

Bernt Schiele
 Computer Science Department
 ETH Zurich
 E-mail: schiele@inf.ethz.ch
<http://www.vision.ethz.ch/schiele>

Abstract

The next generation of computers might be literally wearable. Our vision of such a wearable computing device is an intelligent assistant, which is always with you and helps you to solve your every day tasks. Besides size and power, an important challenge is how to interact with wearable computers. An important aspect and unique opportunity of a wearable device is that it can perceive the world from a first-person perspective: a wearable camera can see what you see in order to analyze, model, and recognize things and people which are around you. In this paper we argue that a promising direction for interaction is to make the computers more aware of the situation the user is in and to model the user's context. Wearable cameras, mounted to the user's glasses, can recognize what the user is looking at, estimate the user's location, and model what the user is doing.

1 Introduction

To date, personal computers have not lived up to their name. Most machines sit on the desk and interact with their owners for only a small fraction of the day. Smaller and faster notebook computers have made mobility less of an issue, but the same staid user paradigm persists. Wearable computing hopes to shatter this myth of how a computer should be used. A personal computer should be worn, much as eyeglasses or clothing are worn, and continuously interact with the user based on the context or the situation. With heads-up displays, unobtrusive input devices, personal wireless local area networks, and a host of other context sensing and communication tools, the wearable computer may be able to act as an intelligent assistant.

In the near future, the trend-setting professional may wear several small devices, perhaps literally built into their clothes. That way, the person may conveniently check messages, finish a presentation or browse the web while sitting on the subway or waiting in line at a bank. Such wearable devices may enhance the person's memory by providing instant access to important information anytime anywhere. Operating these devices however will be an important issue. Often today's computers require your full attention and both hands to be operated. You have to stop everything

you are doing and concentrate on the device [5]. Using speech for input and output will become more popular but may be quite annoying in many situations. Imagine for example your neighbor on a cross Atlantic flight constantly talking and chatting with his or her devices.

Wearable devices promise to be less disruptive, and may interact with people differently from other tools. A computational device that is with you all the time can influence the sense of who you are and what you can do. Just as we have adapted to cellular phones, watches and other personal devices, wearable computers are likely to shape our personal habits around them. Starting with technophiles and migrating to the average person, culture over time will shift to incorporate them. It is too early to tell which approach to wearable design will prove popular. The devices can be built in many ways, and it will take a fashion and style battle to determine what people really want to buy.

Although their potential is vast, many of these devices suffer from a common problem: they are mostly oblivious to you and your situation. They don't know what information is relevant to you personally or when it is socially appropriate to "chime in." The goal in solving this problem is to make electronic aids that behave like a well-trained butler or an intelligent assistant. They should be aware of the user's situation and preferences, so they know what actions are appropriate and desirable – a property we call "situation awareness." They should also make relevant information available before the user asks for it and without forcing it on the user – a feature we call "anticipation and availability."

An important aspect of a wearable device is that it can perceive the world from a first-person perspective: a wearable camera can see what you see and a wearable microphone can hear what you hear in order to analyze, model and recognize things and people which are around you. A promising direction for interaction with wearable devices is therefore to make the computers more aware of the situation the user is in and to model the user's context. Wearable cameras, mounted to the user's glasses, can recognize what the user is looking at and model what the user is doing. Using sensors of various types, the device can also monitor the user's choices and build a model of his or her preferences. A person may actively train the computer by saying, "Yes, that was a good choice; show me more," or "No, never suggest me this again." The models can

also work solely by statistical means, gradually compiling information about the user's likes and dislikes, and coupling those preferences to the context. For anticipation and availability, the wearable device can take a few key facts about the user's situation to prompt searches through a digital database or the World Wide Web. The information obtained in this manner would then be presented in an accessible, secondary display outside the user's main focus of attention.

The importance of context in communication and interface cannot be overstated. In human-to-human communication contextual information such as physical environment, time of day, mental state, and the model each conversant has of the other participants can be critical in conveying necessary information and mood. Using small body-mounted sensors such as cameras may enable wearable computers to model and recognize the context of the user and the situation. As processing power increases, a wearable computer can spend more time observing its user to provide serendipitous information, manage interruptions and tasks, and predict future needs without being directly commanded by the user. This contextual information is one way to achieve seamless interaction with the user. We believe that the use of wearable sensors such as head-mounted cameras or wearable microphones combined with software to model and recognize the user's situation and context has the potential to change human-computer interaction fundamentally.

