* vision systems that can learn new concepts and can adapt to a physical environment. We will present two systems that are being developed in a research project called Visual Active Memory Processes for Interactive REtrieval [17]. Both systems are able to recognise objects and activities in an unconstrained office environment. They can be operated using speech and gesture. Both make use of the principle of multiple computations and store results from different perceptual modules in a hierarchically organised memory. Processes registered in the memories apply contextual reasoning to verify the consistence and correctness of the incoming data. The memories themselves coordinate the registered processes and provide a notification mechanism to activate them if the memory content requires it. As such a memory is thus not a passive unit but rather is another active component of a system, we call it a visual active memory (VAM).

The VAM demonstrator shown in Figures 5 and 6 analyses video signals from calibrated static cameras. Figure 5a depicts a human sitting in front of an office desk which is monitored by two cameras. One is observing the scene from



(a)

(b)

(c)

Figure 4: Adaptation to the user's intention expressed in terms of the evolution of the precision values in different category searches: (a) autoracing, (b) flowers, and (c) golfing. Beginning with the second out of six search steps, positive feedback was provided. The depicted precision values are averaged over 10 experiments.

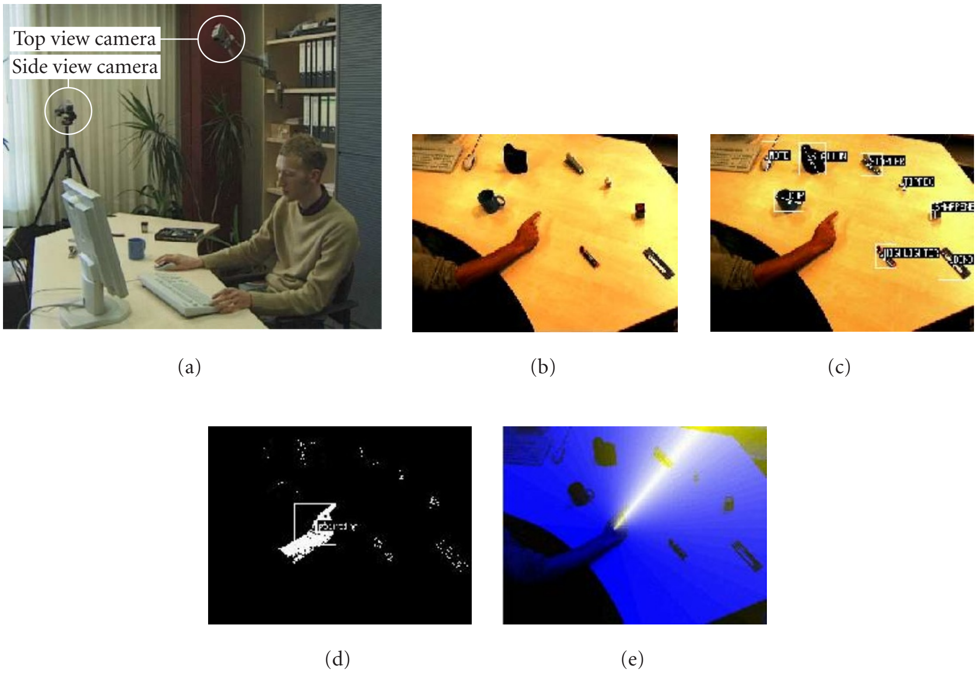
above the other provides a side view of the desk. Figure 5b shows a snapshot recorded with the top-view camera. In this example, the user is pointing to one of the objects on the desktop. In Figure 5c, the results of a view-based object recognition algorithm are cast into the image and Figure 5d displays the results of skin colour segmentation and hand detection. As the index finger is stretched out, a gesture recognition algorithm identified a pointing gesture. Figure 5e visualises the angular probability distribution that indicates the most likely direction of this gesture.

Figure 6a exemplifies the side view on the scene. This viewpoint is used to recognise actions and activities. Figure 6b shows a skin colour segmentation procedure for this example. While larger regions are assumed to depict faces, smaller ones are assumed to represent hands. In Figure 6c, the trajectory of one of the hands is cast into the image. Such trajectories are analysed by a module for action recognition. Furthermore, we see a fan projected into the middle of the image. It indicates the image area near a moving hand where the system expects objects which might be manipulated next. According to the text displayed at the top of the image, the activity that was recognised last in this example was "reach middle" and the object that is currently expected to be manipulated is a cup.

Figure 7 shows the interaction with the mobile VAM demonstrator that was introduced in Figure 1b. By means of verbal commands or pointing gestures, the user can browse through a command menu displayed on the right of his field of view. Selecting or deselecting menu buttons activates different operational modes of the system. Pointing gestures may also be used to reference objects or regions of interest in current the field of view. This resembles the use of the touch screen discussed in the last section. Here, however, space is becoming the interface; gestures are no longer bound to the operation of a physical input device.

### 3.1. Architecture and components

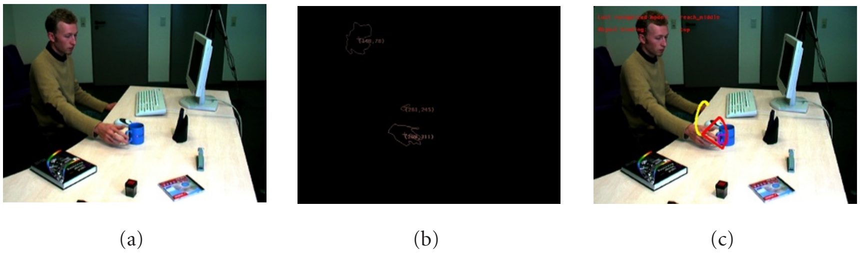
Figure 8a sketches the conceptual architecture of our systems. In the centre, we recognise the memory component. It is organised hierarchically and is able to store image data (i.e., patches cropped from images) and feature-based object descriptions as well as more abstract descriptions of observed events or categories. Several computational modules are grouped around the memory. Note that there is no direct communication between these modules but all data exchange is mediated through the memory. Also note that some of the building blocks represent several algorithms running in parallel.



(a) (b) (c)

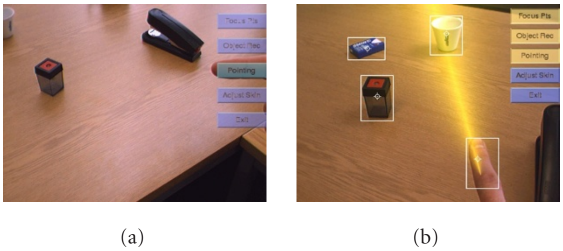
(d) (e)

Figure 5: (a) VAM demonstrator with two static cameras monitoring a human sitting at an office desk. Exemplary results from processing top-view images: (b) gesture seen from above, (c) object recognition results, (d) skin colour detection, and (e) estimated pointing cone.



