*In ACM International Conference on Intelligent User Interfaces (IUI '06), Sydney, Australia, Jan. 29-Feb. 1, 2006*

# Posture and Activity Silhouettes for Self-Reporting, Interruption Management, and Attentive Interfaces

Alejandro Jaimes

FXPAL Japan, Corporate Research Group, Fuji Xerox Co., Ltd., Japan

## ABSTRACT

In this paper we present a novel system for monitoring a computer user's posture and activities in front of the computer (e.g., reading, speaking on the phone, etc.) for self-reporting. In our system, a camera and a microphone are placed in front of a computer work area (e.g., on top of the computer screen). The system can be used as a component in an attentive interface, or for giving the user real time feedback on the goodness of his current posture, and generating summaries of postures and activities over a specified period of time (e.g., hours, days, months, etc.). All elements of the system are highly customizable: the user decides what "good" postures are, what alarms and interruptions are triggered, if any, and what activity and posture summaries are generated. We present novel algorithms for posture measurement (using geometric features of the user's silhouette), and activity classification (using machine learning). Finally, we present experiments that show the feasibility of our approach.

## Categories and Subject Descriptions

I.4.9 [Image Processing and Computer Vision]: Applications; H.5.2 [User Interfaces]: Ergonomics

## General Terms

Algorithms, Measurement, Human Factors

## Keywords

Ergonomics, Computer Vision, posture, ergonomics.

## 1. INTRODUCTION

Many people spend long periods of time in front of their computer and often suffer back, shoulder, and neck pains. Ergonomics [32] for computer users has therefore gained importance, on one hand because every year companies loose millions of dollars due to injuries sustained at the workplace by "information workers," and on the other hand because injuries and discomfort are so common. Problems occur because of many factors, such as *environment* (e.g., inadequate equipment or equipment arrangement), *activities* (e.g., typing for too long without a break, etc.), or *bad user habits* (e.g., inadequate posture, etc.). In fact, the importance of ergonomics in health and well-being is so great, that many guidelines exist for workers in many fields, even in the most unexpected occupations (e.g., [1]).



It is well known in the medical field that depression affects gait, posture, and of course, productivity. An individual that is not productive because he is depressed may sit in unhealthy postures, focus on the wrong activities, or limit the range of activities that he performs. One of the problems, however, is that most of us are not consciously aware of how much time we spend on different activities, and of our body postures while we work. At the same time, it is widely recognized that user attention is a limited resource, so computing devices should negotiate rather than impose the volume and timing of their communications with the user [34]. It would be extremely useful, therefore, to have a system that helps the user increase his own awareness of his habits and activities, and that lets the user decide how that information might be used in other contexts such as interruption management.

With this motivation in mind, we present a novel system for monitoring a computer user's posture and activities in front of the computer. In our system, a camera is placed on top of the computer screen and the computer user is monitored by the system as he works. The system uses the camera to measure the user's posture and determine his current activity (e.g., speaking on the phone, stretching, etc.). Feedback is given to the user, in real time, on the goodness of his upper body posture. In addition, input from the camera and a microphone are used to classify the worker's activities and give him summaries of what he has been doing for a determined period of time. The system can be used in the context of attentive interfaces [34] and for interruption control (e.g., switch off my e-mail and/or phone if I am reading) [18][7][3].

Our approach uses background subtraction to obtain silhouettes. From the silhouettes we extract geometric features to classify activities, and obtain vertical projections to separate head from torso and measure head and shoulder angles. We use input from a microphone to determine audio activity (someone speaking, silence, keyboard being used) to differentiate activities that are visually similar.

Three important aspects of our approach are flexibility, privacy, and simplicity. First, the goal is to give the user total control in defining good or bad postures and deciding when alarms and interruptions are triggered, if at all, and what activities should be included in the summary. Second, the system is meant for self-reporting: that is, posture and activity monitoring are private and not meant as a form of surveillance. Third, our method is computationally efficient and could be easily implemented in mobile personal devices (e.g., a personal posture system).

## 1.1 Related Work

Many commercial products exist to help computer users monitor their activities for the purpose of ergonomics (e.g., [37][38][39][40][41][42]), and studies have shown that ergonomic monitoring software can help improve computer worker productivity [16][17]. Some systems simply remind the user to take a Y-minute long break every X minutes (e.g., in [41] the user is forced to take a break by literally freezing the computer every X minutes according to the user's settings). Other systems include animations to guide user exercises during the breaks [38][39] and in some cases, the user's intensity and quantity of work is analyzed using data from input devices [39][37] (e.g., when a number of keystrokes or mouse movements have been performed [40]). Non-visual sensors have been used for posture detection [36], and keyboard monitoring [8], but we are not aware of any camera-based systems for ergonomics monitoring.

Although posture classification has been studied widely in Computer Vision, we are not aware of other works for the specific application we have constructed. Most approaches focus on classifying postures for surveillance applications or for applications with full-body view (e.g., standing vs. sitting vs. crouching, etc. [10]). The authors of [27] use a camera to detect posture using a fast algorithm that utilizes edge information. The system in [4] classifies postures in a vehicle (e.g., occupied by adult, by child, empty, occupant in-position, occupant out-of-position). The authors of [25] estimate 3D upper body posture using proposal maps, and the authors of [24] investigate "postural comfort zones" for hand gestures. A probabilistic framework for edge matching is presented in [12]. Other approaches include [6][20][30][28][4][14][29][26][15], and [9]. A review of related techniques for body tracking is given in [21].