Obviously, a computer interface which uses contextual and situational information to its fullest is more of a long-term goal than what will be addressed in this paper. However, in the following sections we show how computer interfaces may become more contextually aware through machine vision techniques. In this paper we describe two camera augmented wearable systems. The first system (section 2) uses a head-mounted camera to record and analyze the visual environment of the user as well as to recognize objects the user is looking at. The system can hypothesize which part of the visual environment is interesting to the user and may display information about it when appropriate. The second camera augmented wearable system (section 3) is a computer vision driven assistant for the real-space game Patrol. The goal of this assistant is to track the wearer's location and current task through computer vision techniques and without off-body infrastructure.

2 Recognition of Objects using Wearable Cameras

The first example of a wearable camera augmented computing system is a perceptual remembrance agent, which uses a head-mounted camera to record and analyze the visual environment of the user. In particular a computer vision program recognizes objects in the visual field of view of the user in real-time and displays information the user has associated with them.

An important part of the system is the generic object recognizer which is based on a sound statistical Bayesian framework for object modeling and recognition [9]. Objects are represented by multidimensional

receptive field histograms of vector responses from local neighborhood operators. The approach can be used to determine the most probable object, independent of its position, scale and image-plane rotation. The technique is considerably robust to view-point changes. The probabilistic recognition algorithm can determine the probability of each object based only on a small portion of the image (15%-30%) and is capable to recognize 100 objects correctly in the presence of view-point and scale changes. The recognition system runs at approximately 10Hz.

An application of this camera augmented wearable system is the museum-gallery guide. A museum is a rich visual environment and is often accompanied with facts and details (from a guide, text or web-page) to be associated with the paintings. For example, as you walk around in a museum you can record video clips of a guide's explanation of the paintings. Such video clips can then be associated with the painting itself so that every time you and the wearable system see the painting again the associated video-clip is replayed. The system has been presented publicly several times including SigGraph 1999 (USA), Darpa Image Understanding Workshop 1998 (USA), Nicograph 1998 (Japan), Heinz-Nixdorf Museum Paderborn Podium 1999 (Germany) and Orbit 2000 (Switzerland) and has been used each time by several hundred people.

An important aspect of the system is that it not only recognizes which painting a user is looking at but also knows how long the user actually looked at it. This piece of information can be used directly in various ways: depending on the duration the user looks at a painting the wearable system may offer to deliver more information about that painting for example by accessing the database of the museum. By assuming that the duration of looking at a painting is correlated with the user's interest and by memorizing which paintings the user looked at, the system may be able to profile the interests of the user. Depending on such profiles the system could then suggest other paintings in the museum. The museum could also attempt to create a database of user-profiles, which could be used to give suggestion to new visitors (depending on their user-profile) or to analyze the organization and effectiveness of a particular exhibition. Even though we have not experimented with the above-mentioned extensions of the system intensively we believe that extensions like these will greatly leverage the usefulness and usability of wearable computing devices.

The system's building blocks are depicted in Figure 1. Section 2.1 describes the generic object recognition algorithm and section 2.2 the overall system.

2.1 Generic Object Recognition System

The video camera used by the system is aligned with the line of sight of the user (see figure 1). Therefore, by gazing at interesting objects, the user directs the input to the recognition system which continuously tries to recognize previously recorded objects. The recognition results are then sent to the audio-visual associative memory system which plays the appropriate clip.

The generic object recognition system used has been

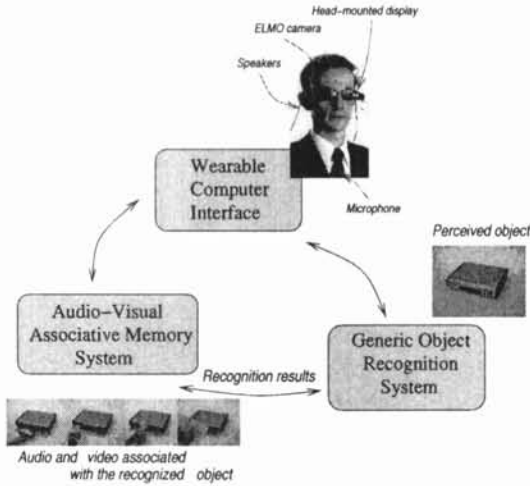


Figure 1: System's architecture

proposed by Schiele and Crowley [8, 9]. A major result of the work is that a statistical representation based on local object descriptors provides a reliable means for the representation and recognition of object appearances.

Objects are represented by multidimensional histograms of vector responses from local neighborhood operators. Figure 2 shows two examples of two-dimensional histograms. Simple matching of such histograms (using χ^2 -statistics or intersection [9]) can be used to determine the most probable object, independent of its position, scale and image-plane rotation. Furthermore the approach is considerably robust to view point changes. This technique has been extended to probabilistic object recognition [9], in order to determine the probability of each object in an image only based on a small image region. Experiments (briefly described below) showed that only a small portion of the image (between 15% and 30%) is needed in order to recognize 100 objects correctly. In the following we summarize the probabilistic object recognition technique used. The current system runs at approximately 10Hz on a Silicon Graphics O2 machine using the OpenGL extension library for real-time image convolution.