(a) (b) (c)

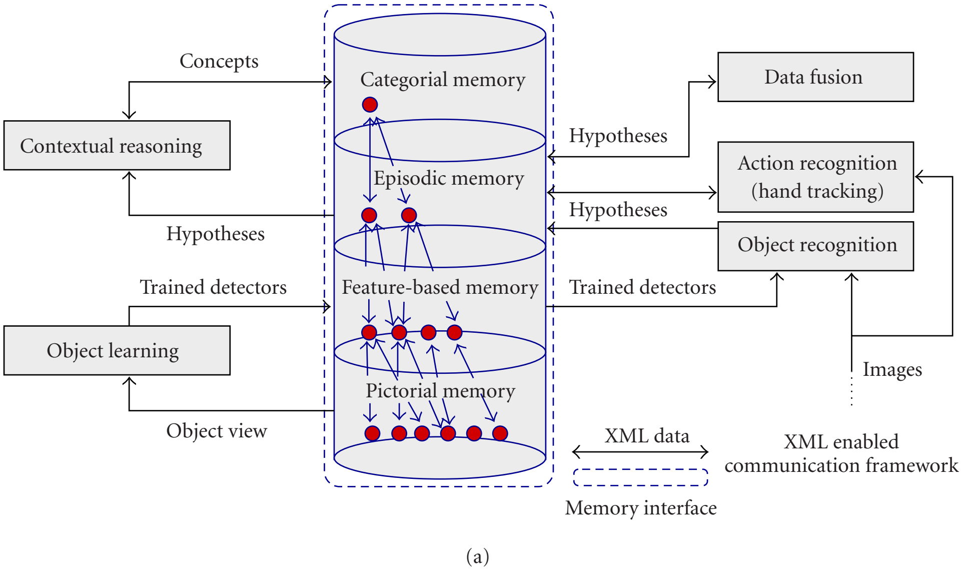
Figure 6: Office scene and results obtained from the side-view camera. (a) Side view of the office scene. (b) Skin coloured regions. (c) Results from action recognition.



(a) (b)

Figure 7: Office desk as seen through the mobile memory spectacles shown in Figure 1b. (a) Menu selection using pointing gestures. (b) Object referencing using pointing gestures.

All algorithms perform in real time and run simultaneously. As we will detail below, the results they forward to the *active memory* are not considered as irrevocable facts but as hypotheses. Processes registered on the database that provides the infrastructure for the memory continuously verify the consistency of incoming hypotheses and assign them a reliability. Corresponding hypotheses from different object recognition modules as well as from the action or gesture recognition components are fused into single abstract descriptions of the scene content. Moreover, since earlier results



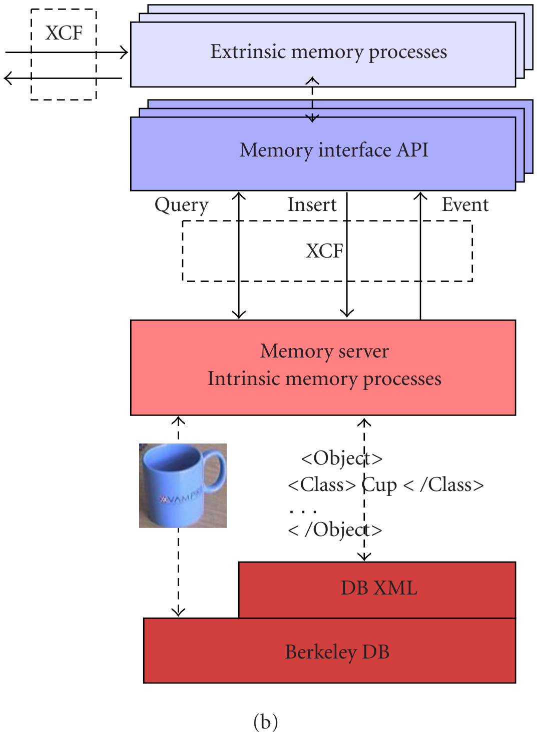


FIGURE 8: (a) Conceptual architecture of the current VAMPIRE demonstrators and (b) active memory infrastructure.

are stored in the memory, temporary occlusion or misinterpretations of the current scene can be filtered out using temporal context. Next, we will outline the applied algorithms and technologies. For implementation details, please refer to [18, 19] for the static and mobile systems, respectively.

### 3.1.1. *Object recognition*

For object recognition, the VAMPIRE systems employ appearance-based methods. On the one hand, VPL classifiers as introduced by Heidemann et al. [20] are applied. First, combining local entropy, symmetry, and edge and corner detection, a saliency value is calculated for each image pixel. Where there is high saliency, patches are cropped from the image and classified in a three-step procedure using vector quantisation, PCA, and LLM neural networks. On the other hand, we also use cascaded weak classifiers (cf. [21, 22]) for object recognition. For each object, windows of different sizes are shifted over the image. For each window, simple texture features are fed into the cascade. Already, in the first layer, most windows not depicting an object are rejected.

Windows successfully passing through the whole cascade depict a known object. Either method is initially trained given manually labelled views of objects which were recorded in different positions and under varying illumination.

Both methods allow for interactive online object learning. Two techniques are being used. Either, the mobile AR-gear is used to focus on an unknown object. To acquire useful views of the object, template-based image feature tracking as proposed by Gräßl et al. [23] compensates head movements. The second method incorporates the pointing mechanism described above. Introducing a rejection class label that is assigned to salient image regions which cannot be classified, these regions can be pointed to. If the user then moves the referred object to produce different views, the system can acquire a series of exemplary image patches. Randomly warping and distorting them yields artificial views which are then used to retrain the classifiers [24]. In either case, object labels are assigned verbally; to this end, the systems are equipped with a speech recognition component [13] that was already mentioned in Section 2.

### 3.1.2. Gesture and action recognition

Both, gesture as well as action recognition, rely on the detection of skin coloured image regions. To ensure robustness, we apply adaptive skin colour segmentation based on Gaussian mixtures models as described by Fritsch [25]. The mobile system provides yet another way for skin colour adjustment. After selecting a command for colour retraining from the interaction menu, moving the hand in front of the head mounted cameras produces data required for the adaption. Skin coloured image patches of a certain size are analysed by a VPL classifier which decides whether they depict a hand or a even a pointing gesture.

Our action recognition framework is based on CONDENSATION particle filtering as introduced by Isard and Blake [26]. Black and Jepson [27] adapted this approach to the classification of hand trajectories. Using parameterised trajectory models, their techniques enable the recognition of activities solely on the basis of hand motions without incorporating any kind of context. For instance, "pick" motions can be detected without information about *what* part has been taken.

In [25], Fritsch proposes an extension to the work of Black and Jepson in order to incorporate contextual knowledge. He distinguishes the *situational* context and the *spatial* context of a gesture.