Activity recognition is an active research area in computer vision and many approaches have been developed [21][31], even using wearable sensors (e.g., distinguish walking, running, etc.) [33]. Our work differs from previous computer user activity recognition approaches [19] in that we focus on posture monitoring, and self-reporting (of activities *and* posture). Unlike the authors of [19], we do not use a face detector because the problem constraints (i.e., user directly in front of the monitor) make finding the face in this application relatively easily and face detection algorithms are usually computationally expensive and sensitive to orientation changes. The authors of [31] present a boosting technique for activity recognition and apply it to computer-user actions such as talking on the phone, yawning, rubbing eyes, and others. The actions in [31] are more detailed, but the method is more computationally expensive. Other projects to summarize activities include MyLifeBits [13], which keeps records of e-mails, applications used, and so on. Finally, we are not aware of any attentive interfaces [34][7][18] that use posture or camera-captured user activity information. In [26] we discuss issues such as privacy and extensions to the framework.

## 1.2 Outline

The rest of the paper is organized as follows. In section 2 we define the problem and give details of the system. Section 3 describes self-reports and applications to interruption management. We present experiments in section 4 and conclude in section 5.

# 2. POSTURE AND ACTIVITY

## 2.1 Problem Definition

Our goal is to have a system that monitors the user's postures and activities in real-time. The results may be used for several purposes, including interruption management and self-reporting. In our system, posture is defined as the position of the user's upper body as he sits in front of the computer.

- *Posture:* since every user is different, the system cannot automatically determine what a good posture is using a single good-for-all measure. Therefore, the user must decide what are good (or comfortable) postures and give the system examples of those postures. As is done in practice, the user may consult a specialist (it is not uncommon for "ergonomic consultants" to evaluate one's workspace and posture and suggest improvements) before deciding which postures should be considered positive and which should be considered negative.
- *Activities:* each user should also determine which activities he is most interested in keeping track of. The particular positions in which activities are performed can vary widely from individual to individual. For example, one person may prefer to read documents when they are set on his desk, while another may prefer to hold them in his hands (variations for the same individual are also common). The system should produce a summary of the user's activities of interest over a specified period of time. In particular, we are interested only in activities that can be visually discriminated by an external observer (e.g., the camera). The goal of the system, therefore, is not to determine, for example, what application the user is working with, but rather focus on higher level activities such as sitting in front of the keyboard, reading, writing on a board, and so on.

Our goal is not to give absolute feedback on good or bad postures, activities of interest, or define when interruptions are suitable—it is entirely upto the user to define what his "good" postures are, his activities of interest, and when interruptions should take place. It is also not our goal to monitor activities for surveillance. The idea is for the user to utilize activity and posture information *for his own benefit*.

## 2.2 System Setup

The basic setup of our system consists of a microphone, and a camera on top of the computer monitor that captures a frontal view of the user (Figure 3). The algorithm proceeds as depicted in Figure 1.

The system contains five basic components: (1) initialization; (2) training; (3) setting of alarm and activity profiles; (4) monitoring; and (5) summarization. We describe each of the stages below:

- *Initialization:* an image of the background (without the user in the image) is stored when the application starts. This process must be repeated if the camera is moved or if there are significant lighting changes. The user initializes the system by pressing a button to capture the background.
- *Training:* the user sits in front of the computer and clicks on a button to indicate he is sitting in correct and incorrect postures. He also gives the system examples of "normal" activities he may be performing in front of the screen. For

example, speaking on the telephone, reading a paper, typing, stretching, taking a break, and so on. There are no pre-defined categories and the training stage is flexible (user decides how many examples he wants to provide—although the number of examples affects the performance of the system). The user can re-train the system at any time.

- *Alarm, activity & interruption profiles:* alarms can be set so that they activate only after certain periods of time or when certain postures occur. The user constructs a "summary and interruption profile" which determines what the summary should contain (e.g., I am only interested in summaries of good posture, or of X activities), and the time periods of the summaries (minutes, hours, days, months). The profile also includes information on when the user should or should not be interrupted.
- *Monitoring:* user can adjust a set of thresholds to modify the sensitivity of the system to his particular motions, switch the system on or off when desired, and view an image of his posture in a small window on the screen, or an indicator bar. In addition, the user can set the system on "privacy mode" so that only silhouettes are saved and not the actual photos.