Multidimensional receptive field histograms are constructed using a vector of arbitrary linear filters. Due to the generality and robustness of Gaussian derivatives, we selected multidimensional vectors of Gaussian derivatives (e.g. the magnitude of the first derivative and the Laplace operator at two or three different scales).

It is worthwhile to point out that the object representation is very general and can be used for a wide variety of objects. The objects most suited for the representation contain enough local texture and structure to be coded by the multidimensional histograms. A useful feature of the recognition system is that it often matches visually similar objects such as two business cards from the same company. In order to discriminate these cards a more specific system such as a character recognition system should be used. Since the response

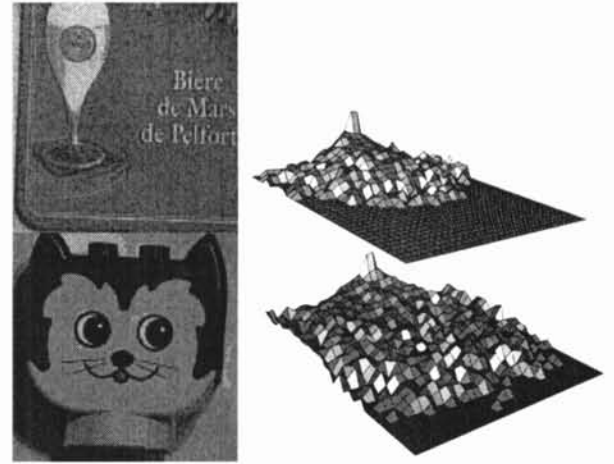


Figure 2: Two-dimensional histograms of two objects corresponding to a particular viewpoint, image plane rotation and scale. The image measurement is given by the Magnitude of the first derivative and the Laplace operator. The resolution of each histogram axis is 32.

time of the system is only in the order of 100ms we are planning to use the result of the system to trigger more specific recognition systems as appropriate.

2.1.1 Probabilistic Object Recognition

In order to recognize an object, we are interested in computing the probability of the object O_n given a certain local measurement M_k (here a multidimensional vector of Gaussian derivatives). This probability $p(O_n|M_k)$ can be calculated using Bayes rule:

$$p(O_n|M_k) = \frac{p(M_k|O_n)p(O_n)}{p(M_k)}$$

with

- $p(O_n)$ the a priori probability of the object O_n ,
- $p(M_k)$ the a priori probability of the filter output combination M_k , and
- $p(M_k|O_n)$ the probability density function of object O_n , which differs from the multidimensional histogram of an object O_n only by a normalization factor.

Having K independent local measurements M_1, M_2, \dots, M_K we can calculate the probability of each object O_n by:

$$p(O_n|M_1, \dots, M_K) = \frac{\prod_k p(M_k|O_n)p(O_n)}{\prod_k p(M_k)} \quad (1)$$

M_k corresponds to a single multidimensional receptive field vector. Therefore K local measurements M_k correspond to K receptive field vectors which are typically from the same region of the image. To guarantee independence of the different local measurements we

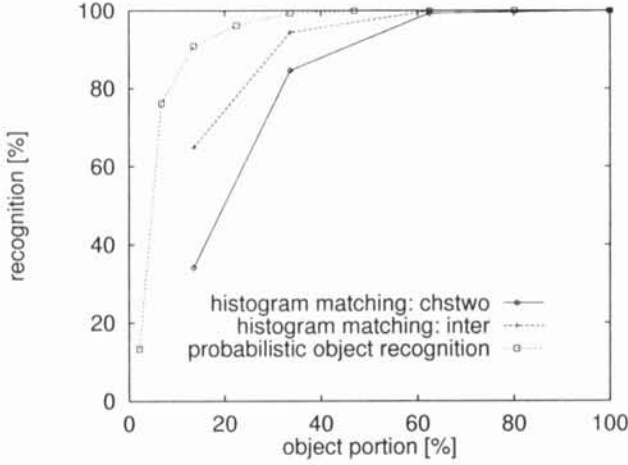


Figure 3: Experimental results for 103 objects. Comparison of probabilistic object recognition and recognition by histogram matching: χ^2_{qv} (chstwo) and \cap (inter). 1327 test images of 103 objects have been used.

choose the minimal distance $d(M_k, M_l)$ between two measurements M_k and M_l to be sufficiently large (in the experiments below we chose the minimal distance $d(M_k, M_l) \geq 2\sigma$).

