Sit Straight (and tell me what I did today): A Human Posture Alarm and Activity Summarization System

Alejandro Jaimes and Jianyi Liu
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 monitors the computer user's postures and summarizes his or her activities. The system gives the user real time feedback on the goodness of his current posture, triggers alarms if the postures are not good postures, and generates 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 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, and discuss privacy issues and applications of the techniques presented (health monitoring, productivity analysis, and others).

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

Recently there has been a strong interest in recording one's personal activities for future retrieval using cameras, microphones, and other sensors. Some initiatives have focused on wearable or portable devices, and others on using specialized rooms (e.g., smart meeting rooms).

Since computer users spend long periods of time in front of a computer, there have also been some initiatives to keep track of what users do on the computer. The MyLifeBits project [14], for example, keeps records of e-mails, applications used, and so on. It would certainly be useful to know, however, not only what applications are being used, but also what the user is doing in his workspace [19] (e.g., reading, speaking on the phone, etc.). Such information can be used for self-reporting and self-monitoring: on one hand, it can help the user manage his time more accurately, and on the other hand it can help him monitor his own performance or progress.

An important area that has not been explored in this context, however, is that of Ergonomics [32]. Although Ergonomics ("an applied science concerned with designing and arranging things people use so that the people and things interact most efficiently and safely" [45]) is applicable in many scenarios, it has gained importance for computer users because injuries due to prolonged computer usage are not uncommon. In fact, every year companies loose millions of dollars due to injuries sustained at the workplace by "information workers." Such injuries can occur because of many factors, such as the environment (e.g., inadequate equipment or equipment arrangement), the activities performed (e.g., typing for too long without a break, etc.), or simply bad user habits (e.g., inadequate posture, etc.).

Posture and productivity are tightly linked. It is well known in the medical field that depression affects gait, posture, and of course, productivity. An individual that is not productive may sit in unhealthy postures, focus on the wrong activities, or limit the range of activities that he performs. The importance of the impact of Ergonomics in productivity (including posture) is so great, that many guidelines exist for workers in many fields, even in the most unexpected occupations (e.g., [1]). Furthermore, studies have shown that ergonomic monitoring software *can* help improve computer worker productivity [18].

There is no doubt that it would be useful to have a system that allows the user to self-report his activities and monitor his own posture in front of the computer. In this context, the goal of our work goes beyond recording for retrieval, to monitoring the user's activities and providing unobtrusive, real-time feedback to help him (or her) improve his work habits.

We present a novel posture alarm and activity summarization system. 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

CARPE'05, November 11, 2005, Singapore.

activities and give him summaries of what he has been doing for a determined period of time.

The proposed system performs the following functions:

- *Posture indicator and alarm:* the system monitors the user's posture, and using an indicator on the screen, shows the user, in real time, how good (or bad) his posture is. The user may set alarms that alert him when he has been sitting in a particular (e.g., unhealthy) posture for a long time (as defined by the user).
- *Posture summary:* the system produces, for a user-determined period of time, a summary of the user's postures.
- *Activity summary:* the system produces, for a user-determined period of time, a summary of activities (e.g., typing, reading, etc.) performed in front of the desktop.

It is important to emphasize two aspects of the system: (1) flexibility, and (2) privacy. First, the goal is to give the user total control in defining good or bad postures and deciding when alarms 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 (this is discussed further in subsequent sections).

Our approach uses background subtraction to extract silhouettes. From the silhouettes we obtain vertical projections to separate head from torso, and extract geometric features to classify activities. Posture is measured by obtaining head and shoulder angles. We use input from a microphone to determine when someone is speaking, when there is silence, or when the keyboard is being used. Using the audio we can differentiate activities that are visually similar. Although many posture algorithms have been developed, this is the first camera-based system we are aware of for posture monitoring.

1.1 Related Work

Many commercial products exist to help computer users monitor their activities for the purpose of Ergonomics (e.g., [36][37][38][39][40][41]), and studies (e.g, [18]) have shown that ergonomic monitoring software can help improve computer worker productivity. Some of the systems monitor keyboard and mouse use, while others simply remind the user to take a Y-minute long break every X minutes. The system in [40], for example, forces the user to take a break by literally freezing the computer every X minutes according to the user's settings. The system in [36] monitors keyboard and mouse use and suggests when the user should take micro-breaks. The Stretch Break system [37] also reminds users to take breaks, but in addition it shows animations so that users can do stretch exercises guided by the computer. RSI Guard [38] monitors mouse and keyboard use, and utilizes animations to encourage stretching, after analyzing the user's intensity and quantity of work using data from the two input devices. Ergotimer [39] suggests breaks when a time limit is reached or when a number of keystrokes or mouse movements have been performed. We are not aware of any camera-based systems for ergonomics monitoring. Other systems (e.g., [35]) use sensors for posture detection, but no cameras (monitoring "bad" keyboard using sensors [8]; "postural comfort zone" for hand gestures in [24]).

Although posture classification has been studied widely in the Computer Vision community, 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.). The authors of [27] use a camera system 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. The system in [11] classifies postures such as standing, sitting, laying and crouching. A probabilistic framework for edge matching is used by [13]. Other approaches include [7],[20],[31],[28],[4],[15],[30],[26],[16], and [9]. A review of related techniques for body tracking is given in [22].