The situational context of a gesture describes its necessary preconditions as well as the effect the gesture has on the scene. The spatial context of a gesture relates hand trajectories to objects being manipulated. Obviously, these objects must be close enough to a hand trajectory to be touched or picked for interaction. Therefore, we define a *context area* to be the image area depicting objects potentially relevant for a specific gesture. The context area is given as a circle segment of a certain radius and angle. For interaction with objects that do not have an intrinsic "handling direction", its orientation is defined relative to the moving direction of a hand. For objects that have an intrinsic "handling direction," the context area has an absolute orientation. Besides defining *where* symbolic context is expected, we need to specify *what* context is expected. This includes the relevance of the context (irrelevant, necessary, or optional) as well as the type of the context object.

Actually incorporating context into recognition is done in two ways: the situational context is applied in the select step of the particle filter in order to initialise and select only those samples whose preconditions match the current situation. The spatial context is taken into account in the update step where it changes the weights of samples that match the observations. The calculation of sample weights is extended by a multiplicative *context factor* representing how well the observed scene fits the expected symbolic context.

### 3.1.3. Probabilistic information fusion

Due to flawed results of the perceptual modules or to a change in the environment, it might occur that hypotheses stored in the memory contradict one another. Consistency validation has to detect such conflicts and resolve them. As motivated in [28, 29], elements in the memory are stored as XML fragments. Apart from information describing objects, these fragments also contain metadata like, for instance, the reliability of a hypothesis. An intrinsic memory process that lowers the reliability of stored data guides the removal, that is, the *forgetting*, of conflicting hypotheses. The risk of conflicting results from object and action recognition is minimised by considering contextual and functional relations among incoming hypotheses. As they easily integrate different types of information, we apply Bayesian networks to model dependencies among the various facts our system gathers during runtime.

Consistency validation is realized as a memory process that uses *functional dependency concepts* (FDCs) to rate stored hypotheses. FDCs basically consist of Bayesian networks that model expectations for the relations between specific types of hypotheses.

As an example, consider a situation where the user is sitting in front of a terminal and occasionally performs an action called "typing." Images of this situation that were recorded with a head-mounted camera are shown in Figure 9. Recognising a "typing" action is reasonable only under certain contextual prerequisites. For example, if there is no keyboard in the scene, "typing" hypotheses have to be doubted. Figure 10 shows a Bayesian network and the corresponding conditional dependency tables used to represent contextual prerequisites for the "typing" action.

Nodes with the prefix vis_ denote *observable* variables, whereas exist_-nodes are *hidden* and can only be inferred by the process. Inferring a computer, for instance, requires the observation of a keyboard, a mouse, and a monitor. The object context required by a "typing" action is modelled as a directed arc from the action node exist_A_typing to the object node exist_O_computer.

The power of this approach lies in its applicability to any functional context. It allows for top-down as well as for bottom-up control and, as described in [30], this representation of contextual knowledge can guide object recognition

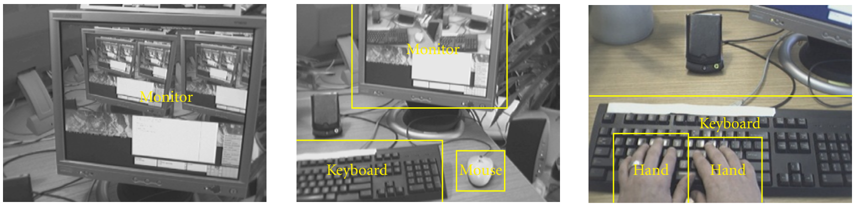


Figure 9: Three images of a sequence with annotated observations.

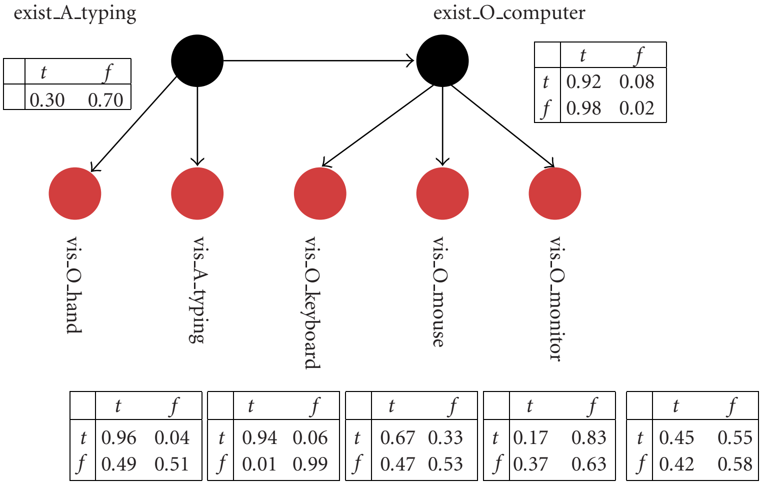


Figure 10: Bayesian network for a computer setup scenario.

and scene understanding. Conflicting memory content is detected as follows: for a given VAM content, the variables of an FDC are assigned *evidences* $\mathbf{e} = \{e_1, e_2, \ldots, e_m\}$. From evaluating the whole network, a conflict value conf can be calculated as a kind of emergence measure defined in [31]:

$$\operatorname{conf}(\mathbf{e}) = \log_2 \frac{\prod_{i=1}^{m} P(e_i)}{P(\mathbf{e})}. \tag{4}$$

Here, $P(\mathbf{e})$ denotes the overall probability of the given evidences while the $P(e_i)$ are the marginal probabilities of the involved random variables of the Bayesian network. If there is a conflict, the probability $P(\mathbf{e})$ is expected to be small compared to the product of the probabilities $P(e_i)$ because in this case the evidences are not explained by the given FDC. Therefore, we will have $\operatorname{conf}(\mathbf{e}) > 0$ which allows the detection of conflicts.

In order to cope with uncertainty of the underlying perception processes, *soft evidences* are used for the observable nodes. Their variables are assigned an evidence vector $\vec{e} = (e_{\text{true}}, e_{\text{false}})^T$ with $0 \leq e_i \leq 1$ and $\sum e_i = 1$. A node's evidence is controlled by the reliability of the corresponding hypothesis. The more reliable the hypothesis is, the *harder* is its evidence. Evidences are set according to

$$(e_{\text{true}}, e_{\text{false}})^T = (0.5(1+r), 0.5(1-r))^T. \tag{5}$$

Thus, for a reliability $r = 1$, the evidence is set to $\vec{e} = (1,0)^T$ while $r = 0$ will yield $\vec{e} = (0.5, 0.5)^T$ which is equivalent to an unobserved variable with no evidence. Details on the lowering of reliabilities in case of conflicts can be found in [32].