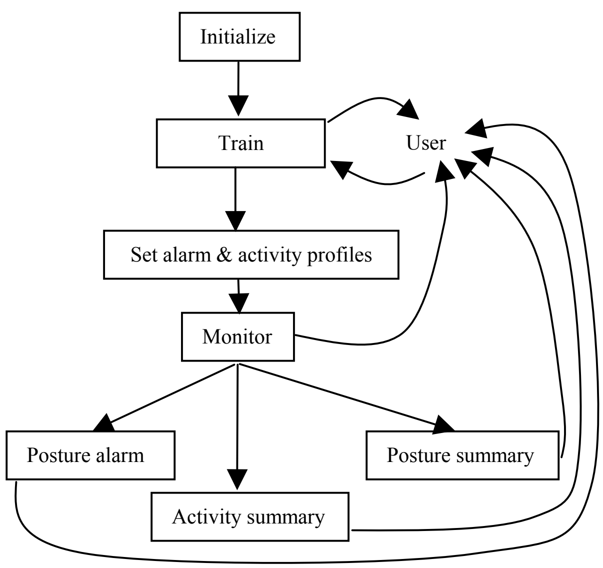


**Figure 1. Process overview. The user can retrain the system and modify his profiles at any time.**

## 2.3 Visual Processing

The background image obtained at initialization is used to perform background subtraction every *t* milliseconds (this depends on particular hardware used). This yields an image to which a threshold *th* is applied in order to obtain a binary image corresponding to foreground objects. The next step is to perform morphological operations (*erode, fill*, and *dilate*) on the binary image to eliminate holes and reduce noise (see silhouettes in Figure 7). The process may produce more than one region, so we label regions using a connected component algorithm. Since we assume that the only moving object is the user, we process only the largest region obtained. Except in rare cases (e.g., very similar background and foreground pixels), the user will yield a single region, as long as lighting conditions are constant and an appropriate threshold is selected (see Figure 5 and Figure 6). Once the largest region has been selected, the system extracts the following features:

1. For the region, extract bounding box width, bounding box length, bounding box x and y location, center of mass, perimeter, area, angle of main axis, length of primary axis, length of secondary axis, Feret's diameter, and eccentricity.
2. Draw *n* lines that originate at the center of mass, separated by equal angle increments (e.g., for an angle of 45º we obtain 8 lines). For each line, find the external boundaries of the region (see lines in Figure 3). These points define a polygon used for activity classification.

All of the extracted features are concatenated to obtain an n-dimensional feature vector $fv=\{f_1, f_2, \ldots, f_n\}$ for each activity. For instance, if we use 16 lines from the center of mass, we obtain 29 features (16 line lengths plus 13 features described).

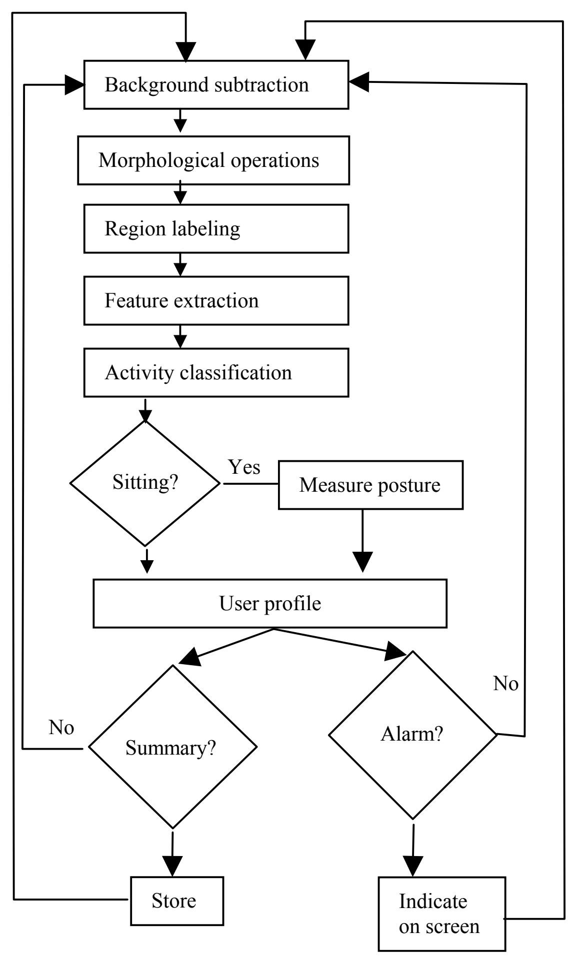


**Figure 2. Algorithm outline.**

## 2.4 Training and Classification

The user trains the system (Figure 1) by showing examples of good and bad postures and by showing examples of his common activities. During training, the user simply clicks a button on the interface to indicate that a given posture is "good" or "bad". For activities, the user labels examples of each of the activities (e.g., on the phone, reading, etc.). This process yields several sets of training examples, one for each class (e.g., on the phone, stretching, sitting only, etc.). The feature vectors are then used by a Machine Learning algorithm (e.g., Nearest Neighbor) to learn an *n*-class classifier (e.g., reading, on the phone, etc.).

In the monitoring stage, we extract the same features and use the classifier that was learned to determine which activity is being performed by the user. The classification results are used for the summary, or for interruption management. Then for the *sitting* activity we measure posture as described next.

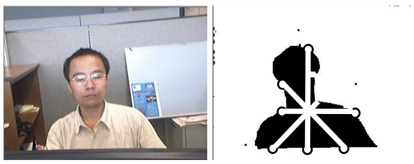

**Figure 3. The endpoints of the lines define a polygon used for activity classification.**

## 2.5 Posture Measurement

In Figure 4 the first posture (top left) can be considered a "correct" posture (according to the user) since he is sitting straight while he looks at the screen. The other postures deviate from the "good posture" by a measurable quantity, in particular, by the angle of the head, and the angles of the shoulders. We compute this as follows:

1. Obtain vertical projection profile for the extracted region (area to the left of the silhouette in Figure 5).
2. Find the deepest valley of the vertical profile and use the location of the valley to separate the head from the torso in the image of the extracted region (horizontal line in Figure 5).
3. Fit a diamond to the head (Figure 5, right; see [2]).
4. Using the line that divides the head and the torso, search for shoulder edge pixels below in perpendicular direction. Once a number of edge pixels is found, fit a line to each of the shoulder edges using linear regression (see lines and diamond in right, Figure 5).