In the following we assume all objects to be equally probable: $p(O_n) = \frac{1}{N}$ with N the number of objects. We use $p(M_k) = \sum_i p(M_k|O_i)p(O_i)$ for the calculation of the a priori probability $p(M_k)$. Since the probabilities $p(M_k|O_n)$ are directly given by the multidimensional receptive field histograms, Equation (1) shows a calculation of the probability for each object O_n based on the multidimensional receptive field histograms of the N objects. Perhaps the most remarkable property of Equation (1) is that no correspondence needs to be calculated. That means that the probability can be calculated for arbitrary points in the image. Furthermore the complexity is linear in the number of image points used.

Equation (1) has been applied to a database of 103 objects [9]. In an experiment 1327 test images of the 103 objects have been used which include scale changes up to $\pm 40\%$, arbitrary image plane rotation and view point changes. Figure 3 shows results which were obtained for six-dimensional histograms, e.g. for the filter combination $Dx - Dy$ (first Gaussian derivatives in x - and y -direction) at three different scales ($\sigma = 2.0$, $= 4.0$ and $= 8.0$). A visible object portion of approximately 62% is sufficient for the recognition of all 1327 test images (the same result is provided by histogram matching). With 33.6% visibility the recognition rate is still above 99% (10 errors in total). Using 13.5% of the object the recognition rate is still above 90%. More remarkably, the recognition rate is 76% with only 6.8% visibility of the object. See [9] for further details.

2.2 Overview of the system

The following describes the audio-visual association module and gives a short overview of the hardware.



Figure 4: Sample Output Through heads-up-display (HUD)

Audio-Visual Associative Memory System:

The audio-visual associative memory operates on a record-and-associate paradigm. Audio-visual clips are recorded by the push of a button and then associated to an object of interest. Subsequently, the audio-visual associative memory module receives object labels along with confidence levels from the object recognition system. If the confidence is high enough, it retrieves from memory the audio-visual information associated with the object the user is currently looking at and overlays this information on the user's field of view.

Whenever the user decides to record the current interaction, he moves his head mounted video camera and microphone to specifically target and *shoot* the footage required. Thus, an audio-video clip is formed. After recording such a clip, the user selects the object that should trigger the clip's playback. This is done by directing the camera towards an object of interest and triggering the unit (i.e. pressing a button). The system then instructs the vision module to add the captured image to its database of objects and associate the object's label to the most recently recorded A/V clip. Additionally, the user can indicate negative interest in objects which might get misinterpreted by the vision system as trigger objects (i.e. due to their visual similarity to previously encountered trigger-objects). Thus, both positive and negative reinforcement can be used in forming these associations. Therefore the user can actively assist the system to learn the differences between uninteresting objects and important cue objects.

Whenever the user is not recording or associating, the system is continuously running in a background mode trying to find objects in the field of view which have been associated to an A/V sequence. The system thus acts as a parallel perceptual remembrance agent that is constantly trying to recognize and explain – by remembering associations – what the user is paying attention to. Figure 4 depicts an example of the overlay process. Here, in the top of the figure, an “expert” is demonstrating how to change the bag on a vacuum cleaner. The user records the process and then associates the explanation with the image of the vacuum's body. Thus, whenever the user looks at the vacuum (as in the bottom of the figure) he or she automatically sees an animation (overlaid on the left of his field of view) explaining how to change the dust bag. The recording, association and retrieval processes are all

performed online in a seamless manner.

2.2.1 Wearable Computer Interface

It is important to note that any wearable system has to be useful and usable by the person wearing it. Ideally we would like a non-intrusive system that does not require new infrastructure to be incorporated in the environment – such as tags, infrared transmitters, etc. – and which can be used in a seamless way by its user.

Using a camera attached to the user’s eye glasses and the generic real-time computer vision object recognition system described in section 2.1 our system is able to perceive, identify and recognize the objects that the user is looking at. Using such a vision system circumvents many problems associated with tagging technologies, such as cost, size, range, power consumption and flexibility. From a perceptual viewpoint, the system (in the same way as some other wearable systems [3, 10, 7]) sees what the user sees and hears what the user hears, being closer to the user’s perception of the world.







VISUAL TRIGGER	ASSOCIATED SEQUENCE
	
	
	

Figure 5: Associating A/V Sequences to Objects

The primary functionality of the system is implemented in a simple 3 button interface (via a wireless mouse). The user can select from a record button, an associate button and a garbage button. The record button stores the A/V sequence. The associate button merely makes a connection between the currently viewed visual object and the previously recorded sequence. The garbage button associates the current visual object with a NULL sequence indicating that it should not trigger any play back. This helps resolve errors or ambiguities in the vision system. This association process is shown in Figure 5. In the current implementation of the system the interface is literally a three button interfaces. Obviously a small vocabulary speech recognizer could be used to replace the three buttons with spoken words.

