M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 465
To appear IEEE ISWC '98

Visual Contextual Awareness in Wearable Computing

Thad Starner, Bernt Schiele, and Alex Pentland

Media Laboratory, Massachusetts Institute of Technology
Cambridge, MA 02139
{testarne,bernt,sandy}@media.mit.edu

Abstract

Small, body-mounted video cameras enable a different style of wearable computing interface. 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 paper introduces an assistant for playing the real-space game Patrol. This assistant tracks the wearer's location and current task through computer vision techniques and without off-body infrastructure. In addition, this paper continues augmented reality research, started in 1995, for binding virtual data to physical locations.

1 Introduction

For most computer systems, even virtual reality systems, sensing techniques are a means of getting input directly from the user. However, wearable computers offer a unique opportunity to re-direct sensing technology towards recovering more general user context. Wearable computers have the potential to "see" as the user sees, "hear" as the user hears, and experience the life of the user in a "first-person" sense. This increase in contextual and user information may lead to more intelligent and fluid interfaces that use the physical world as part of the interface.

The importance of context in communication and interface can not be overstated. 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. An anecdote from Nicholas Negroponte's book "Being Digital" illustrates this point:

> Before dinner, we walked around Mr. Shikanai's famous outdoor art collection, which during the daytime doubles as the Hakone Open Air Museum. At dinner with Mr. and Mrs. Shikanai, we were joined by Mr. Shikanai's private male secretary who, quite significantly, spoke perfect English, as the Shikanais spoke none at all. The conversation was started by Wiesner, who expressed great interest in the work by Alexander Calder and told about both MIT's and his own personal experience with that great artist. The secretary listened to the story and then translated it from beginning to end, with Mr. Shikanai listening attentively. At the end, Mr. Shikanai reflected, paused, and then looked up at us and emitted a shogun-size "Ohhhh."
>
> The male secretary then translated: "Mr. Shikanai says that he too is very impressed with the work of Calder and Mr. Shikanai's most recent acquisitions were under the circumstances of ..." Wait a minute. Where did all that come from?
>
> This continued for most of the meal. Wiesner would say something, it would be translated in full, and the reply would be more or less an "Ohhhh," which was then translated into a lengthy explanation. I said to myself that night, if I really want to build a personal computer, it has to be as good as Mr. Shikanai's secretary. It has to be able to expand and contract signals as a function of knowing me and my environment so intimately that I literally can be redundant on most occasions.

There are many subtleties to this story. For example, the "agent" (i.e. the secretary) *sensed* the physical location of the party and the particular object of interest, namely, the work by Calder. In addition, the agent could attend, parse, understand, and translate the English spoken by Wiesner, *augmenting* Mr. Shikanai's abilities. The agent also *predicted* what Mr. Shikanai's replies might be based on a *model* of his tastes and personal history. After Mr. Shikanai consented/specified the response "Ohhhh," the agent took an appropriate action, filling in details based on a model of Wiesner and Negroponte's interests and what they already knew. One can imagine that Mr. Shikanai's secretary uses his model of his employer to perform other functions as well. For example, he can prevent "information overload" by attending to complicated details and prioritizing information based on its relevancy. In addition, he has the knowledge and social grace to know when and how Mr. Shikanai should be interrupted for other real-time concerns such as a phone call or upcoming meeting.

Obviously, such a computer interface 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. Section 2 describes the importance of object identification in combining the virtual environment of the wearable computer with the physical environment of the user. Section 3.2 details how the location of the user may provide salient cues to his current context. This section also describes a particular implementation for determining location without off-body infrastructure in the context of the real-time, real-space game Patrol. Finally, a means of determining the user's current task in Patrol is discussed in Section 3.3.

2 Identification of Relevant Objects

One of the most distinctive advantages of wearable computing is the coupling of the virtual environment with the physical world. Thus, determining the presence and location of physical objects relative to the user is an important problem. Once an object is uniquely labeled, the user's wearable computer can note its presence or assign virtual properties to the object. Hypertext links, annotations, or Java-defined behaviors can be assigned to the object based on its physical location [19, 7, 8]. This form of ubiquitous computing [22] concentrates infrastructure mainly on the wearer as opposed to the environment, reducing costs and maintenance, and avoiding some privacy issues.

Objects can be identified in a number of different ways. With Radio Frequency Identification (RFID), a transmitter tag with a unique ID is attached to the object to be tracked. This unique ID is read by special devices over ranges from a few inches to several miles depending on the type and size of the tag. Unfortunately, this method requires a significant amount of physical infrastructure and maintenance for placing and reading the tags.