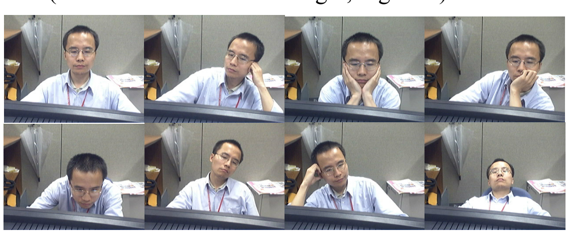

**Figure 4. Different user postures.**

The angle of the head and the angles of the shoulders are used to measure posture.

## 2.6 Audio Processing

We use a microphone to detect when there is voice, when there is silence, or when there is typing on the keyboard. We implement an audio classifier for these three classes using simple features such as volume, mean pitch, pitch standard deviation, and pitch intensity (using the method described in [22]). Since there are pauses when a person speaks, a voice segment will often include silence gaps. Therefore, we use constraints on the amount of time a voice is heard (e.g., a phone conversation must last at least several seconds). The results are combined with the visual activity classification results to disambiguate activities that are visually similar.

Many complex methods exist to classify audio signals, but it is not necessary to apply them here since the accuracy constraints are low (we do not need millisecond accuracy). Keyboard activity can also be detected using software, but since in our framework the audio signal is processed to detect speech anyway, it is reasonable to distinguish the sounds of the keyboard from other sounds.

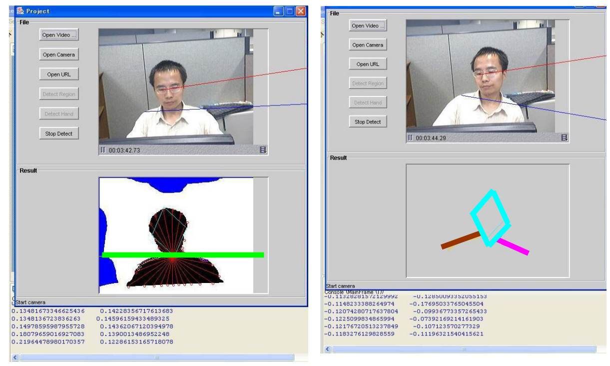

**Figure 5. Extraction of head, and shoulder angles.**

## 3. INTERRUPTIONS, ATTENTIVE INTERFACES, & SELF-REPORTS

In the author's experience, one of the biggest problems with most of the ergonomic monitoring systems available is that they interrupt the user without considering his current activity, often causing annoyance rather than relief. One of the goals of our system, therefore, is to give the user unobtrusive feedback. This is achieved by showing, on screen, small indicators of his posture in real time. A small icon on the bottom of the screen can show the user his head and body angles, providing immediate feedback on his posture (e.g., is his head straight? are his shoulders straight?). Another possibility is to use an indicator bar: when the bar is in the green area the user's posture is OK (according to his "good posture" examples). If the bar is on the red areas it indicates that the user is not in good posture (e.g., leaning right or left).

As explained in earlier sections, the user sets up his activity and interruption profiles. Therefore, it would be possible to connect the output of the system with other applications (e.g, e-mail client, phone, etc.) for interruption management and to use the system in the context of attentive interfaces. Gaze is an important indicator of attention and several methods exist (see [21] for a review) which could be implemented in our system. However, they would make the approach computationally expensive and gaze may not be needed for distinguishing most high-level activities.

One of the main purposes of the system is increasing the user's awareness of his own postures and work habits. Since it can be set to record activities continuously, activity and posture summaries can be generated for any period of time (e.g., a day, a week, the last hour, etc.), only for postures or activities the user is interested in (according to his profile). Even if only the silhouettes are recorded, it is interesting to look back at the results by simply browsing through the images particularly because they give the user visual feedback of his posture. One interesting issue, therefore, is how to best present the information collected (e.g., a sped up animation of his posture over a period of time, a table, etc.).

As with all personal data, it is important to consider privacy issues. One way to help the user feel more comfortable is to give him full control of what the application does. For instance, not saving any images and showing only real-time information, saving only silhouettes, saving only posture information, and so on. Another way is to ensure that the data is secure by encrypting it and saving it in a separate personal store (e.g., a portable posture measuring device).

# 4. EXPERIMENTS

We have implemented a prototype system in Java using ImageJ [2] and Weka [35]. The current implementation, with a standard USB webcam (320x240) runs at 15 frames per second.

## 4.1 Experiment One

In the first experiment we randomly selected 8 "sitting in front of the computer" postures (by one user) and compared the posture angles by manually drawing the corresponding lines on the image and computing the difference with the lines obtained automatically. As Table 1 shows, the average error is about 6.5 degrees. In Figure 5 the lines on the top image of are drawn manually, the horizontal line shows the automatic separation of head and torso, and the diamond on the bottom right is obtained automatically, as are the shoulder lines (bottom right image Figure 5).