Probabilities for the conditional dependencies of the networks were estimated from manually annotated or correctly preprocessed video data. Figure 9 shows three out of 700 training images for the network in Figure 10. If all nodes of the network are observable, parameter estimation simply means counting the different configurations. Otherwise, with some nodes being not observed, an EM algorithm is used (cf. [33]).

To evaluate our consistency validation approach, we defined FDCs for different constellations of objects and actions that are typical for an office scenario. Figure 11 displays prototypic results for the FDC of the "typing" action.

Figure 11a depicts a situation corresponding to a consistent memory content. It shows highly reliable hypotheses vis_O_monitor and vis_O_keyboard, which mutually support each other. Note that $\operatorname{conf}(\mathbf{e}) < 0$.

On the other hand, the configuration in Figure 11b represents a conflict leading to $\operatorname{conf}(\mathbf{e}) > 0$. In this example, there are hypotheses of a monitor and a "typing" action but no hypothesis for the keyboard which violates the expectation that a keyboard should be visible while typing.

### 3.2. *System integration*

Developing complex vision systems is not only a matter of conceptual design but also a software engineering task.

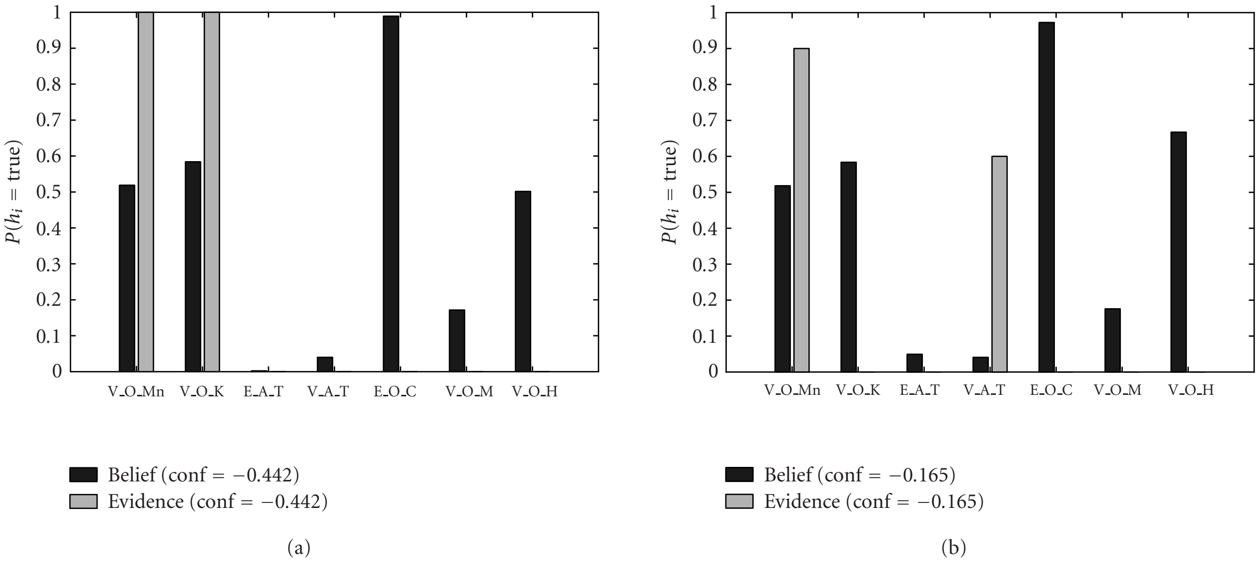


Figure 11: Two examples of beliefs and conf-value for the FDC of the "typing" action. (a) Fitting set of hypotheses. (b) Conflict, weak reliability

Concerning the development of a VAM, there are two major issues: (i) information storage and data organisation for the VAM and (ii) a suitable communication framework allowing to distribute the different algorithms over several computers.

### 3.2.1. VAM infrastructure

Since it is very flexible and suited for abstract concept descriptions, XML was chosen to describe content stored in the memory. Thus, a schema for symbolic data derived from vision algorithms (e.g., objects, actions, etc.) was developed whose instance documents are composed of common and specific element structures (e.g., metadata-like reliability values). Beyond the simple and self-describing nature of XML documents, this has several other advantages. For example, the partition into common and specific elements is beneficial for the realisation of generic software modules where schema evolution allows for extensibility and XQuery/XPath techniques provide standardised access and selection mechanisms.

According to these consideration, a native XML database [34] provides the basic infrastructure for the VAM. On top of this embedded library, a server architecture as shown in Figure 8b was implemented that provides data management not only for XML but also for referenced binary data. Thus, pictorial data can also be used in the active memory and shared by several processes in parallel. Reference management is carried out using RDF information that links symbolic vision data to pictorial memory data. For both kinds of data, powerful standard DBMS methods like *insert*, *update*, *remove*, and *query* are exposed. Node selection and referral is based on XPath statements.

Within this active memory server, for reasons of close coupling and performance, a run-time environment for intrinsic memory processes like *forgetting* or other, more generic, statistical processes was realised. Typical scenarios are small, fast computations that work on large subsets of the system data. Furthermore, a subscription model for distributed event listeners was implemented, so that memory events can trigger registered processes and the memory indeed becomes *active*. Though realized in C++ there also is a Matlab interface for rapid prototyping of further recognition or active memory components.

### 3.2.2. Communication framework

Faced with the problem of distributing the algorithms discussed above over different machines in order to guarantee real-time performance, a comparative study of existing framework technologies was carried out [35]. It yielded that by now there is no suitable integration framework tailored to the needs of cognitive vision. As most vision researchers are not middleware experts, the use of CORBA, for example, was ruled out due to its complexity and bloated standardisation. Rather, owing to the academic background of this work, an integration framework for an agile software process (cf. [36]) is needed.

This led to the development of an XML enabled communication framework (XCF) based on the Internet communication engine [37]. It provides an easy-to-use middleware for building distributed object oriented systems. Its architecture features a pattern-based design and offers communication semantics like (a)synchronous streams, remote procedure calls, and event channels. Similar to the data storage in the VAM component of our systems, data exchange between different modules is based on XML but wrapping and transport of binary data (e.g., images) are possible as well. Since interfaces are specified using XML schema, run-time type safety is ensured, rapid prototyping is possible, and interface programming is intuitive even for middleware novices.