2.2.2 Hardware

Currently, the system is fully tetherless with wireless radio connections allowing the user to roam around a significant amount of space (i.e. a few office rooms). Plans for evolving the system into a fully self-sufficient, compact and affordable form are underway. However,

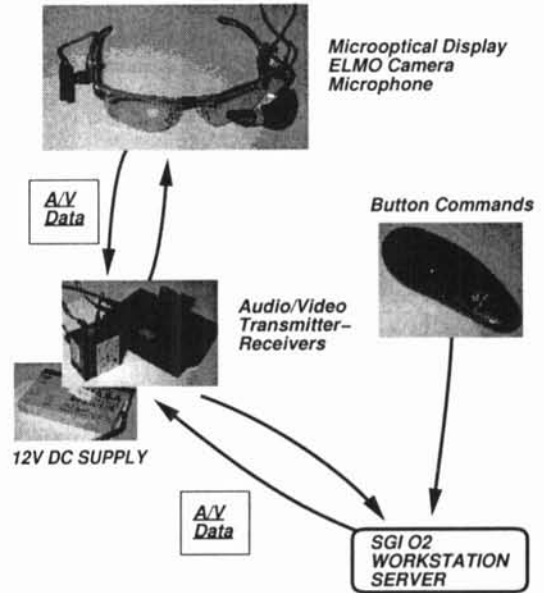


Figure 6: The Wearable Hardware System

for initial prototyping, a wireless system linked to off board processing was acceptable.

Figure 6 depicts the major components of the system which are worn by the user during operation. The user wears standard sun or eye glasses with a MicroOptical display attached. Also attached to the glasses is an ELMO video camera (with wide angle lens) which is aligned as closely as possible with the user’s line of sight [10]. Thus the vision system is directed by the user’s head motions to interesting objects. In addition, a nearby microphone is incorporated. The A/V data captured by the camera and microphone is continuously broadcast using a wireless radio transmitter. This wireless transmission connects the user and the wearable system to an SGI O2 workstation where the vision and other aspects of the system operate. The workstation collects the A/V data into clips, scans the visual scene using the object recognition system, and transmits the appropriate A/V clips back to the user. The clips are then displayed on the user’s MicroOptical. Two A/V wireless channels are used at all times for a bi-directional real-time connection (user to SGI and SGI to user).

3 Recognition of Location and Action of the User with Wearable Cameras

The second camera augmented wearable computing system is designed for the real-space game called *Patrol*. Patrol is a game played by MIT students every week in a campus building and provides a scenario to test techniques in less constrained environments. The participants are divided into teams denoted by colored headbands. Each participant starts with a rubber suction dart gun and a number of darts. After proceeding to the second floor to “resurrect” the teams converge on the basement, mezzanine, and first floors to hunt each

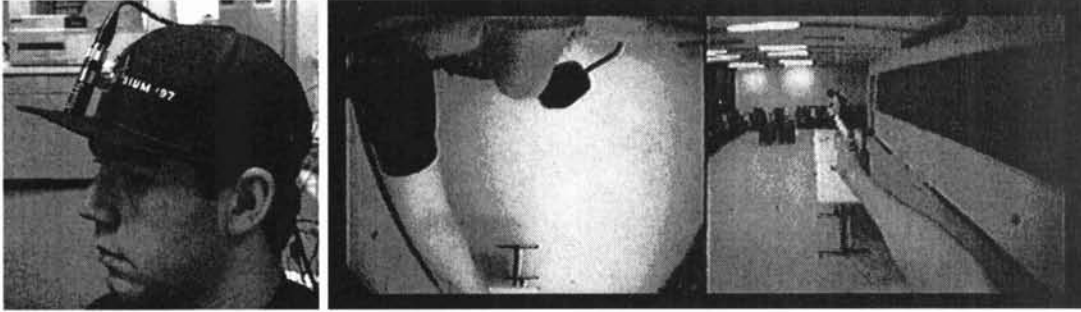


Figure 7: The Patrol cap with two cameras (left). The larger, visible camera points downward and a second, smaller camera faces forward (hidden by the brim). On the right are two images taken with the two cameras respectively.

other. When shot with a dart, the participant removes his headband, and proceeds to the second floor before replacing his headband and returning. While there are no formal goals besides shooting members of other teams, some players emphasize stealth, team play, or holding “territory”. Originally, Patrol provided an entertaining way to test the robustness of wearable computing techniques, such as hand tracking for the sign language recognizer [11]. However, it quickly became apparent that the gestures and actions in Patrol provided a relatively well-defined language and goal structure in a very harsh “real-life” sensing environment. As such, Patrol became a context-sensing project within itself. Here we shortly discuss some work on determining player location and task using purely on-body wearable cameras.