Wearable cameras and sensors have been used to recognize activities [33] (e.g., walking, running, etc.). The system in [19] uses a camera to classify video scenes according to user tasks. Our work is similar to [19], whose authors *only* classify a computer user's activities. However, our system focuses on posture monitoring, and the generation of self-reports (for activities *and* posture). Unlike the authors of [19], we do not use a face detector for activity classification. One reason for this is that due to the natural problem constraints (i.e., user directly in front of the monitor), finding the face in this application is not very challenging. In addition, although many techniques have been developed for face detection, in general they are more computationally expensive and sensitive to orientation changes (e.,g, non-frontal faces) than the techniques we present.

1.2 Outline

The rest of the paper is organized as follows. In section 2 we define the problem we are trying to solve and give an overview of the system. In section 3 we describe our technique for activity detection and posture monitoring. Section 4 describes real-time feedback and summaries in the application. In section 5 we present experiments, and discuss applications and other issues in section 6. We conclude in section 7.

2. SYSTEM OVERVIEW

2.1 Problem Definition

The problem we are trying to solve is two-fold. On one hand, the idea is to have a system that alerts the user when his posture is not suitable. By posture we mean the position of his upper body as he sits in front of the computer. On the other hand, the goal is for the system to produce a summary of the user's activities in his workspace.

- *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. The goal is for the system to give the user unobtrusive, real time feedback based on his posture preferences, and produce a summary of his postures for a specified time period. Alarms are set by the user so they are unobtrusive.

- *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 or activities of interest—it is entirely upto the user to define what his "good" postures are and his activities of interest. 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*. We recognize, however, that privacy issues must be addressed, and discuss these (as well as "group" activity monitoring) in later sections.

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 1).

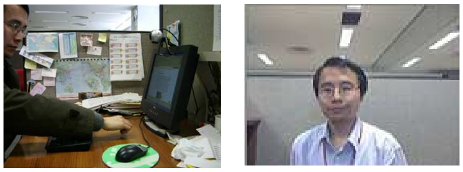


Figure 1. Basic system setup. The camera is placed on top of the computer screen or on another location facing the user. A microphone is also placed near the screen to capture voice activity.

The algorithm proceeds as depicted in Figure 2. 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. This process must be repeated if the camera is moved or if there are significant lighting changes[1]. The user initializes the system by pressing a button to capture the background.
- *Training:* the user sits in a comfortable position with correct posture. Then the user clicks on a button to indicate that it is his standard correct posture (or postures). He may also give the system negative examples of postures that are not desired, and 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 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 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).
- *Monitoring:* user can adjust a set of thresholds to modify the sensitivity of the system to his particular motions, switch the monitor on or off when desired, and view an image of his posture in a small window on the screen, an indicator bar, or other indicators of good or bad posture. In addition, the user can set the system on "privacy mode" so that only silhouettes of his image are saved and not the actual photos.

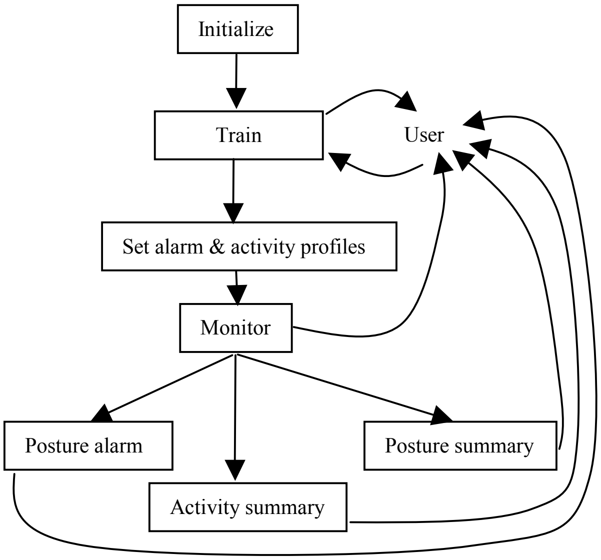


Figure 2. Process overview. The user can retrain the system and modify his profiles at any time.

The system is highly flexible. For example, a researcher initializes the system by showing examples of his activities. He also sets up an "alarm profile" that will trigger when particular activities occur for determined periods of time. For example, he can set the profile so that an alarm is triggered if he is typing for more than *t* minutes (e.g., 60 minutes), or if he is on the phone for too long. One of the criterion, therefore, is the time spent on each activity, so this way an "activity profile" is created. He can also adjust the system at any time by giving more positive or negative examples.

[1] In the current version we do not compensate for lighting changes, but this can be easily improved if the changes are gradual (e.g., if the desk is near a window lighting changes can occur because of changes in weather or time of day).

2.3 Visual Processing Overview

The system is based on a background subtraction algorithm (see overview in Figure 3). 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.