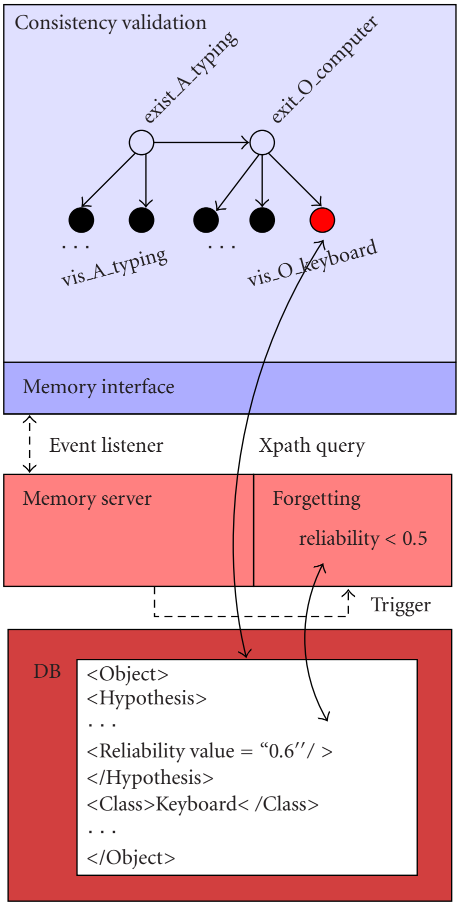


Figure 12: Example of interaction between extrinsic and intrinsic memory processes.

Figure 12 presents a more technical sketch of the consistency validation example discussed above. After an extrinsic memory process, like object recognition, inserts a new hypothesis into the database, consistency validation is triggered. Related database content is queried using XPath and a conflict value is computed. Changes in the reliability values of stored hypotheses will trigger another intrinsic process. If they become too unreliable, hypotheses will be purged from the memory.

This example underlines that, in combination, the XML-based memory infrastructure and the XCF framework enable to realize an architecture with low coupling between components. Furthermore, this decoupling and the capability of the memory to asynchronously gather and provide information yields a high robustness against component failure.

### 3.3. *Technical performance*

Currently, the static system is running on five standard Linux PCs (Pentium 4, 2.4 GHz, 512 MB); images are captured using SONY DFW VL 500 firewire cameras providing a resolution of 640 × 480 pixels. The mobile demonstrator is running on a high-performance DELL notebook (Pentium 4, 1.8 GHz, 512 MB); images are captured from fire-I firewire cameras with a resolution of 320 × 200 pixels.

Evaluating the core components of our systems as if they were stand alone modules yielded the following results: at a frame rate of 4 Hz, the VPL-based recognition of gestures and objects yields an accuracy between 90% and 82% depending on the number of objects that have been trained [24]. The cascaded classifier approach to object recognition processes 6 images per second and yields 92% correctness. Trained with averaged trajectories from different videos and manually annotated information about object context, actions like "drinking from a cup," "reading a book," "phoning" or "typing on the keyboard" can reliably be recognised. A test with 420 sequences yielded an accuracy of 93% [25]. Finally, local queries with low selectivity (approximately 1% of the whole dataset is returned) on a memory instance require an average of 0.57 seconds on a basis of 100, 000 documents in a persistent memory (for an in-depth technical discussion of the evaluation of the XML enabled framework and the memory component, please refer to [28, 29]).

Having read all these figures, it appears that traditional performance assessment does not tell much about the overall performance of an integrated vision system. It is obvious that it does not take into account the continuous nature of human-machine interaction. Interaction with a flexible vision system is a process throughout which there will be mutual adaption. Learning and adaption may improve the system performance over time; recognition and interpretation errors that may appear during an interactive session might be corrected later on.

These considerations thus raise the problem of how to assess the long-term performance of an interactive vision system. Based on the experience reported in the next section, we are tempted to claim that asking the *human* in the loop may provide a solution.

## 4. INTEGRATED SYSTEM EVALUATION

Modern evaluation of intelligent systems for advanced human-machine interaction has a history of about 10 years (cf., e.g., [38, 39]). Proposed approaches range from assessment by means of exemplary benchmarks [40] to the definition of measurable performance indices [41]. However, practical experience with performance measures was not reported. Moreover, neither do the methods known from literature consider situations of triadic interaction, that is, situations where two agents coordinate their perception about a third person, thing, or event, nor do they regard adaptive systems.

In the following, we will outline a holistic evaluation methodology that was applied to assess the capabilities of the INDI system [14]. Apart from collecting technical data like mentioned in the previous section, we also examined the *usability* of our system. To this end, we carried out interactive experiments where we not only measured features like the average success rate in target search but also asked our subjects to fill out questionnaires in order to investigate human factors in interactive image retrieval. This focused on the following criteria adopted from Preece [39].

Table 1: Experimental results with respect to target image.

| Target image | $T_E = \frac{\text{time[s]}}{\text{experiment}}$ | $FB_E = \frac{\text{feedbacks}}{\text{experiment}}$ | $N_I = \frac{\text{iterations}}{\text{experiment}}$ | $T_I = \frac{\text{time[s]}}{\text{iteration}}$ | $FB_I = \frac{\text{feedbacks}}{\text{iteration}}$ |
|---|---|---|---|---|---|
| RaceCar-78 | 73.0 | 9.2 | 2.1 | 33.95 | 4.28 |
| Balloon-36 | 81.3 | 10.8 | 3.3 | 24.65 | 3.29 |
| Flowers-32 | 96.5 | 15.3 | 4.2 | 22.98 | 3.65 |

Table 2: Experimental results with respect to input modality.

| Modality | $S_E$ | $T_E = \frac{\text{time[s]}}{\text{experiment}}$ | $FB_E = \frac{\text{feedbacks}}{\text{experiment}}$ | $N_I = \frac{\text{iterations}}{\text{experiment}}$ | $T_I = \frac{\text{time[s]}}{\text{iteration}}$ |
|---|---|---|---|---|---|
| M | 0.73 | 88.6 | 15.13 | 4.33 | 20.46 |
| T | 0.8 | 71.8 | 9.33 | 2.86 | 25.1 |
| MS | 0.73 | 79.66 | 11.8 | 2.93 | 27.18 |
| TS | 0.67 | 94.4 | 10.93 | 2.73 | 34.57 |

(i) The *speed* of task execution.

(ii) The *functionality* of the system, that is, how many different tasks can be performed?

(iii) The *quality* of the results, that is, how good is the average performance in different tasks?

(iv) The *speed of learning*, that is, how quick can users learn to perform tasks with the system?

(v) The *mental load*, that is, do users have to think carefully while interacting with the system?

(vi) *User satisfaction*, that is, do users like working with the system?

### 4.1. Procedure and design

We considered a database of 1250 images from 10 semantic categories which are taken from the ArtExplosion image collection. A total of 20 computer experienced subjects (2 female and 18 male) who had never before operated a CBIR system were tested. They were divided into four groups of five people each and the input modalities,

(i) *mouse* (M),

(ii) *mouse and speech* (MS),

(iii) *touch screen* (T),

(iv) *touch screen and speech* (TS),

were evaluated. The modalities *mouse and touch screen* as well as *mouse, touch screen and speech* were not examined since initial tests revealed that people never used mouse and touch screen simultaneously.