Computer vision provides several advantages over RFID. The most obvious is to obviate the need for expensive tags for the objects to be tracked. Another advantage of computer vision is that it can adapt to different scales and ranges. For example, the same hardware and software may recognize a thimble or a building depending on the distance of the camera to the object. Computer vision is also directed. If the computer identifies an object, the object is known to be in the field of view of the camera. By aligning the field of view of the camera with the field of view of the eye, the computer may observe the objects that are focus of the user's attention.

Figure 1: Multiple graphical overlays aligned through visual tag tracking.

In the past, the MIT Wearable Computing Project has used computer vision identification to create a physically-based hypertext demonstration platform [19, 5] as shown in Figure 1. While this system uses the processing power of an SGI, it maintains the feel of a wearable computer by sending video to and from the SGI and head-mount wirelessly. In this system, visual "tags" uniquely identify each active object. These tags consist of two red squares bounding a pattern of green squares representing a binary number unique to that room. A similar identification system has been demonstrated by Nagao and Rekimoto [8] for a tethered, hand-held system. These visual patterns are robust in the presence of similar background colors and can be distinguished from each other in the same visual field. Once an object is identified, text, graphics, or a texture mapped movie can be rendered on top of the user's visual field using a head-up display as shown in Figure 1. Since the visual tags have a known height and width, the visual tracking code can recover orientation and distance, providing 2.5D information to the graphics process. Thus, graphics objects can be rotated and zoomed to match their counterparts in the physical world. The result may be thought of as a physically-realized extension to the World Wide Web.

This system has been used to give mini-tours of the Media Lab since 1995. Both active LED and passive tags have been used in the past. Whenever the camera detects a tag, the computer juxtaposes a small red arrow on top of that object indicating a hyperlink. If the user is interested in that link and turns to see it, the object is labeled with text. Finally, if the user approaches the object, 3D graphics or a texture mapped movie are rendered on the object to demonstrate its function. Using this strategy, the user is not overwhelmed upon walking into a room but can explore interesting objects at leisure.

This idea continues to develop. The DyPERS system [3] demonstrates how visual tags become unnecessary when a more sophisticated object recognition system (similar to what is used in a later section for gesture tracking) is employed. By determining user location and head orientation using the Locust indoor location system [18] and inertial sensors, strong priors can be established on which objects may be visible. A similar methodology may be used outside with GPS. This lessens the burden on the vision system from trying to distinguish between all potential objects the user may see over the day to the handful that might be currently visible.

Such physically-based hypertext systems are appropriate for tours of a city or a museum. Also, they may create a sense of community by providing a means to share annotations on the physical world asynchronously between people with similar interests, say architecture students or a high school biology class. As reliability and accessibility to wireless networks improves, such systems might be used for repair, inspection, and maintenance of hidden physical infrastructure, such as electrical wiring or plumbing. Similarly, AR systems might be used as navigation guides or task reminders for the mentally handicapped. As recognition performance increases and the hardware costs decline, many new applications will be found for such contextually-aware computing.

3 The Patrol Task

The "Patrol task" is an attempt to test techniques from the laboratory in less constrained environments. Patrol is a game played by MIT students every weekend in a campus building. The participants are divided into teams denoted by colored head bands. Each participant starts with a rubber suction dart gun and a small number of darts. After proceeding to the second floor to "resurrect" the teams converge on the basement, mezzanine, and first floors to hunt each other. If shot with a dart, the participant removes his head band, waits for fighting to finish, and proceeds to the second floor before replacing his head band and returning. While there are no formal goals besides shooting members of other teams, some players maintain a "kill ratio" of the number of players he shot versus the number of times he was shot. Others emphasize stealth, team play, or holding "territory."

Originally, Patrol provided an entertaining way to test the robustness of wearable computing techniques and apparatus for other projects, such as hand tracking for the sign language recognizer [20]. 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. The next sections discuss current work on determining player location and task using only on-body sensing apparatus.

3.1 Apparatus

Sensing for the Patrol task is performed by two hat-mounted cameras (Figure 2). The larger of the two cameras points downwards to watch the hands and body. The smaller points forward to observe what the user sees. Each camera is fitted with the widest angle lens available. Figure 3 shows sample images from the hat. Both cameras require an attached "camera control box" placed in a backpack with a video mixer and a Hi-8 camcorder. The video mixer combines up to four channels of video into one NTSC signal by mapping each channel into one of four quadrants of the screen. The output is then recorded by the Hi-8 camcorder. While it is possible to provide enough on-body computation to run feature detection in real-time, the reference video tape is needed for experimental purposes. The resultant backpack is larger than is desirable for a daily-use wearable computer but allows enough maneuverability for the two to three hours of a Patrol session.