**Table 1. Comparison of automatic (A) and manual (M) head and shoulder angles.**

| Head angle (degrees) | | | Shoulder angle (degrees) | | |
|---|---|---|---|---|---|
| **A** | **M** | **Error** | **A** | **M** | **Error** |
| 8.60 | 8 | 0.60 | 8.03 | 2 | 6.03 |
| 30.96 | 22 | 8.96 | 15.48 | 10 | 5.48 |
| -6.88 | 8 | 14.88 | -5.73 | -10 | 4.27 |
| 13.18 | 12 | 1.18 | 32.10 | 12 | 20.10 |
| 34.39 | 12 | 22.39 | 19.49 | 13 | 6.49 |
| 9.17 | 8 | 1.17 | 6.31 | 4 | 2.31 |
| -0.57 | 2 | 2.57 | 0 | -2 | 2 |
| -19.49 | -20 | 0.51 | 12.04 | 8 | 4.04 |
| **Average** | | 6.53 | | | 6.34 |

We found similar performance for different users, and in general we found that head-body separation is not problematic when the user's hands are lowered. Difficulties arise when the hands are lifted or when the user performs other activities such as speaking on the telephone (see Figure 6(g)). However, head-torso separation is only of interest in the sitting case when none of the other activities are being performed. Nonetheless, skin detection could be used to deal with more difficult cases (e.g., long hair).

The current angle measures give a very general indication of the user's posture, and additional measures could be used. For example, the size of the fitted diamond could give a rough estimate of the 3D orientation of the head and additionally lines could be fitted to the sections of arms that are visible (below the shoulder). Adding more features, however, raises the question of how to convey the information to the user in a simple, effective way (e.g., 3D animation?).

## 4.2 Experiment Two

For this experiment one person sat in front of the computer and performed the following activities: sit, read, write, speak on the phone, stretch, and others. The process described in section 2 was applied, but in this experiment we only used the lengths of each of the lines (Figure 3; experiment 3 uses all the features).

We obtained 100 samples (20 for read, 20 for sit, and 15 for each of the other classes). The results of automatic classification (using 10-fold cross-validation) are summarized in Table 2. Although the training set is small, the results are promising. For the classifier with highest accuracy (IB1), the class with highest precision is "reading". The "resting" class corresponds to the activity in which the user holds his hands on the back of his head (like stretching). Not surprisingly, this class yields the highest recall as the silhouette is most different from the rest. As expected, the most difficult class is the "phone" class as it is similar to the sitting class.

In general, as expected, we found that using more lines to represent the polygon gives higher accuracy. The average accuracy increased by about 4% when doubling the number of vertices, with the best results shown in Table 2—we used 32 lines in all experiments.

**Table 2. Results (%) of automatic classification, using 1-nearest neighbor (IB1), 3-nearest neighbor (IB-3), and Naïve-Bayes classifiers (NB). Precision (P) and Recall (R) values are shown.**

| **Postures** | **IB1** | | **IB3** | | **NB** | |
|---|---|---|---|---|---|---|
| | P | R | P | R | P | R |
| **Sit** | 78 | 90 | 55 | 80 | 64 | 70 |
| **Read** | 83 | 75 | 75 | 45 | 71 | 75 |
| **Write** | 74 | 93 | 77 | 87 | 92 | 80 |
| **Phone** | 71 | 80 | 60 | 80 | 50 | 40 |
| **Resting** | 79 | 100 | 71 | 100 | 100 | 100 |
| **Others** | 75 | 20 | 100 | 70 | 77 | 87 |
| **Accuracy** | 77% | | 66% | | 75% | |

Some representative examples for this experiment are shown in Figure 6. The system succeeded in cases (a) through (d), and failed in the other cases. Case (e) is interesting because it shows one of the limitations of using only the lines: including additional features (e.g., silhouette bounding box) eliminates this error. Note that the system does not fail in (c) because a line is projected along the arm. Case (f) is more difficult and shows the limitations of a silhouette since information within the silhouette is not used. The errors in cases (g) and (h) could be easily eliminated incorporating the results of the audio analysis.

Results with other users were similar in terms of overall performance, but we found interesting variations. For example, the algorithm did well on the phone class (and reading) with some users, but very poorly with others. This is logical since different people have different habits and some people's activity silhouettes are more easily distinguishable from others (e.g., my "reading" vs. "phone" silhouette may be very different to each other, but someone else's same activities may look very similar).