In this scenario, a two cap-mounted cameras perform sensing. The first camera points downwards to watch the hands and body. The other, smaller camera points forward to observe what the user sees. Each camera is fitted with a wide-angle lens (see Figure 7 for sample images). In the Patrol task, location and action of the user is determined solely based on the images provided by those cameras. There are used to determine the location of the player inside the building as well as to model and recognize the action of the player such as aiming, reloading and shooting of the dart guns. This information about the action and location of the player is not only valuable for the wearable computer of the respective player but can be transmitted to other team members. That way other team members are aware of ongoing fights, which team members are involved and how they are positioned to each other. A team strategist may for example deploy this kind of information as appropriate for maintaining territory.

3.1 Location

User location often provides a valuable clue to the user’s context. For example, if the user is in his supervisor’s office, he is probably in an important meeting and does not want to be interrupted for a phone call or an email unless it is an emergency. By gathering data over many days, the user’s motions throughout the day might be modeled. This model may then be used to predict when the user will be in a certain location and

for how long. Such information may be invaluable for network caching in the case that the user’s wireless network does not provide coverage everywhere on a campus. Several options exist for outdoor positioning such as GPS. However, indoor systems are much less prominent. Active badge systems [12] and beacon architectures [4] can trade varying levels of accuracy with the amount of infrastructure that must be installed and maintained. The system described here identifies the user’s location solely based on sensing without need for off-body infrastructure. The Patrol environment consists of 14 rooms that are defined by their strategic importance to the players. The rooms’ boundaries were not chosen to simplify the vision task but are based on the long standing conventions of game play. The playing areas include hallways, stairwells, classrooms, and mirror image copies of these classrooms whose similarities and “institutional” decor make the recognition task difficult. However, four of the possible rooms have relatively distinct coloration and luminance combinations, though two of these are not often traveled.

Hidden Markov models (HMM’s) were chosen to represent the environment due to their potential language structure and excellent discrimination ability for varying time domain processes. For example, rooms may have distinct regions or lighting that can be modeled by the states in an HMM. In addition, the previous known location of the user helps to limit his current possible location. By observing the video stream over several minutes and knowing the physical layout of the building, many possible paths may be hypothesized and the most probable chosen based on the observed data. Prior knowledge about the mean time spent in each area may also be used to weight the probability of a given hypothesis. HMM’s fully exploit these attributes. A full review of HMM’s is not appropriate here, but the reader should see [2, 6] for HMM implementation details and tutorials.

As a first attempt, the mean colors of three video patches are used to construct a feature vector in real-time. One patch is taken from approximately the center of the image of the forward looking camera. The means of the red, green, blue, and luminance pixel values are determined, creating a four element vector. This patch varies significantly due to the continual head motion of the player. The next patch is derived from the downward looking camera in the area just to

the front of the player and out of range of average hand and foot motion. This patch represents the coloration of the floors. Finally, since the nose is always in the same place relative to the downward looking camera, a patch is sampled from the nose. This patch provides a hint at lighting variations as the player moves through a room. Combined, these patches provide a 12 element feature vector.

Approximately 45 minutes of Patrol video were analyzed for this experiment. Processing occurs at 10 frames per second on an SGI O2. Missed frames are filled by simply repeating the last feature vector up to that point. The video is then subsampled to six frames per second to create a manageable database size for HMM analysis. The video is hand annotated using a VLAN system to provide the training database and a reference transcription for the test database. Whenever the player steps into a new area, the video frame number and area name are recorded. Both the data and the transcription are converted to Entropic's HTK [13] format for training and testing.

For this experiment, 24.5 minutes of video, comprising 87 area transitions, are used for training the HMMs. As part of the training, a statistical (bigram) grammar is generated. This "grammar" is used in testing to weight those rooms which are considered next based on the current hypothesized room. An independent 19.3 minutes of video, comprising 55 area transitions, are used for testing. Note that the computer must segment the video at the area transitions as well as label the areas properly.

Table 1 demonstrates the accuracies of the different methods tested. For informative purposes, accuracy rates are reported both for testing on the training data and the independent test set. Accuracy is calculated by

$$Acc = \frac{N - D - S - I}{N}$$

where N is the total number of areas in the test set, D (deletions) is the number of area changes not detected, S (substitutions) is the number of areas falsely labeled, and I (insertions) is the number of area transitions falsely detected. Note that, since all errors are counted against the accuracy rate, it is possible to get large negative accuracies by having many insertions, as shown by several entries of the table.