Shadows can sometimes be problematic, particularly if the user sits close to the background (e.g., a wall) because the user's movement may cause lighting changes in the scene. In order to increase robustness, we use a rule-based skin detector [21]. Pixels that correspond to skin and are different from background pixels according to a second (lower) threshold th_2 are also included in the binary image. We experimentally found that adding this constraint improves the separation of the user from the background, and it does not affect the detection of other motion areas (e.g., body covered by clothing). Figure 6 shows the results without the skin filter. The images in Figure 12 show the results using the skin filter—notice the improvement around the eyes (see actual silhouettes in Figure 13).

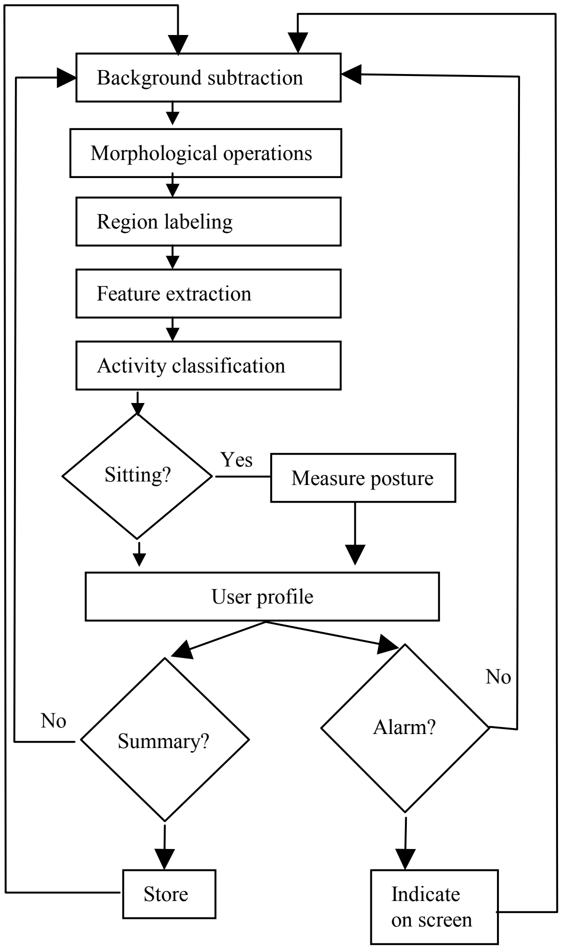


Figure 3. Algorithm outline.

The next step is to perform morphological operations (*erode, fill*, and *dilate*) on the binary image to eliminate holes and reduce noise (see region in Figure 6 *before* the skin filter and morphological operations; see actual silhouettes in Figure 13). The process may produce more than one region, so we label regions using a connected component algorithm. Since we assume that, for the most part, the only moving object will be the user, we proceed to process only the largest region obtained, which corresponds roughly to the silhouette of the user. 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 11 and Figure 12).

Next we extract the region's bounding box and the following region features: bounding box width, bounding box length, bounding box x and y location, center of mass of region, perimeter of region, region area, angle of primary axis of region, length of primary axis of region, length of secondary axis of region, Feret's diameter (the greatest distance possible between any two points along the boundary of the region), and region eccentricity. The features are used by a learning algorithm during training, and by a classifier during monitoring for activity classification. If the current activity is determined to be "sitting in front of the computer", we extract additional visual features to measure posture (section 3).

2.4 Audio Processing

We use a microphone to detect when there is voice, when there is silence, or when there is typing on the keyboard. For this task 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 [23]).

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 in to classify audio signals, but it may not be 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 anyway, it is reasonable to distinguish the sounds of the keyboard from other sounds. In the next section we describe the features used for this task.

3. ACTIVITY CLASSIFICATION AND POSTURE MEASUREMENT

3.1 Feature Extraction

Once the largest region has been selected, as described in the previous section, the system extracts several features as follows:

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 45° we obtain 8 lines).

3. For each line, find the external boundaries of the region (see lines in Figure 4). These points define a polygon used for activity classification (in our experiments in section 5 we use the length of each line to represent the polygon).

In addition, from the audio signal we extract volume, mean pitch, pitch standard deviation, and pitch intensity, using the method described in [23].

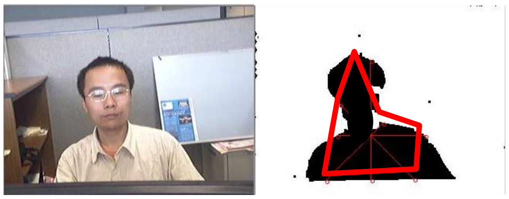


Figure 4. Features extracted from largest foreground region.

The features are used for classifying activities as described in the next section.

3.2 Training and Classification

The user trains the system (Figure 2) by showing examples of good and bad postures and by showing examples of his common activities. This is done because only the user can really determine what he considers good (or comfortable) postures, using either recommended ergonomic guidelines or his own preferences. The types of activities that each person performs at his desk might also vary, so the user can also decide which activities to include. 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.).

For each example we obtain the largest region and extract the features just described: bounding box width, bounding box length, bounding box x and y location, center of mass of region, perimeter of region, region area, angle of main region axis, length of primary region axis, length of secondary region axis, Feret’s diameter (the greatest distance possible between any two points along the boundary of a region), and eccentricity. We then concatenate these features with the length of each of the lines originating in the center of mass and obtain, for each example, an n-dimensional feature vector $fv=\{f_1, f_2, \ldots, f_n\}$. For instance, if we use 16 lines from the center of mass, we obtain 29 features (16 line lengths plus the 13 features described). There is some redundancy in this measure since the polygon vertices are sufficient to distinguish some of the activities.