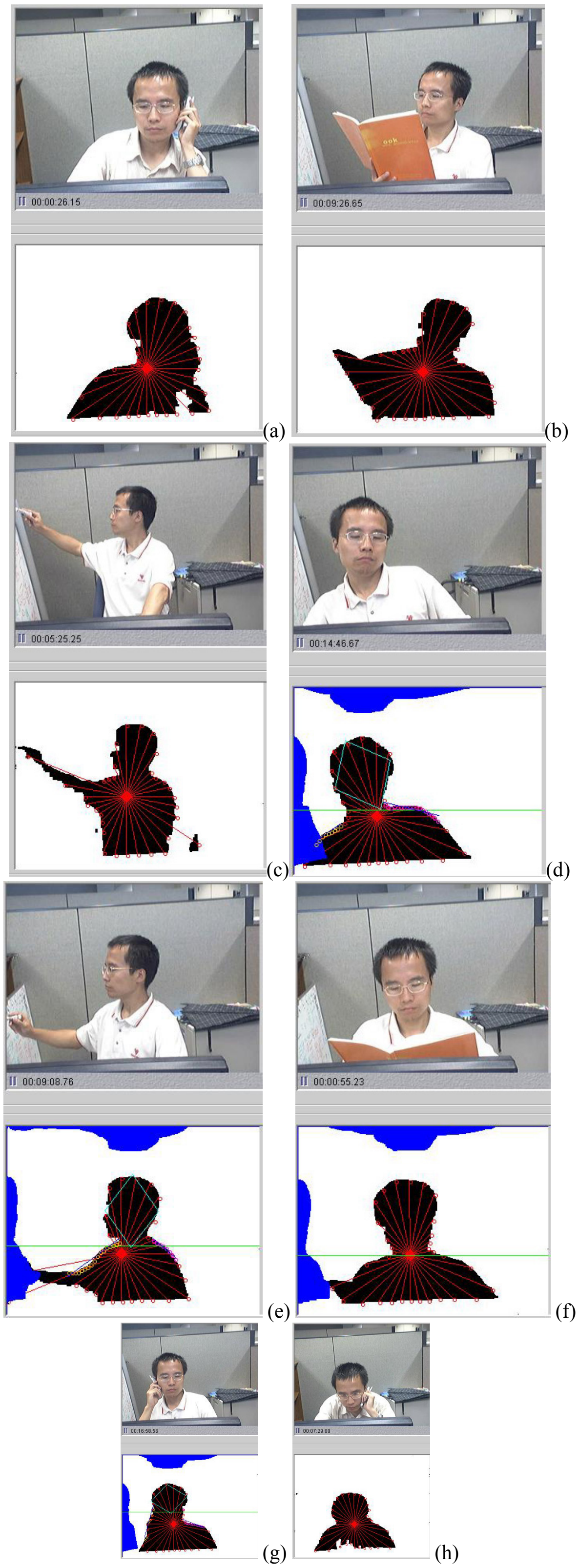


**Figure 6. Classification of different activities.**

## 4.3 Experiment Three

In the third experiment four subjects performed the following activities: speak on the phone, stretch, read, converse with a colleague standing next to the desk, and sit (e.g., as in typing—see silhouettes in Figure 7). We obtained a total of 64 examples and constructed a binary classifier (sitting only and others) and a 5-class classifier (for each of the five types of activities). The results of automatic classification (using 10-fold cross-validation) are summarized in Tables 3 (binary classifier) and 4 (five-class classifier). As expected, performance of the binary classifier is, overall, slightly higher than for the n-class classifier. The performance of other algorithms such as NN3 and Naïve Bayes is comparable to the results presented. One disadvantage of Nearest Neighbor approaches is the slow speed at classification time. Thus if the number of training examples is large, NN approaches should be avoided.

**Table 3. Results of automatic binary classification (in % values) for data of four people, using 1-nearest neighbor (NN1), multilayer perceptron (MLP), and SVM classifiers. Precision (P) and Recall (R) values are shown.**

| | NN1 | | MLP | | SVM | |
|---|---|---|---|---|---|---|
| **Activity** | P | R | P | R | P | R |
| **Sit** | 63.3 | 86.4 | 72.0 | 81.8 | 80.0 | 72.7 |
| **Others** | 91.2 | 73.8 | 89.7 | 83.3 | 86.4 | 90.5 |
| **Accuracy** | 78.1 % | | 82.8 % | | 84.4 % | |

Classification using 66% of the samples for training and the rest for testing, for the two and five class problems (using NN1), gives 90.1% accuracy (correct classifications) in both cases. The errors are misclassifications between sit and phone.

**Table 4. Results of automatic classification (in % values), using 1-nearest neighbor (NN1), multilayer perceptron (MLP), and SVM. Precision (P) and Recall (R) values are shown.**

| | NN1 | | MLP | | SVM | |
|---|---|---|---|---|---|---|
| **Activity** | P | R | P | R | P | R |
| **Sit** | 63.3 | 86.4 | 78.3 | 81.8 | 58.8 | 90.9 |
| **Conversation** | 90.0 | 100 | 90.0 | 100 | 90.0 | 100 |
| **Read** | 83.3 | 45.5 | 66.7 | 54.5 | 100 | 36.4 |
| **Rest** | 100 | 100 | 100 | 100 | 100 | 100 |
| **Phone call** | 75.0 | 50.0 | 75.0 | 75.0 | 66.7 | 33.3 |
| **Accuracy** | 76.6% | | 81.3% | | 73.4% | |

As noted earlier, the performance may vary from individual to individual. The results of classifying activities from different individuals, however, gives us insight into the robustness of the approach to variations. A single user's activity silhouettes may vary slightly depending on how close he sits to the camera, the clothing he is wearing (e.g., a heavy jacket may have some impact on the silhouette) or simply his position. Thus, the approach must be flexible enough to recognize such variations, but robust enough to make distinctions between activities (and postures).

## 4.4 Experiment Four

In the fourth experiment, we tested an audio classifier built using data from one person. The training set consisted of 30 seconds of keyboard input, 30 seconds of silence, and one minute of voice (speaking on the phone). We extracted the features described in section 2.6, namely volume, mean pitch, pitch standard deviation, and pitch intensity, as implemented in [22]. We used the MAD framework in Matlab to extract pitch using autocorrelation using frames of length 1024 (32 kHz sampling rate) and one second segments.