Figure 2: The two camera Patrol hat.

An extended version of the apparatus is currently being tested. In this system, an identical hat is handed to a team-mate or opposing player. The two channels of additional video are sent wirelessly to the first player where they are recorded using the spare channels of the video mixer. Both instrumented players are fitted with noise-cancelling close-fitting microphones to provide two channels of audio. Eventually, this second perspective

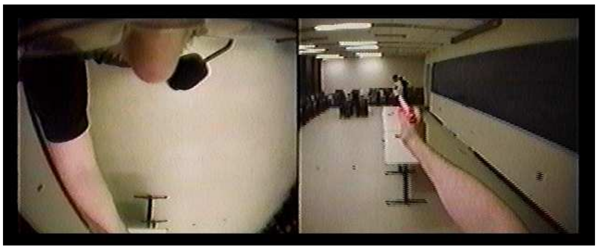

Figure 3: The downward- and forward-looking Patrol views.

may provide interesting data on player interaction, but currently the harsh RF environment causes a significant amount of noise in the system.

3.2 Location

User location often provides valuable clues 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 phone calls or e-mail except for emergencies. 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 [9]. Such information is invaluable for network caching in case the user's wireless network does not provide coverage everywhere on a campus.

Today, most outdoor positioning is performed in relation to the Global Positioning System (GPS). Differential systems can obtain accuracies of less than one meter, and update rates of one second are common. However, indoor systems require different methods. Current systems such as active badges [21, 15, 4, 10] and beacon architectures [6, 14, 18] require increased infrastructure for higher accuracy. This increased infrastructure implies increased installation and maintenance. However, in the Patrol task, we attempt to determine location based solely on the images provided by the Patrol hat cameras, which are fixed-cost on-body equipment.

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 [17, 2, 11] 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 [23] 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

method	*training set*	*independent test set*
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 4. 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.

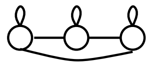

Figure 4: HMM topology for Patrol.

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 5 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.

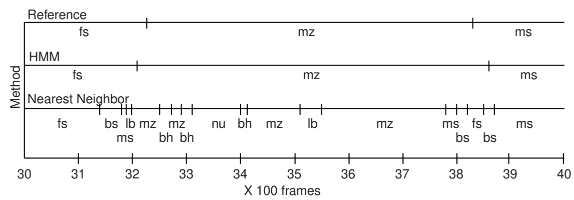

Figure 5: Typical detection of Patrol area transitions.

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 standoffs 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.

3.3 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, 16, 19]. However, a wearable computer might also take a more passive role, simply determining the importance of potential interruptions (phone, e-mail, 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 of an e-mail. However, during a conversation, the wearable computer may present the name of a potential caller unobtrusively in the user's head-up display.

In the Patrol scenario, tasks include aiming, shooting, and reloading. Other user actions such as standing, walking, running, and scanning the environment can be considered as tasks which may be executed simultaneously with the previous tasks. In this section we describe a computer vision system for the recognition of such user tasks. The system is based on a generic object recognition system recently proposed by Schiele and Crowley [13]. A major result of their work is that a statistical representation based on local object descriptors provides a reliable means for the representation and recognition of object appearances.

In the context of the Patrol data this system can be used for recognition of image patches that correspond to particular motions of a hand, the gun, a portion of an arm, or any part of the background. By 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. Preliminary results are described in the next section.

3.4 Probabilistic Image Patch Recognition

Schiele and Crowley [13, 12] presented a technique to determine the identity of an object in a scene using multidimensional histograms of responses of vectors of local neighborhood operators. They showed that matching such histograms can be used to determine the most probable object, independent of its position, scale and image-plane rotation. Furthermore, they showed the robustness of the approach to changes in viewpoint.

This technique has been extended to probabilistic object recognition [13] in order to determine the probability of each object in an image based only on multidimensional receptive field histograms. Experiments showed that only a relatively small portion of the image (between 15% and 30%) is needed in order to recognize 100 objects correctly. In the following we describe briefly the local characteristics and the technique used for probabilistic object recognition. The system runs at approximately 10Hz on a Silicon Graphics machine O2 using the OpenGL extension for real-time image convolution.