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, sitting only, etc.).

In addition, our system uses input from the microphone to reinforce the classification results for activities that involve voice, namely typing, speaking on the phone and speaking to someone nearby. In the first case, it is expected that the user will give the system examples where he is holding the phone. Adding the voice constraint helps differentiate postures similar to those when the phone is used (in Figure 10 the second posture from top to bottom is similar to the posture when holding a phone). The audio component is very simple: there will only be significant audio input in a limited number of cases: (1) background voices or noise; (2) loudspeaker announcements; (3) person on the phone; or (4) person speaking with a colleague near his desk; (5) keyboard input. Clearly, for cases one and two the energy level is low compared to the energy level of cases three and four. We use volume, mean pitch, pitch standard deviation, and pitch intensity. The feature vector f_a containing these values is used by a learning algorithm to build a classifier, currently for silence, voice, and keyboard activity (see [29] for a discussion on real-time pitch extraction).

In the monitoring stage, we perform the same audio-visual processing to 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 (section 4.2). Then for the *sitting only* posture we compute a measure based on geometric features (e.g., head, shoulder angles) to provide real-time feedback to the user (next section).

3.3 Posture Measurement

In our system we are only interested in real-time feedback when the user is in the *sitting only* position (not performing other activities such as using the phone). In Figure 5 the first two postures (top left) can be considered “correct” postures (according to the user) since he is sitting straight while he looks at the screen. In the rest of the images, the user is in different types of postures. These postures deviate from the “good postures” by a measurable quantity, in particular, by the angle of the head, and the angle of the shoulders.

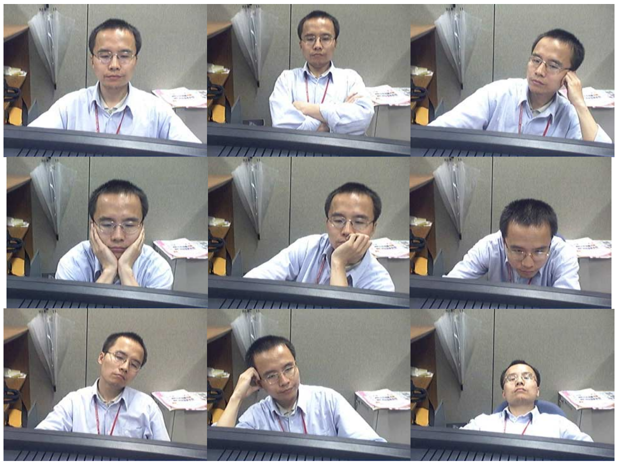
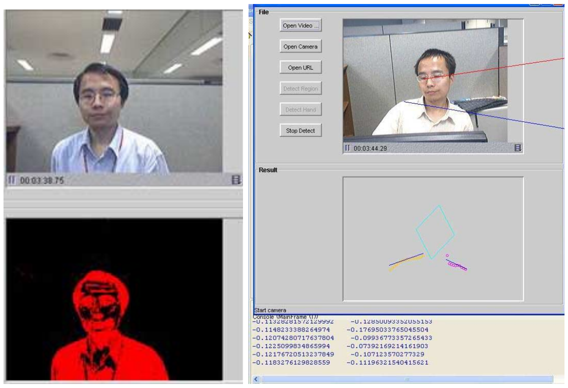


Figure 5. Different postures by the user.

In order to give the user real-time feedback about his sitting only posture, we extract an additional set of features as follows:

1. Obtain vertical projection profile for the extracted region (blue area on left in Figure 11).
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 green lines in Figure 11).
3. Fit a diamond to the head (Figure 6, 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 6).



```
-0.11328281572129992    -0.12850093352055153
-0.1148233388264974    -0.17695033765045504
-0.12074280717637804    -0.09936773357265433
-0.1225099834865994    -0.07392169214161903
-0.12176720513237849    -0.107123570277329
-0.1183276129828559    -0.11196321540415621
```

Figure 6. Foreground object (left) and automatic posture measurement (lower right).

Figure 7 shows an example of the angles extracted from two images. As the figure shows, for the "good posture" (top), the angle of the head inclination (yellow) is close to 90°. The three angles (head, shoulder 1, shoulder 2) are used to determine the goodness of the posture.

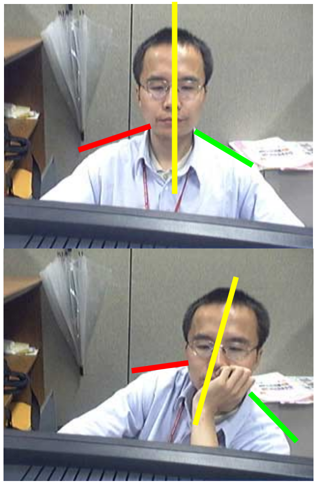


Figure 7. Features extracted in each frame.