Table 1: Patrol area recognition accuracy

<i>method</i>	<i>training set</i>	<i>independent test set</i>
1-state HMM	20.69%	-1.82%
2-state HMM	51.72%	21.82%
3-state HMM	68.97%	81.82%
4-state HMM	65.52%	76.36%
5-state HMM	79.31%	40.00%
Nearest Neighbor	-400%	-485.18%

The simplest method for determining the current room is to determine the smallest Euclidean distance

between a test feature vector with the means of the feature vectors comprising the different room examples in the training set. In actuality, the mean of 200 video frames surrounding a given point in time is compared to the room classifications. Since the average time spent within an area is approximately 600 video frames (or 20 seconds), this window should smooth the data such that the resulting classification shouldn't change due to small variations in a given frame. However, many insertions still occur, causing the large negative accuracies shown in Table 1.

Given the nearest neighbor method as a comparison, it is easy to see how the time duration and contextual properties of the HMM's improve recognition. Table 1 shows that the accuracy of the HMM system, when tested on the training data, improves as more states are used in the HMM. This results from the HMM's overfitting the training data. Testing on the independent test set shows that the best model is a 3-state HMM, which achieves 82% accuracy. The topology for this HMM is shown in Figure 8. In some cases accuracy on the test data is better than the training data. This effect is due to the grammar which limits the possible transitions between areas. Once a wrong turn has been made, the system can pass through many areas before converging again with the correct path. The longer the test path, the higher the potential for being misled for extended periods of time.



Figure 8: HMM topology for Patrol.

Accuracy is but one way of evaluating the methods. Another important attribute is how well the system determines when the player has entered a new area. Figure 9 compares the 3-state HMM and nearest neighbor methods to the hand-labeled video. Different rooms are designated by two letter identifiers for convenience. As can be seen, the 3-state HMM system tends to be within a few seconds of the correct transition boundaries while the nearest neighbor system oscillates between many hypotheses. Changing the size of the averaging window might improve accuracy for the nearest neighbor system. However, the constantly changing pace of the Patrol player necessitates a dynamically changing window. This constraint would significantly complicate the method. In addition, a larger window would result in less distinct transition boundaries between areas.

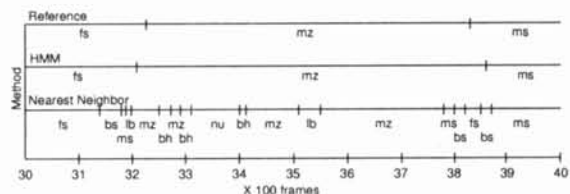


Figure 9: Typical detection of Patrol area transitions.

As mentioned earlier, one of the strengths of the

HMM system is that it can collect evidence over time to hypothesize the player's path through several areas. How much difference does this incorporation of context make on recognition? To determine this, the test set was segmented by hand, and each area was presented in isolation to the 3-state HMM system. At face value this should be a much easier task since the system does not have to segment the areas as well as recognize them. However, the system only achieved 49% accuracy on the test data and 78% accuracy on the training data. This result provides striking evidence of the importance of using context in this task and hints at the importance of context in other user activities.

While the current accuracy rate of 82% is good, several significant improvements can be made. Optical flow or inertial sensors could limit frame processing to those times when the player is moving forward. This would eliminate much of the variation, often caused by stand-offs and firefights, between examples of moving through a room. Similarly, the current system could be combined with optical flow to compensate for drift in inertial trackers and pedometers. Windowing the test data to the size of a few average rooms could improve HMM accuracies as well. Additionally, instead of the average color of video patches, color histograms could be used as feature vectors. Finally, all these techniques could be applied to create an automatic map of a new building as the Patrol player explored it.

3.2 User Tasks

By identifying the user's current task, the computer can assist actively in that task by displaying timely information or automatically reserving resources that may be needed [1]. However, a wearable computer might also take a more passive role, simply determining the importance of potential interruptions (phone, email, paging, etc.) and presenting the interruption in the most socially graceful manner possible. For example, while driving alone in an automobile, the system might alert the user with a spoken summary. However, during a conversation, the wearable computer may present the name of a potential caller unobtrusively in the user's head-up display.

In order to determine the user's action the same recognition system as in section 2 is employed. In the context of the Patrol data the system can be used for recognition of image patches, which correspond to particular appearances of a hand, the gun, a portion of an arm, or any part of the background. Feeding the calculated probabilities as feature vectors to a set of hidden Markov models (HMM's) it is possible to recognize different user tasks such as aiming and reloading. Since aiming and shooting are very similar actions, we consider them as the same task. Aiming can be recognized very well, since it is relatively distinctive with respect to reloading and "everything-else". However, reloading and "everything-else" are difficult to distinguish, since the reloading action happens only in a very small region of the image (close to the body) and is sometimes barely visible. See [10] for a more detailed description of the system and results.