Local Characteristics based on Gaussian Derivatives: Multidimensional receptive field histograms can be constructed using a vector of any linear filter. Schiele [12] experimentally compares the invariant properties for a number of receptive field functions, including Gabor filter and local derivative operators. Those experiments showed that Gaussian derivatives provided the most robust and equi-variant recognition results. Accordingly, in the work described in this paper we use filters which are based on equi-variant Gaussian derivatives.

Given the Gaussian distribution $G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$, the first derivative in x and y–direction is given by: $Dx(x, y) = -\frac{x}{\sigma^2}G(x, y)$ and $Dy(x, y) = -\frac{y}{\sigma^2}G(x, y)$ The Laplace operator is calculated as: $G_{xx}(x, y) = (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2})G(x, y)$, $G_{yy}(x, y) = (\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2})G(x, y)$ and $Lap(x, y) = G_{xx}(x, y) + G_{yy}(x, y)$

Probabilistic Object Recognition: In the context of probabilistic object recognition we are interested in the calculation of the probability of the object O_n given a certain local measurement M_k. This probability $p(O_n|M_k)$ can be calculated by the 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 $p(M_k|O_n)$ is 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, ..., M_K we can calculate the probability of each object O_n by:

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

In our context the local measurement 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 the independence of the different local measurements we choose the minimal distance $d(M_k, M_l)$ between two measurements M_k and M_l sufficiently large (in the experiments described below we choose the minimal distance $d(M_k, M_l) \geq 2\sigma$).

For the experiments we can assume that all objects do have the same probability $p(O_n) = \frac{1}{N}$, where N is the number of objects. Therefore equation (1) simplifies to:

$$p(O_n|\bigwedge_k M_k) = \frac{\prod_k p(M_k|O_n)}{\sum_n \prod_k p(M_k|O_n)} \quad (2)$$

In the following we assume the a priori probabilities $p(O_n)$ to be known and 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 tempting property of equation (2) is that we do not need correspondence. That means that the probability can be calculated for arbitrary points in the image.

Equation (2) has been applied to a database of 103 objects. 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 6 shows results which were obtained for six-dimensional histograms, e.g. for the filter combination $Dx - Dy - Lap$ at two different scales ($\sigma = 2.0$ and $= 4.0$). The figure compares probabilistic object recognition and recognition by histogram matching: χ^2_{qv} (chstwo) and $\cap$ (inter). 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

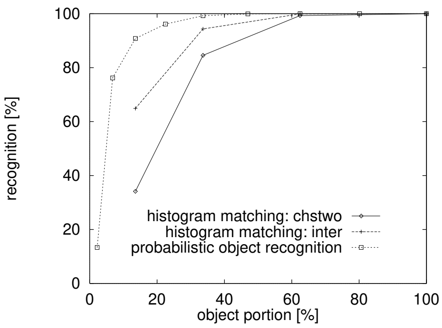

Figure 6: Recognition results of 103 objects.

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.

Task recognition with Hidden Markov Models: The preceding section shortly described a generic object recognition system which is the basis of our computer vision system for the recognition of user tasks. As mentioned above, the recognition system is used for the recognition of image patches which correspond to appearances of a hand, a portion of an arm or any part of the background. In order to use the recognition system we define a library of 30 images (grouped into images corresponding to the same action and chosen arbitrarily from the Patrol data). Each of the images are split into 4x4 sub-images which are used as image patch database. In the experiment below we define three different image groups, one of each action. When applied to the incoming video stream from the camera, the system calculates 3 groups$\times$16 = 48 probabilities at 10Hz. This probability vector is then used as feature vector for a set of HMM which have been trained to recognize different tasks of the user.

Preliminary results: In the following we describe preliminary results for the recognition of user tasks such as aiming, shooting and reloading. Since aiming and shooting are very similar actions, we consider them as the same task in the following.

For both actions (aiming/shooting and reloading) we train a separate HMM containing 5 states. In order to train the HMM's we annotated 2 minutes of the video data. These 2 minutes contained 13 aiming/shooting actions and 6 reloading actions. Everything which is neither aiming nor shooting is modeled by a third class, the "other" class (10 sequences in total). These actions (aiming, reloading and "other") have been separated into a training set of 7 aiming actions, 4 reloading actions and 3 other sequences for training of the HMM's. Interestingly the actions are of very different length (between 2.25sec and 0.3sec). The remaining actions have been used as test set. Table 2 shows the confusion matrix of the three action classes.

	aiming	reloading	"other"
aiming	6	0	0
reloading	0	1	1
"other"	0	1	6

Table 2: Confusion matrix between aiming, reloading, and other tasks.