4. SELF-REPORTS

4.1 Real-Time Feedback

The interface may show the user his own image and the angles (as shown in Figure 8) in real time. A small icon on the bottom of the screen shows the user the main features of his posture, thus he can get immediate feedback on his posture (e.g., is his head straight? are his shoulders straight?).

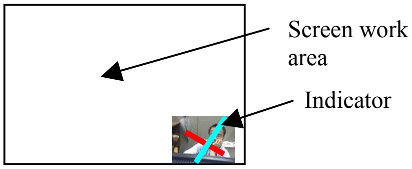


Figure 8. Real time interface.

In most cases, however, the user may just want to see a simple indicator that tells him how good his posture is. For this purpose, we implement several alternatives. An example is shown in Figure 9. When the bar is in the green area it indicates the user's posture is OK (according to his "good posture" examples). If the black bar is on the red areas it indicates that the user is not in good posture (e.g., leaning right or left).

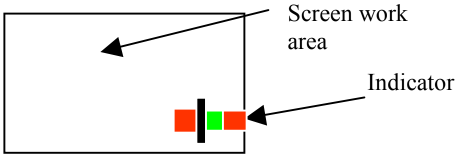


Figure 9. Posture monitor in real-time using simple indicator.

Note that feedback in the first case (Figure 8) does not require any type of classification since it simply shows the user's current head, shoulder, and torso angles. For the second case (Figure 9) we simply measure the similarity between the training examples and current posture.

4.2 Activity And Posture Summaries

Since the system is meant to run continuously, the user is able to obtain automatically generated summaries for any time period, as depicted in Figure 10 (e.g., a day, a week, the last hour, etc.). As explained earlier, however, the user first sets up a summary profile so that the system can show a summary that is relevant for his purposes. The summary profile contains a list of postures or activities that the user wants to include in the summary.

A summary, therefore, can be generated for activities and/or postures. Activities summarized may include reading, speaking on the phone, speaking to someone, filing documents or looking for something near the desk.

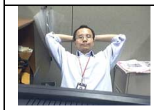
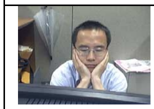
	40% of the time.
	20% of the time.
	25% of the time.
	15% of the time.

Figure 10. Sample posture summary for a determined time-period (e.g., a day, a week, etc.).

5. EXPERIMENTS

We have implemented a first prototype system in Java using ImageJ [2] and Weka [34]. The current implementation, with a standard webcam (320x240) runs at 15 frames per second.

We performed four sets of experiments to evaluate our system. In the first experiment we evaluated the head and shoulder angle extraction for posture monitoring. In the second experiment we evaluated activity classification using a subset of the visual features (only polygon line lengths). In the third experiment we evaluated activity classification using all of the visual features. Finally, we evaluated the audio classification component.

5.1 Experiment One

In the first experiment we compared the head and shoulder angles obtained automatically with their manual counterparts. For the experiment we randomly selected 8 "sitting in front of the computer"[2] postures and compared the angles. The angles were compared 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 both cases. Figure 11 shows several of the images in this experiment. The lines on the top images are drawn manually, and the areas in blue correspond to the horizontal and vertical projections. The green line shows the automatic separation of head and torso, and the red line vertices define the polygon used for activity classification. The diamond (light green inside head) over the head silhouette is also obtained automatically, as are the red shoulder lines (see top left image in Figure 11).

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

In general, and as the images suggest, we found that head-body separation is not problematic when the user sits in front of the computer with his hands down. Difficulties arise when the hands are lifted or when the user performs other activities such as speaking on the telephone (see Figure 12(h)). However, head-torso separation is only of interest in the sitting case when none of the other activities are being performed. We did not consider users with long hair, which could be problematic using this technique. Using the results of the skin detector, however, would allow us to treat such cases.

Although in this experiment we did not use multiple (positive, negative) examples, the idea, as described in section 2.2 is to use the angle measurements to determine how good or bad the posture is with respect to the examples given by the user.

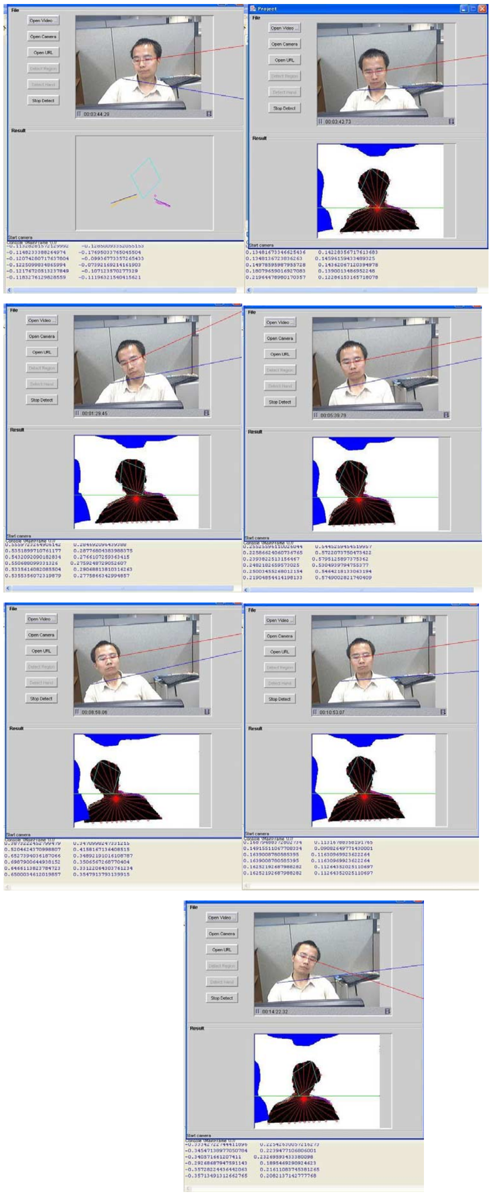


Figure 11. Extraction of head, and shoulder angles.