Each subject took part in three interactive experiments. In each experiment, they were asked to retrieve an image from the database that was shown to them at the beginning (see Figure 13).

In every iteration of an interactive search, 27 images were displayed to the subjects which they could rate in order to navigate through the database and find the query image. They could either score entire images or select certain regions from an image. The maximum amount of time for each experiment was limited to three minutes; if a subject was not able to retrieve the requested image within this time, the experiment was counted as a failure.

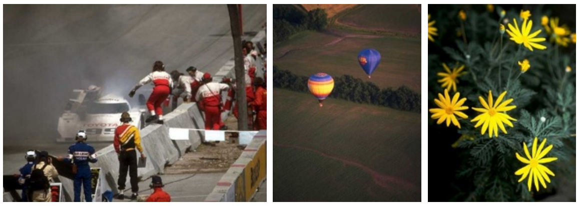


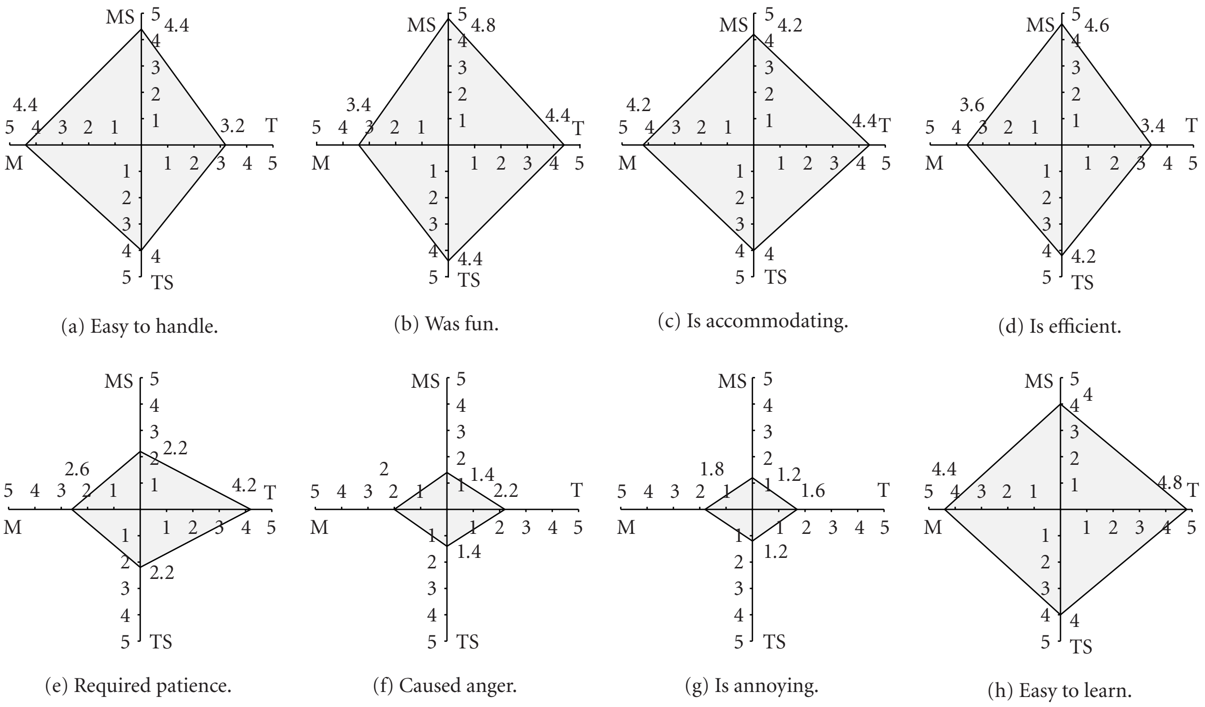
Figure 13: Target images for query tasks.

Besides the success rate $S_E$ averaged over all experiments, the quality of interaction is characterised by the average time $T_E$ the subjects needed to perform an experiment and by the mean number $FB_E$ of user inputs, that is, the amount of feedback provided in an experiment. Given the average number $N_I$ of iterations of a query, it is possible to deduce the ratios $T_I$ and $FB_I$ describing the average time per iteration and number of feedbacks per iteration, respectively. The above-mentioned aspects of learning, mental load, and user satisfaction were examined by means of the questionnaires the subjects were asked to fill out. Faced with statements like “It was fun to interact with the system,” they ranked their sensation on a scale from 1 (no) to 5 (yes).

### 4.2. Results

Tables 1 and 2 and Figure 14 summarise our findings. Looking at the figures in Table 1, it is noticeable that the three target searches were of increasing complexity. This is expressed in the increasing amount of time and feedback as well as in the growing number of interactions shown in the table.

Table 2 lists the figures we measured with respect to the different input modalities. We can see that subjects who only used the mouse provided most relevance feedback but did not achieve the best success rate. We also see that users of the touch screen device performed best and fastest while users of speech and touch screen were the slowest and least successful ones.



(a) Easy to handle.

(b) Was fun.

(c) Is accommodating.

(d) Is efficient.

(e) Required patience.

(f) Caused anger.

(g) Is annoying.

(h) Easy to learn.

FIGURE 14: Averaged results of a questionnaire survey on usability aspects in interactive CBIR. For each interaction modality (mouse (M), touch screen (T), mouse and speech (MS), touch screen and speech (TS)), each aspect had to be rated on a scale from 1 (no) to 5 (yes).

The latter observation is especially interesting if we regard Figure 14. The diagrams in this figure depict the average ranking of the factors asked for in the questionnaires. In Figure 14a, for instance, we notice that the easiness of handling the mouse and the easiness of handling mouse and speech were both ranked 4.4; for the touch screen and speech modality, it yielded 4.0 and the easiness of only using the touch screen reached 3.4, These figures accord with those in Figure 14e which summarise our subjects notion regarding the patience their interaction required. Here, the touch screen users felt that they had to be most patient. Another interesting result becomes apparent from Figure 14d: users of multimodal input devices rated their interaction with our CBIR system to be more efficient than those subjects who only worked with the mouse or touch screen.

### 4.3. Discussion

With respect to our six evaluation criteria, our findings suggest the following. (i) *Speed, functionality, and quality*: concerning the time $T_E$, the number of iterations $N_I$, as well as the number of user feedbacks $FB_E$, performances of monomodal and multimodal interaction diverge. While using mouse and speech is faster than only using the mouse, it is the other way round for using touch screen and speech. However, in any case, different target searches can be performed satisfyingly with regard to the average success as well as to average time need. (ii) *Learnability*: regarding the tested input facilities, users did not sense a significant difference among modalities. (iii) *Mental load*: measured results and user sensations are inconsistent. Even though the touch screen group performed best, their sensations concerning easiness and efficiency were worst. (iv) *User satisfaction*: multimodal input facilities are well appreciated by the users of our system. Even though their results in interactive image retrieval were not the best, the subjects who could use speech and another modality felt least annoyed and considered the interaction they had with the system to be efficient and fun.