4 Discussion and Conclusions

By observing context, wearable computers can aid in task and interruption management, provide just-in-time information, and make helpful predictions of future behavior. Through head mounted wearable cameras and machine vision techniques, several examples of contextually aware interfaces are presented in this paper.

The systems described above suggest that computer vision can indeed be applied in a wearable setting. An interesting and important aspect of wearable cameras is that they process information, namely visual information, which is extremely familiar to the user. Therefore the system can use visual information not only to analyze, model, and recognize what the user sees but also to communicate with the user. The above systems have used visual information to recognize what the user looks at, to determine the user's location indoors, and to recognize a set of gestures and actions the user is performing. Such information might prove invaluable for novel and interesting interfaces.

Recognizing objects in a wearable scenario demands a high degree of robustness of the object recognition module not only with respect to scale and viewpoint changes but also with respect to realistic environmental changes. We have been able to demonstrate that recognition with wearable cameras is possible in scenes of realistic complexity and of realistic environmental conditions. The number of objects (100+) which can be distinguished is still quite small and does not yet allow large scale deployment. However, by using the user's location and context environment we can implement systems with compelling functionality and context awareness already today.

As an extension of the Patrol task and by using glass-mounted displays, the players could keep track of each other's locations. A strategist can deploy the team as appropriate for maintaining territory. If aim and reload gestures are recognized for a particular player, the computer can automatically alert nearby team members for aid.

But contextual information can be used more subtly as well. For example, if the computer recognizes that its wearer is in the middle of a skirmish, it should inhibit all interruptions and information, except possibly an "X" on the person at whom the user is aiming. Similarly, a simple optical flow algorithm may be used to determine when the player is scouting a new area. Again, any interruption should be inhibited. On the other hand, when the user is "resurrecting" or waiting, the computer should provide as much information as possible to prepare the user for rejoining the game.

The model created by the HMM location system above can also be used for prediction. For example, the computer can weight the importance of incoming information depending on where it believes the player will move next. An encounter among other players several rooms away may be relevant if the player is moving rapidly in that direction. In addition, if the player is shot, the computer may predict the most likely next area for the enemy to visit and alert the player's teammates as appropriate. Such just-in-time information

can be invaluable in such hectic situations.

Additional vision techniques such as optical flow or motion differencing may be added to determine if the user is standing, walking, running, visually scanning the scene, or using the stairs. Ultimately, with development, systems based on vision and other sensor modalities such as accelerometers and microphones might be used to observe and model everyday user tasks and human to human interactions.

5 Acknowledgements

The author would like to express his gratitude to Alex Pentland who provided a very stimulating environment during his stay at the MIT Media Laboratory. In particular the author wants to thank Tony Jebara, Nuria Oliver and Thad Starner, which contributed to the work described in the paper.

References

- [1] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62, 1993.
- [2] X.D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.
- [3] T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland. Stochasticks: Augmenting the billiards experience with probabilistic vision and wearable computers. In *Proceedings of the First Intl. Symposium on Wearable Computers ISWC97*, pages 138–145, Cambridge, MA, 1997.
- [4] S. Long, R. Kooper, G. Abowd, and C. Atkeson. Rapid prototyping of mobile context-aware applications: The cyberguide case study. In *MobiCom*. ACM Press, 1996.
- [5] A. Pentland. Wearable intelligence. *Scientific American presents: Exploring Intelligence*, 9(4), 1998.
- [6] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989.
- [7] J. Rekimoto and K. Nagao. The world through the computer: computer augmented interaction with real world environments. *UIST'95*, pages 29–36, 1995.
- [8] B. Schiele and J.L. Crowley. Object recognition using multidimensional receptive field histograms. In *ECCV'96 Fourth European Conference on Computer Vision, Volume I*, pages 610–619, 1996.
- [9] B. Schiele and J.L. Crowley. Object recognition without correspondence using multidimensional receptive field histograms. *International Journal on Computer Vision*, 36(1):31–50, 2000.
- [10] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Second International Symposium on Wearable Computers*, pages 50–57, Oct 1998.
- [11] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. 20(12):1372–1375, 1998.
- [12] R. Want and A. Hopper. Active badges and personal interactive computing objects. *IEEE Trans. on Consumer Electronics*, 38(1):10–20, Feb. 1992.
- [13] S. Young. *HTK: Hidden Markov Model Toolkit V1.5*. Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, 1993.