5.2 Experiment Two

For the second experiment we recorded a set of activities and extracted a feature vector for each activity. The second author participated in this experiment (four persons participated in experiment 3, described below). The author sat in front of the computer and performed the following activities: sit, read, write, speak on the phone, stretch, and others. As the person sat in front of the computer performing each activity, several video frames were obtained for each category (e.g., if the user is

[2] As mentioned earlier, we only measure the angles when the activity is *sitting only* (e.g., not writing on a board, etc.)

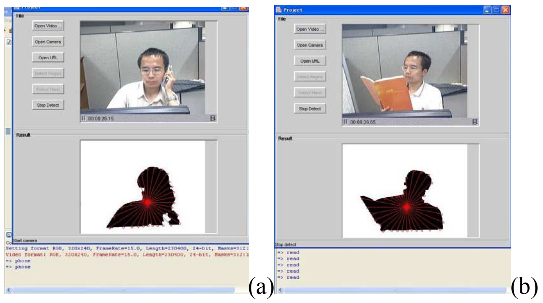
reading for x minutes, y sample frames are extracted for that particular activity). The process described in section 3.1 was applied, but in this experiment we only used the lengths of each of the polygon lines (experiment 3 uses all the features).

In this process 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.

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 examples for experiment two are shown in Figure 12 (using 32 lines originating in the center of mass). The system succeeded in cases (a) through (g) and failed in cases (h) through (l). The errors in cases (h) and (i) could be easily eliminated incorporating the results of the audio analysis from the microphone. Case (j) is interesting because it shows one of the limitations of using only the vertices: including additional features (e.g., silhouette bounding box) would improve performance in this case. Case (k) is more difficult and shows the limitation of using only the silhouette. An alternative here would be to detect lines within the silhouette. But if the user is reading a paper document, at least in this case it would be hard to do the correct classification (difficult to separate document from body). Finally, in case (l) the system determined the user was on the phone. As with cases (h) and (i), the use of audio would improve the performance.



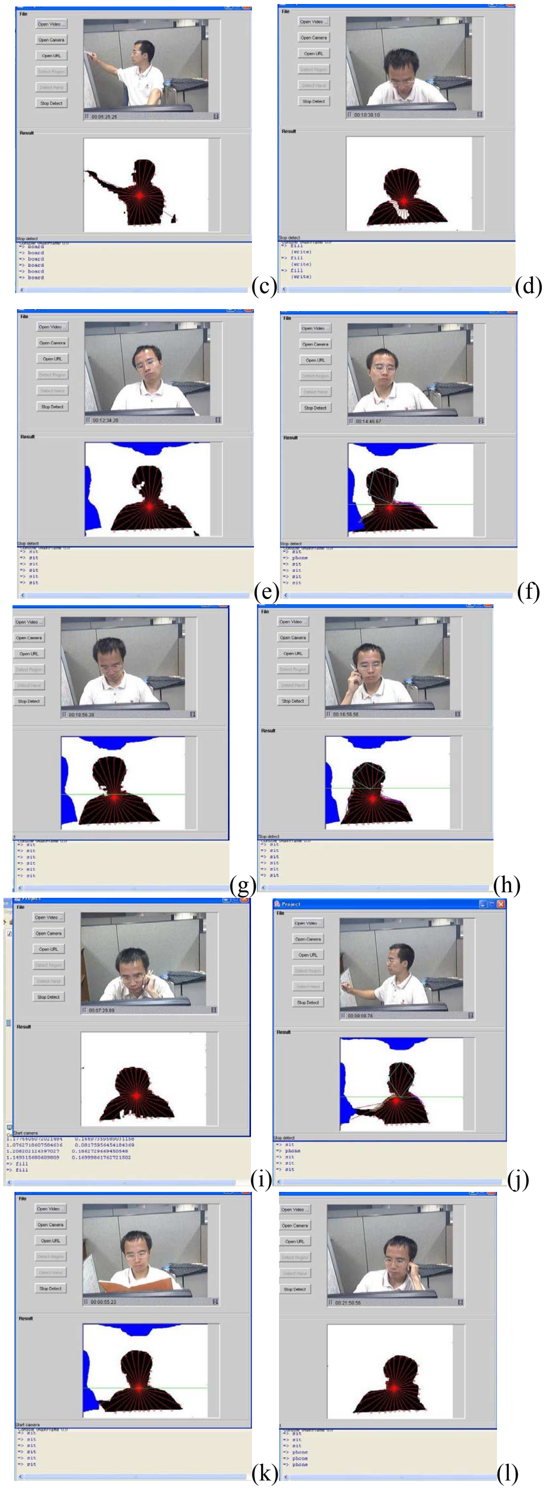


Figure 12. Classification of different activities.