Aiming is relatively distinctive with respect to reloading and "other", since the arm is stretched out during aiming, which is probably the reason for the perfect recognition of the aiming sequences. However, reloading and "other" 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.

These preliminary results are certainly encouraging, but have been obtained for perfectly segmented data and a very small set of actions. However, one of the intrinsic properties of HMM's is that they can deal with unsegmented data. Furthermore the increase of the task vocabulary will enable the use of language and context models which can be applied on different levels and which will help the recognition of single tasks.

3.5 Use of Patrol Context

While preliminary, the systems described above suggest interesting interfaces. By using head-up 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.

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 team as appropriate. Such just-in-time information can be invaluable in such hectic situations.

4 Conclusion and Future Work

Through body centered cameras and machine vision techniques, several examples of contextually aware interfaces are presented. By observing context, the computer can aid in task and interruption management, provide just-in-time information, and make helpful predictions of future behavior. While larger annotated data sets are necessary to test the techniques used for the Patrol task, the preliminary results are promising. Additional methods 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. By using the new apparatus to analyze video and audio from two simultaneous participants, player interaction might be modeled. Hopefully, with development, such a system will be used to observe and model everyday user tasks and human to human interactions as well.

5 Acknowledgments

Thanks to Jeff Levine who extended the early AR system shown here as part of his MEng and to Steve Mann who demonstrated the advantages of wireless video for prototyping interfaces. Thanks also to Ken Russell who wrote the original AR Open Inventor graphics subsystem and to Tavenner Hall for Figure 5 and for early proof reading.

6 References

[1] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62, 1993.

[2] X. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[3] T. Jebara, B. Schiele, N. Oliver, and A. Pentland. Dypers: dynamic and personal enhanced reality system. Number 463, 1998.

[4] M. Lamming and M. Flynn. Forget-me-not: Intimate computing in support of human memory. In *FRIEND21: Inter. Symp. on Next Generation Human Interface*, pages 125–128, Meguro Gajoen, Japan, 1994.

[5] J. Levine. Real-time target and pose recognition for 3-d graphical overlay. Master's thesis, MIT, EECS, May 1997.

[6] 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.
[7] S. Mann. Smart clothing: The wearable computer and wearcam. *Personal Technologies*, March 1997. Volume 1, Issue 1.
[8] K. Nagao and J. Rekimoto. Ubiquitous talker: Spoken language interaction with real world objects. In *Proc. of Inter. Joint Conf. on Artifical Intelligence (IJCAI)*, pages 1284–1290, Montreal, 1995.
[9] J. Orwant. Doppelganger goes to school: Machine learning for user modeling. Master's thesis, MIT, Media Laboratory, September 1993.
[10] J. Orwant. For want of a bit the user was lost: Cheap user modeling. *IBM Systems Journal*, 35(3), 1996.
[11] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.
[12] B. Schiele. *Object Recognition using Multidimensional Receptive Field Histograms.* PhD thesis, I.N.P.Grenoble, July 1997. English translation.
[13] B. Schiele and J. Crowley. Recognition wothout correspondence using multidimensional receptive field histograms. Technical Report 453, M.I.T. Media Laboratory, Perceptual Computing Section, December 1997.
[14] W. Schilit. *System architecture for context-aware mobile computing.* PhD thesis, Columbia University, 1995.
[15] C. Schmandt. *Voice Communication with Computers.* Van Nostrand Reinhold, New York, 1994.
[16] R. Sharma and J. Molineros. Computer vision-based augmented reality for guiding manual assembly. *Presence*, 6(3), 1997.
[17] T. Starner. Visual recognition of American Sign Language using hidden Markov models. Master's thesis, MIT, Media Laboratory, February 1995.
[18] T. Starner, D. Kirsch, and S. Assefa. The locust swarm: An environmentally-powered, networkless location and messaging system. Technical Report 431, MIT Media Lab, Perceptual Computing Group, April 1997. Presented ISWC'97.
[19] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence*, 6(4):386–398, Winter 1997.
[20] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer-based video. *IEEE Trans. Patt. Analy. and Mach. Intell.*, To appear 1998.
[21] R. Want and A. Hopper. Active badges and personal interactive computing objects. *IEEE Trans. on Consumer Electronics*, 38(1):10–20, Feb. 1992.
[22] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, September 1991.
[23] S. Young. *HTK: Hidden Markov Model Toolkit V1.5.* Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, 1993.