## 5. CONCLUSION

This contribution reported on vision systems which make use of the concept of the human in the loop. The first system we described is designed to enable efficient, intuitive, and easy content-based retrieval from image databases. On the one hand, it applies flexible techniques for image feature extraction and adaption on the lower levels of computer vision. On the other hand, it provides several input modalities. Understanding the problem of integrating the different modalities as a probabilistic decoding task enables to fuse the different types into consistent interpretations. As a consequence, natural and seamless interaction with the system becomes possible.

The two other systems we presented follow the cognitive vision paradigm. They are intended to demonstrate the idea of *visual active memory* (VAM). Situated in an unconstrained office environment, both systems recognise typical office objects as well as actions involving them. Information about recognised objects and events is stored in a memory and can be retrieved later on. Both systems are operated using speech or gesture; the mobile demonstrator uses AR technology to display memory content or control interfaces.

Robustness results from applying the principles of multiple computations and contextual reasoning. Different algorithms for object and gesture recognition process image sequences are obtained from different views or from a set of head mounted cameras. The results of these computations are not seen as irrevocable facts but first of all as hypotheses. Hypotheses resulting from recognition processes applied to salient parts of the signal are forwarded to a memory component. There, processes that make use of probabilistic, top-down and bottom-up Bayesian reasoning verify their consistency. As processes like consistency verification and data deletion are triggered by the memory component, the memory indeed is an *active* module.

Basing the memory infrastructure on an XML database and realising the technical system integration using an XML enabled framework results in ease of use, extensibility, and robustness against component failure. Moreover, the human-in-the-loop approach provides an avenue to even more flexibility. While for the image-retrieval system, adaption was only possible by weight adjustment on the feature level of visual processing, the presented VAMs can learn on higher cognitive levels. Through interaction with their users, they can extend preacquired knowledge and learn representations and labels for new objects.

The systems introduced in this contribution thus demonstrate that the goals of the cognitive vision paradigm are not just illusory. Machine learning, contextual reasoning, relevance control, and active system introspection can be brought together and human-machine interaction can compensate for embodiment. And indeed, in combination, these techniques result in integrated systems of high robustness and flexibility.

However, dealing with the evaluation of complex integrated vision systems, human-machine interaction comes along with new challenges. Up to now, there is only scarce literature on how to characterise the mid- and long-term performance of interactive systems. By means of our image retrieval system, we thus exemplified how usability studies might help to assess the cognitive capabilities of artificial systems. As a matter of fact, some of the results are surprising: even though the users of simple interaction devices felt least content with the performance of the system, they performed best. On the other hand, users of input devices of higher cognitive adequacy (natural language) experienced their interaction with the system to be very pleasant and efficient. Even though they practically obtained the worst retrieval results. Therefore, at least for now, it seems fair to conclude that research in cognitive vision must face the fact that cognition first of all lies in the eye of the beholder.

## ACKNOWLEDGMENTS

This work has been supported by the BMB+F under contract 01IB 001B and by the European Union IST 2001-34401 project VAMPIRE. The authors would like to thank Silke Fischer for the valuable support and suggestions she provided for our usability experiments.

## REFERENCES

[1] J. L. Crowley and H. I. Christensen, Eds., *Vision as Process*, Springer-Verlag, Berlin, Germany, 1995.

[2] European Research Network for Cognitive Vision Systems, 2004, http://www.ecvision.info.

[3] H. I. Christensen, "Cognitive (vision) systems," *ERCIM News*, no. 53, pp. 17–18, 2003.

[4] H. Cruse, "The evolution of cognition—a hypothesis," *Cognitive Science*, vol. 27, no. 1, pp. 135–155, 2003.

[5] M. Stricker and A. Dimai, "Spectral covariance and fuzzy regions for image indexing," *Machine Vision and Applications*, vol. 10, no. 2, pp. 66–73, 1997.

[6] S. Brandt, J. Laaksonnen, and E. Oja, "Statistical shape features in content-based image retrieval," in *Proc. 15th IEEE International Conference on Pattern Recognition (ICPR '00)*, vol. 2, pp. 1062–1065, Barcelona, Spain, September 2000.

[7] P. Montesinos, V. Gouet, and R. Deriche, "Differential invariants for color images," in *Proc. 14th IEEE International Conference on Pattern Recognition (ICPR '98)*, vol. 1, pp. 838–840, Brisbane, Queensland, Australia, August 1998.

[8] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, vol. 2, no. 2, pp. 125–137, 2001.

[9] Y. R. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 644–655, 1998.

[10] Y. R. Rui and T. S. Huang, "Optimizing learning in image retrieval," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 1, pp. 236–243, Hilton Head, SC, USA, June 2000.

[11] C. Bauckhage, T. Käster, M. Pfeiffer, and G. Sagerer, "Content-based image retrieval by multimodal interaction," in *Proc. 29th Annual Conference of the IEEE Industrial Electronics Society (IECON '03)*, vol. 2, pp. 1882–1887, Roanoke, Va, USA, November 2003.

[12] T. Kämpfe, T. Käster, M. Pfeiffer, H. Ritter, and G. Sagerer, "INDI—intelligent database navigation by interactive and intuitive content-based image retrieval," in *Proc. IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 921–924, June 2002.

[13] G. A. Fink, "Developing HMM-based recognizers with ESMERALDA," in *Proc. International Workshop Text, Speech and Dialogue (TSD '99)*, vol. 1692 of *Lecture Notes in Artificial Intelligence*, pp. 229–234, Springer-Verlag, Berlin, Germany, September 1999.

[14] T. Käster, M. Pfeiffer, C. Bauckhage, and G. Sagerer, "Combining speech and haptics for intuitive and efficient navigation through image databases," in *Proc. 5th International Conference on Multimodal Interfaces (ICMI '03)*, pp. 180–187, Vancouver, British Columbia, Canada, November 2003.

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Francisco, calif, USA, 1988.

[16] S. Wachsmuth and G. Sagerer, "Bayesian networks for speech and image integration," in *Proc. 18th National Conference on Artificial Intelligence (AAAI '02)*, pp. 300–306, Edmonton, Alberta, Canada, August 2002.

[17] VAMPIRE, http://www.vampire-project.org.

[18] C. Bauckhage, M. Hanheide, S. Wrede, and G. Sagerer, "A Cognitive vision system for action recognition in office environments," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II-827–II-833, Washington, DC, USA, June–July 2004.