5.3 Experiment Three

In the third experiment we asked four subjects to perform the following tasks while sitting in front of the computer: 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

13). 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).

Table 3. Results of automatic binary classification (in % values) for data of four people, using 1-nearest neighbor (IB1), multilayer perceptron (MLP), and SVM classifiers. Precision (P) and Recall (R) values are shown.

	IB1		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 %	

Table 4. Results of automatic classification (in % values), using 1-nearest neighbor (IB1), multilayer perceptron (MLP), and SVM. Precision (P) and Recall (R) values are shown.

	IB1		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 expected, performance of the binary classifier is, overall, slightly higher than for the n-class classifier.

5.4 Experiment Four

In the fourth experiment, we tested an audio classifier built using training by one of the authors. 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 2.4, namely volume, mean pitch, pitch standard deviation, and pitch intensity, as implemented in [23]. We used the MAD framework in Matlab to extract and pitch using autocorrelation [10]. Pitch was obtained 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 combing the results could have an important impact on performance.

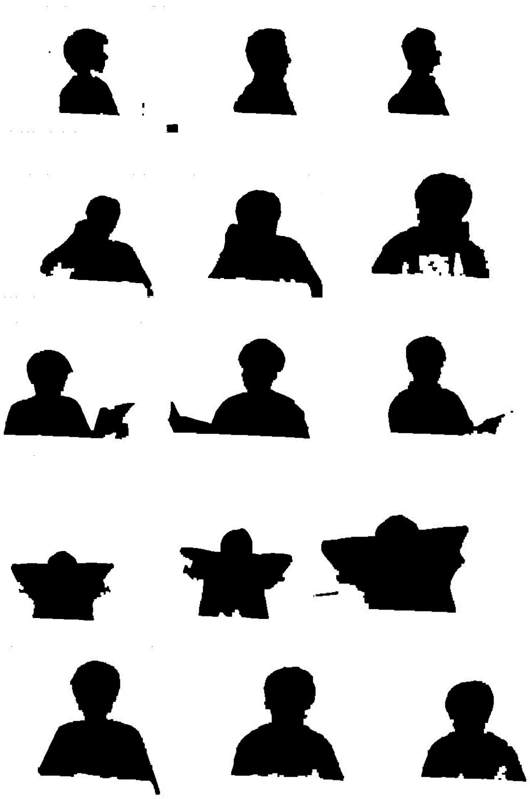


Figure 13. Example silhouettes obtained from experiment 3 for various activities by several individuals (from top to bottom: converse with a colleague, speak on the phone, read, stretch, and sit).

6. APPLICATIONS

As mentioned earlier, the goal of the system is self-reporting. In other words, the information provided by the system is in principle only to be used by the user himself. This decision depends on the particular deployment, however, as group data (maintaining individual privacy) can also be very useful (e.g., get an estimate of productivity of a group of workers; determine posture problems in a group are due to uncomfortable chairs, etc.).

The system can be used for the following purposes:

- Ergonomics: the posture detection and alarm system can help reduce fatigue and work-related injuries.
- Time-management: the activity summaries can help the user manage his time in a better way.
- Productivity measurement: the data collected by the system can be used to measure worker productivity (e.g., approximate times spent on different tasks)
- Worker well-being: the data could be used by the user to discover changes or patterns in his affective state. Long

periods of inactivity or fixed posture can be interpreted as periods of high stress or depression.

Next we discuss in detail the two major applications: (1) health monitoring; (2) productivity analysis.

6.1 Health Monitoring

The user sets up a "user profile" for the alarms and summary. In particular, he sets parameters that determine *when* an alarm is triggered and *what* the summary contains (see summary example in Figure 10). For example, the user might want to see a summary after 6 months and show it to his doctor, who can use the summary to explain why the user might be having shoulder or back pains. He can specify if the summary should contain the time spent on each posture, and which postures are important for the summary (e.g., he might be interested in only one or two postures).

The health monitoring application can be used by anyone, but it is of particular interest to information workers that spend several hours per day in front of the computer. This includes secretaries, researchers, data entry operators, and many others.

6.2 Productivity Analysis

The purpose of the productivity analysis application is to let the user monitor *his own* productivity while working in front of the computer. The user, as described in Figure 2, trains the system to indicate different activities. Examples include: typing, reading, stretching, talking to someone, filing, speaking on the phone, etc.

The "summary profile" is customized by the user to fit his own needs, which will vary depending on his preferences and particular occupation. For instance, if the worker is a salesperson, then speaking on the phone for a long time might be considered positive. In such case, the "productivity analysis" done by the person, based on the summary, will determine that his productivity is high if he spent a long time on the phone.

The user can view a summary of his activities at any time. Thus, this will help him quickly determine how much time he is spending on different activities and adjust his work for the rest of the day, week, or month to compensate. For example, if the researcher notices he has not been doing any reading in the last two days, he may set time apart for reading papers.