Using 10-fold cross-validation we obtained, for a 1-nearerst neighbor classifier, 95% accuracy (keyboard precision 93% and recall 89%; voice precision 94% and recall 95%; and 100% precision and recall for silence). Although we did not combine the results of the audio and visual classifiers, it is clear that combining the results would have a positive impact on performance.

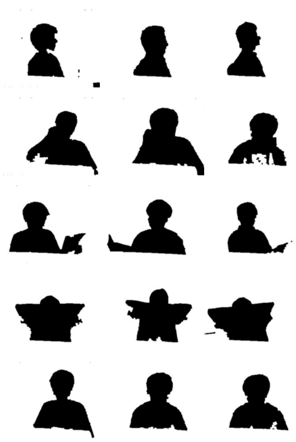

**Figure 7. Example silhouettes obtained from experiment 3 for various activities by several individuals (from top to bottom: converse with a colleague, speak on the phone, read, stretch, and sit).**

## 4.5 Discussion

The experiments show reasonable results for the activities chosen using very small training sets. In the current version of the software, the user decides in advance which activities he is interested in. Another way to use the data, however, is to perform automatic clustering of the activities and postures seen so far by the system up to a certain time, and displaying the results to the user so he can select the postures and activities that he is interested in. This approach may work better in practice as the user can "discover" his natural postures instead of posing to collect examples.

As with all learning-based techniques, two important issues are the number of training examples and the quality of the training set. If the system collects just one image per second, that yields 3,600 images per hour, or around 28,800 images in an 8 hour period. With today's storage capabilities, that is not a lot of space for images of size 320x240, particularly if only the silhouettes are saved in a highly compressed format. Clustering the images and showing them to the user for labeling could work extremely well, but a higher burden would be placed on the user and it would be necessary to build an interface to effectively and efficiently view and modify the clusters if needed.

# 5. CONCLUSIONS & FUTURE WORK

We have presented a novel system for monitoring a computer user's upper body posture, and activities in front of the computer (e.g., reading, speaking on the phone, etc.) for self-reporting. Three important aspects of our approach are flexibility, privacy, and simplicity. The system allows the user to decide what his good and bad postures are, as well as to select which activities should be monitored. The approach is based on silhouettes and it is therefore computationally efficient and easy to implement on inexpensive hardware.

Our experiments show promising results. However, accuracy could be improved by using more features, and further user testing is necessary. Some extensions to the framework include estimating 3D pose, incorporating additional monitoring functionalities (e.g., keyboard use), and using more cameras and other types of sensors. It would also be interesting to infer the user's mental state using posture, using approaches similar to those in [7], and perhaps even using the silhouettes for user identification (using more features).

# ACKNOWLEDGEMENTS

I would like to thank Jianyi Liu for his initial work on this project, and my colleagues at Fuji Xerox for participating in the experiments.

# 6. REFERENCES

[1] *A Guide to Occupational Health and Safety in the New Zealand* Sex *Industry*. Occupational Safety and Health Serv., Dept. of Labor, Wellington, New Zealand, June 2004.

[2] M.D. Abramoff, P.J. Magelhaes, and S.J. Ram, S.J. "Image Processing with ImageJ," *Biophotonics International*, volume 11, issue 7, pp. 36-42, 2004.

[3] P.D.. Adamczyk and B.P. Bailey, "A Method and System for Intelligent Interruption Management," in proc. *4th International Workshop on TAsk MOdels and DIAgrams for user interface design For Work and Beyond*, Gdansk, Poland, September 26-27, 2005.

[4] A. Adamek, N.E. O'Connor, and G. Jones, "An Integrated Approach for Object Shape Registration and Modeling," in *2005 Intl. Workshop on Multimedia Info. Retrieval in conjunction with SIGIR 2005*, Rio de Janeiro, Brazil, 2005.

[5] O. Basir, D. Bullock, and E. Breza, "Visual classification and posture estimation of multiple vehicle occupants," *US Patent #20040220705*, Nov. 4, 2004.

[6] B. Boulay, F. Bremond, and M. Thonnat, "Human Posture Recognition in Video Sequence," in pro. *Joint IEEE International Workshop on VS-PETS*, pp.23-29, Nice, France, Oct. 11-12, 2003.

[7] D. Chen, and R. Vertegaal, "Using Mental Load for Managing Interruptions in Psychologically Attentive User Interfaces," *ACM CHI 2004*, Montreal, Canada, April 2004.

[8] M.-S. Chuang, and M.-S. Huang, "System and method for detecting unhealthy operation posture of using keyboard," *US Patent #20030151595*, Feb. 7, 2002.

[9] M.D. Cordea, E.M. Petriu, N.D. Georganas, D.C.Petriu, and T.E. Whalen, "Real-time 2(1/2)-D Head Pose Recovery for Model-Based Video Coding," *IEEE Trans.*

*on Instrument. and Measurement*, Vol. 50, No. 4, August 2001.

[10] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Probabilistic Posture Classification for Human-Behavior Analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 35, No.1, January 2005.

[11] R. Donkin, "Typing Injury FAQ: Software Monitoring Tools," (http://www.faqs.org/faqs/typing-injury-faq/software), 1995, accessed Aug. 20. 2005.