[19] G. Heidemann, I. Bax, H. Bekel, et al., "Multimodal interaction in an augmented reality scenario," in *Proc. 6th*

*International Conference on Multimodal Interfaces (ICMI '04)*, pp. 53–60, State College, Pa, USA, October 2004.

[20] 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. 3rd International Conference on Computer Vision Systems (ICVS '03)*, pp. 22–33, Graz, Austria, April 2003.

[21] J. Kittler, A. Ahmadyfard, and D. Windridge, "Serial multiple classifier systems exploiting a coarse to fine output coding," in *Proc. 4th International Workshop Multiple Classifier Systems (MCS '03)*, vol. 2709 of *Lecture Notes in Computer Science*, pp. 106–114, Springer-Verlag, Guildford, UK, June 2003.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. I-511–I-518, Kauai, Hawaii, USA, December 2001.

[23] C. Gräßl, T. Zinßer, and H. Niemann, "Illumination insensitive template matching with hyperplanes," in *Proc. 25th Pattern Recognition Symposium (DAGM '03)*, vol. 2781 of *Lecture Notes in Computer Science*, pp. 273–280, Springer-Verlag, Magdeburg, Germany, September 2003.

[24] H. Bekel, I. Bax, G. Heidemann, and H. Ritter, "Adaptive computer vision: online learning for object recognition," in *Proc. 26th Pattern Recognition Symposium (DAGM '04)*, vol. 3175 of *Lecture Notes in Computer Science*, pp. 447–454, Springer-Verlag, Tübingen, Germany, August 2004.

[25] J. Fritsch, *Vision-based recognition of gestures with context*, Ph.D. thesis, Bielefeld University, Bielefeld, Germany, 2003.

[26] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[27] M. J. Black and A. D. Jepson, "A Probabilistic framework for matching temporal trajectories: CONDENSATION-based recognition of gestures and expressions," in *Proc. European Conference on Computer Vision (ECCV '98)*, pp. 909–924, Freiburg, Germany, June 1998.

[28] S. Wrede, J. Fritsch, C. Bauckhage, and G. Sagerer, "An XML based framework for cognitive vision architectures," in *Proc. 17th IEEE International Conference on Pattern Recognition (ICPR '04)*, no. 1, pp. 757–760, Cambridge, UK, August 2004.

[29] S. Wrede, M. Hanheide, C. Bauckhage, and G. Sagerer, "An active memory as a model for information fusion," in *Proc. 7th International Conference on Information Fusion*, no. 1, pp. 198–205, Stockholm, Sweden, June–July 2004.

[30] K. Murphy, A. Torralba, and W. T. Freeman, "Using the forest to see the trees: a graphical model relating features, objects, and scenes," in *Proc. Advances in Neural Information Processing Systems 16 (NIPS '03)*, Vancouver, British Columbia, Canada, December 2003.

[31] F. V. Jensen, *Bayesian Networks and Decision Graphs*, Information Science and Statistics, Springer-Verlag, 2001.

[32] M. Hanheide, C. Bauckhage, and G. Sagerer, "Memory consistency validation in a cognitive vision system," in *Proc. 17th IEEE International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 459–462, Cambridge, UK, August 2004.

[33] S. Lauritzen, "The EM algorithm for graphical association models with missing data," *Computational Statistics & Data Analysis*, vol. 19, no. 2, pp. 191–201, 1995.

[34] Berkeley DB XML, *Sleepycat Software*, 2004, http://www.sleepycat.com/products/xml.shtml.

[35] S. Wrede, W. Ponweiser, C. Bauckhage, G. Sagerer, and M. Vincze, "Integration frameworks for large scale cognitive vision systems—an evaluative study," in *Proc. 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 1, pp. 761–764, Cambridge, UK, August 2004.

[36] A. Cockburn, *Agile Software Development*, Addison-Wesley, Boston, Mass, USA, 2001.

[37] The Internet Communications Engine, 2004, http://www.zeroc.com/ice.html.

[38] G. Lindegaard, *Usability Testing and System Evaluation: a Guide for Designing Useful Computer Systems*, Chapman & Hall, London, UK, 1994.

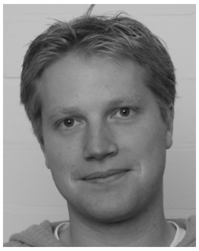
[39] J. Preece, Y. Rogers, and H. C. Sharp, *Beyond Human-Computer Interaction*, John Wiley & Sons, Chichester, UK, 2002.

[40] C.-P. Tung and A. C. Kak, "Integrating sensing, task planning and execution for robotic assembly," *IEEE Trans. Robot. Automat.*, vol. 12, no. 2, pp. 187–201, 1996.

[41] N. Beringer, U. Kartal, K. Louka, F. Schiel, and U. Türk, "PROMISE—a procedure for multimodal interactive system evaluation," in *Proc. Workshop Multimodal Ressources and Multimodal System Evaluation*, Las Palmas, Gran Canaria, Spain, 2002.

**Christian Bauckhage** studied computer science in Bielefeld and Grenoble. He received the Diploma and the Ph.D. degree from Bielefeld University in 1998 and 2002, respectively. Afterwards, he worked in the European Union IST project VAMPIRE. Currently, he is a Postdoctoral Fellow at the Centre for Vision Research, York University, Toronto. He is interested in computer vision and cognitive systems as well as in theory and application of machine learning techniques.

**Marc Hanheide** received the Diploma in computer science from Bielefeld University in 2001. Afterwards, he joined the Research Group for Applied Computer Science in Bielefeld where he is working in the IST project VAMPIRE. He is interested in modelbased object recognition, image processing, and computer vision.



**Sebastian Wrede** received the Diploma in computer science from Bielefeld University in 2002. Afterwards, he joined the Research Group for Applied Computer Science in Bielefeld to work in the IST project VAMPIRE. His research interests are vision systems, software architecture, system integration, middleware, and database technologies.

**Thomas Käster** studied computer science at Bielefeld University. After receiving his Diploma in 2001, he became a Member of the Research Group for Applied Computer Science in Bielefeld where he was working within the BMB+F project LOKI. His research interests are computer vision, pattern recognition, machine learning, image retrieval, and database technologies.

**Michael Pfeiffer** received the Diploma in electrical engineering from the Technical University of Braunschweig in 1994. From 1997 to 2000, he was working in the Real-Time Systems Group, the RWTH Aachen University. He joined the Applied Computer Science Group, Bielefeld University in 2000. There he was working in the BMB+F project LOKI. His research interests are computer vision and intelligent systems.

**Gerhard Sagerer** received the Diploma and the Ph.D. degree in computer science from the University of Erlangen-Nuremberg in 1980 and 1985, respectively. Since 1990, he is a Professor of computer science at Bielefeld University Bielefeld, Germany, where he heads the Research Group for Applied Computer Science. His fields of research are speech understanding, computer vision, and cognitive systems.