We note that the posture analysis system can also be used for productivity analysis: if the summary shows that the user spends a long time in a similar posture (or postures), it may indicate fatigue or low productivity.

6.3 Privacy

Privacy is an important issue in any application that monitors users. The idea in our system is to give the user full control over the application. In particular, privacy in the system is maintained through different functionalities:

- The user may chose to not save any of the information. Thus, the summaries are discarded after they are viewed, and no information is saved.
- The user may chose to save only anonymous pictures of his activities. In this case, the system only saves the silhouette (e.g., bottom of Figure 6). This is important because his facial expressions may be considered very private (the user may also choose to mask only the faces)
- The audio component does not record the audio or the conversations, it only records activity, thus respecting user privacy.
- The system may store all of the information in encrypted form to prevent others from viewing it.

6.4 Extensions

The following extensions can be made to the framework. Using similar techniques it is possible to also monitor:

- Hand positions and hand postures. Summaries can be generated and an alarm system can be implemented as above.
- Legs and feet: the same as above. Additional cameras are required, but lighting will likely be a problem, so infrared cameras may be more appropriate.
- Integration with other sensors: it is possible to integrate the framework with other types of sensors (e.g., chair sensors, etc.)
- Integration with information from other input devices/software (e.g., monitor mouse and keyboard usage, web page, word processor, etc.)
- 3D pose estimation: the visual analysis algorithms could be improved to estimate the 3D position of the upper body and give a more accurate measure of position. Alternatively, side views could also be used to improve robustness.

Additional processing can also be performed to improve performance and extend the framework. For example, vision-based eye tracking could be used to determine where the person is looking (see [22] for a brief review). This information could be used for activity classification and to complement the posture measurement. The system could also be used for building attentive interfaces [5], or for interruption management [3]. The current version only processes one frame at a time, but using several frames might lead to better performance.

An important aspect we have not addressed yet is how users will actually react to the system's summaries and alarms. One of the problems with mouse and keyboard monitoring software is that they blindly interrupt the user at undesirable times, thus perhaps causing more frustration than helping the user improve his work habits. We tried to address this by providing unobtrusive real time feedback, but user studies are necessary to determine how effective this method is. The issue, then, is not only performance, but how and when alarms are used.

7. CONCLUSIONS & FUTURE WORK

We have presented a novel system for monitoring a computer user's posture (i.e., body position in front of the computer) and activities (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 monitors the computer user's postures and summarizes his or her activities. The system gives the user real time feedback on the goodness of his current posture, triggers alarms if the postures are not good postures, and generates summaries of postures and activities over a specified period of time (e.g., hours, days, months, etc.). The algorithms measure the computer user's posture using geometric features, and use machine learning for activity classification.

Our first prototype of the system shows promising results. However, more work is needed in increasing performance and user testing. In particular, future work includes using more sophisticated detection algorithms for estimating 3D pose, incorporating additional monitoring functionalities (e.g., keyboard use), and using more cameras and sensors.

8. REFERENCES

[1] *A Guide to Occupational Health and Safety in the New Zealand Sex Industry*. Occupational Safety and Health Service, Department 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 Information Retrieval in conjunction with SIGIR 2005*, Rio de Janeiro, Brazil, 2005.

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

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

[7] 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

[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 Transactions on Instrumentation and Measurement*, Vol. 50, No. 4, August 2001.

[10] M.P. Cooke, and G.J. Brown, "Interactive explorations in speech and hearing." *J. Acoust. Soc. Japan (E)*, 20, 2, 89-97, 1999 (http://www.dcs.shef.ac.uk/~martin/).

[11] 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.

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

[13] A. Elgammal, V. Shet, Y. Yacoob, L. S. Davis, "Gesture Recognition using a Probabilistic Framework for Pose Matching," *The Seventh International Conference on Control, Automation, Robotics and Vision, ICARCV 2002*, Singapore in December 2-5, 2002.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[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. *Conceptual Structures and Computational Methods for Indexing and Organization of Visual Information*. Ph.D. Thesis*, Department of Electrical Engineering*, Columbia University, February 2003.

[22] A. Jaimes and N. Sebe, "Multimodal Human Computer Interaction: A Survey," *IEEE International Workshop on Human-Computer Interaction (HCI 2005) in conjunction with IEEE International Conference on Computer Vision (ICCV 2005)*, Beijing, China, Oct. 15-21, 2005.

[23] A. Jaimes, J. Liu, and N. Sebe, "Affective Meeting Video Analysis," in proc. *IEEE ICME 2005*, Amsterdam, The Netherlands, July 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,” *IEEE Conference on Computer Vision and Pattern Recognition 2000*, pp.721-727 vol.2, Hilton Head Island, USA, June 2000.

[29] F. Sha and L. K. Saul, “Real-Time Pitch Determination of One or More Voices by Nonnegative Matrix Factorization,” in L. K. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems 17*. MIT Press: Cambridge, MA, 2005.

[30] 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

[31] 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.

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

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

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

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

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

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

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

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

[42] Sonet Acoustic Privacy System (http://www.speechprivacysystems.com/)

[43] http://www.ergoweb.com

[44] http://www.hfes.org

[45] http://www.m-w.com

[46] http://ergonomicsinhealthcare.org