[12] A. Elgammal, V. Shet, Y. Yacoob, L. S. Davis, "Gesture Recognition using a Probabilistic Framework for Pose Matching," *The 7th Int. Conf. on Control, Automation, Robotics and Vision, ICARCV*, Singapore in Dec. 2-5, 2002.

[13] J. Gemmell, R. Lueder, and B. Gordon, "The MyLifeBits Lifetime Store," *ACM SIGMM 2003 Workshop on Experiential Telepresence (ETP 2003)*, November 7, 2003, Berkeley, CA.

[14] S.R. Gunn and M.S. Nixon, "Snake Head Boundary Extraction using Global and Local Energy Minimisation," in proc. *IEEE Int. Conf. on Pattern Recognition*, pp. 581-585. Vienna, 1996.

[15] I. Haritaoglu, D. Harwood, and L. Davis, "Ghost: A Human Body Part Labeling System Using Silhouettes," in *Proc. Intl. Conf. On Pattern Recognition*, Vol. 1, pp. 77-82, 1998.

[16] A. Hedge, and E.J. Ray, "Effects of an electronic height-adjustable worksurface on self-assessed musculoskeletal discomfort and productivity among computer workers," proceedings of the *Human Factors and Ergonomics Society 48th Annual Meeting*, New Orleans, Sept. 20-24, HFES, Santa Monica, 1091-1095, 2004.

[17] A. Hedge, "Effects of Ergonomic Management Software on Employee Performance," *Cornell Human Factors Laboratory Technical Report* #/RP9991, 1991.

[18] E. Horvitz, P. Koch, and J. Apacible, "BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption," in *CSCW 2004*, Chicago, IL, Nov. 2004.

[19] H. Ikeda, M. Maeda, N. Kato, and H. Kashimura, "Classification of Human Actions Using Face and Hands Detection," in proc. *ACM Multimedia 2004*, New York City, Oct. 2004.

[20] S. Illic, M. Salzmann, and P. Fua, "Implicit Surfaces Make for Better Silhouettes," *Computer Vision and Pattern Recognition*, (1) 2005: 1135-1141. San Diego, CA, June 2005.

[21] A. Jaimes and N. Sebe, "Multimodal Human Computer Interaction: A Survey," *IEEE Int. Workshop on Human-Computer Interaction at IEEE Int. Conf. on Computer Vision (ICCV 2005)*, Beijing, China, Oct. 15-21, 2005.

[22] A. Jaimes, J. Liu, and N. Sebe, "Affective Meeting Video Analysis," in proc. *IEEE ICME 2005*, Amsterdam, The Netherlands, July 2005.

[23] A. Jaimes, and J. Liu, "Sit Straight (and tell me what I did today): A Human Posture Alarm and Activity Summarization System," in *ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE 05) in conjunction with ACM MM 2005*, Singapore, 2005.

[24] M. Kölsch, A. C. B., and M. Turk, "Postural Comfort Zone for Reaching Gestures," in *HFES Annual Meeting Notes*, October 2003.

[25] M.W. Lee and I. Cohen, "Human Upper Body Pose Estimation in Static Images, " in proc. *ECCV 2004*, Vol II, pp. 126–138, 2004.

[26] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju, "Finding Optimal Views for 3D Face Shape Modeling," in proc. *International Conference on Automatic Face and Gesture Recognition*, pp. 31-36, Seoul, Korea, May 2004.

[27] H. Nomura and T. Shima, "Image feature extraction apparatus, method of extracting image characteristic, monitoring and inspection system, exposure system, and interface system," *US Patent #20020015526*, Feb. 7, 2002.

[28] R. Rosales and S. Sclaroff, "Inferring Body Pose Without Tracking Body Parts," in *IEEE Conf. on Comp. Vision and Pattern Recognition*, Hilton Head Island, USA, June 2000.

[29] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast Pose Estimation with Parameter-Sensitive Hashing," *IEEE International Conference on Computer Vision (ICCV)*, Vol. 2, pp. 750-757, October 2003

[30] C. Sminchisescu and A. Telea, "Human Pose Estimation From Silhouettes A Consistent Approach Using Distance Level Sets," *WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, 2002.

[31] P. Smith, N.d.V. Lobo, and M. Shah, "Temporal Boost for Event Recognition," in proc. *ICCV 2005*, Beijing, China, Oct. 2005.

[32] N. Stanton, A. Hedge, K. Brookhuis, E. Salas, and H.W. Hendrick. *Handbook of Human Factors and Ergonomics Methods*. CRC Press, 2004.

[33] T. Suzuki and D. Miwako, "Life support apparatus and method and method for providing advertisement information," *US Patent #20010049471*, Dec. 6, 2001.

[34] R. Vertegaal (ed.), "Attentive User Interfaces" special issue, *Communications of the ACM*, Vol. 46, No. 3, March, 2003.

[35] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[36] Y. Zheng, "Method and apparatus for sensing body gesture, posture and movement," *US Patent # 20040024312*, Feb. 5, 2004.

[37] Ergonomix (http://www.publicspace.net/ergonomix/)

[38] Stretch Break (http://www.paratec.com/)

[39] RSI guard (http://www.rsiguard.com/)

[40] Ergotimer (http://www.tropsoft.com/ergotimer/ergonomics.htm)

[41] Break Time (http://kadmi.com/products.html)

[42] Protector of Health (http://www.olympsoft.com/)