A Framework of Personal Assistant for Computer Users by Analyzing Video Stream

Zixuan Wang
Electrical Engineering
Stanford University
zxwang@stanford.edu

Jinyun Yan
Computer Science
Rutgers, New Brunswick
jinyuny@cs.rutgers.edu

Hamid Aghajan
Electrical Engineering
Stanford University
aghajan@stanford.edu

ABSTRACT
The engagement time on the computer is increasing steadily with the rapid development of the Internet. During the long period in front of the computer, bad postures and habits will result in some health risks, and the unawareness of fatigue will impair the work efficiency. We investigate how users behave in front of the computer with a camera. Face pose, eye gaze, eye blinking, and yawn frequency are considered. These visual cues are then used to give suggestions to users for correcting wrong posture and indicating the need for a break. We propose a novel framework of personal assistant for a user when he uses computer for a long time. The camera produces the video stream which records the user behavior. The automatically assistant system will analyze the visual inputs and give suggestions at the right time. Our experiment shows that it achieves high accuracy of detecting visual cues, and makes reasonable suggestions to users. The work initializes the area of assistant system for individuals who use computer frequently.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Representation]: User Interfaces

General Terms
Application

Keywords
Face Tracking, Eye Gaze, Health Assistant

1. INTRODUCTION
The invention of the computer and the Internet brings tremendous convenience to human life. It even changes the world in many aspects: connecting people all over the world, making oversea transactions easy and quick, etc. A study shows that the average time a person uses computer per day increased to 2 hours. In particular, some fields can not work well without a computer, like trading and software development companies.

However, some serious health risks arise for using computer, for example, eye diseases like dry eyes are due to using eyes continuously for several hours. Other common symptoms, e.g. back, shoulder and wrist pain, are also because of computer overuse. Some people also ignore the fatigue signal when they use computer, and in the long run, this will impair the work efficiency and causes headache and stress. In addition, sitting still for a long time, a blood clot forms in the leg due to long period of inactivity, which will cause more serious problems.

When people are obsessed with the computer for whatever reasons: entertainment or working, an assistant system which can remind them of their bad postures and the fatigue status will be very useful. We propose a framework showing in Figure 1. The camera monitors the user, and sends the video stream to the feature extraction component which extracts face pose, eye gaze, eye blinking and yawn frequency through the video stream every 10 seconds. According to the extracted features, the analysis component will determine whether to give corresponding suggestions using a heuristic algorithm. The main challenges lie in the feature extraction from the video stream, and the feature selection for the heuristic suggestion algorithm.

The main contributions of the paper are as follows:
1. We propose a novel framework that help users who use computer heavily everyday to prevent potential diseases.
2. We extract useful features, including face pose, eye blinking, eye gaze and yawn frequency, from the video stream to determine user’s postures and the fatigue status.
3. We come up a heuristic algorithm which works well in practice to make suggestions to the user based on previous behaviors.

The paper is organized as follows. Section 2 outlines the related work. Section 3.1 describes how we extract each feature from the video stream. Section 3.2 explains our heuristic suggestion algorithm. Section 4 concludes the paper and discusses our future work.

2. RELATED WORK
Fatigue detection from face images has been explored by many researchers. In [1], the authors proposed an efficient
approach to recognize facial expressions of interest. The current facial expression can be classified with a desired confident level via belief propagation. Fatigue detection can also be used when driving [2] [5]. In [5], the authors presented an intelligent vehicle control based on driver fatigue detection. The eyes are tracked based on Unscented Kalman Filter in real time. Driver fatigue can be detected whether the eyes are closed over 5 consecutive frames using vertical projection matching.

3. THE FRAMEWORK

![Figure 1: The overview of our system.](image)

As shown in Figure 1, the system contains three major components: feature extraction, feature analysis and recommendation. The input of the feature extraction component is the video stream recorded from the camera attached to the computer. We set the length of the time window to be 10 seconds, which means a feature vector is generated from images in the past 10 seconds.

For the feature extraction part, we are interested in features which give cues for the posture and fatigue status: the face pose including the face orientation and the distance between the face and the screen; the eye blinking rate, the eye gaze at which the eyes stare more than 100ms and the yawn frequency. These features are used by other studies for driver fatigue detection. Since our purpose is to correct the bad posture, our feature extraction and usage will be of some difference, and this will be explained in the following Section 3.1.

At every 10 seconds, we build a feature vector $x_t$. The feature analysis and recommendation part will analyze the feature and consider the momentum of each feature for a period of past 2 minutes. Then a heuristic algorithm is proposed based on the momentum of each feature. Our recommendation is presented combining with a text summary of the user’s behavior. One example is that “We notice you’ve been too close to screen during the past 15 minutes. We suggest you adjusting the distance between you and the screen to protect your eyes”.

3.1 Feature Extraction

Here we describe all features we used and the detail of how we extract them.

3.1.1 Face Features

![Figure 2: The pipeline of the face pose estimation.](image)

Face pose feature includes face orientation, the distance between the face and the screen and the yawn frequency.

**Face orientation:** We call the frontal face facing the center of the screen the canonical pose, which is the good posture for health. The face pose is the offset of the canonical pose.

To estimate the face pose of each frame, we adopt the local descriptor-based pipeline as shown in Figure 2. The Viola-Jones face detector [4] is used to detect face patches from the input frame. All face patches are normalized to the same size. We use the algorithm in [3] to detect seven facial landmarks from each face patch including four eye corners, two mouth corners and the nose. To align each face patch, seven landmarks are registered to the canonical pose using the perspective transformation. The canonical position of each landmark is obtained from averaging landmarks of all faces. Then all seven facial landmarks are aligned by the computed homography matrix. Let $X^f = (x_0^f, x_1^f, 1)^T$ be the homogeneous coordinates for the landmark $f$ of a non-aligned image, and $Y^f = (y_0^f, y_1^f, 1)^T$ the desired coordinates for the same landmark. We want to obtain the homography $A_{3 \times 3}$ such that $y^f = Ax^f$. To obtain the eight parameters of $A$ four landmarks are needed, however, we can use more landmarks to minimize the least squares error. We use the first eight entries of $A$ to form the feature vector to describe the face pose: $p = (a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32})^T$.

**Face Distance:** We use the pinhole camera model to estimate the distance between the camera to the face:

$$d = \frac{X \times f}{x}$$

where $x$ is the size of the object on the image plane, $f$ is focal length of the camera, which can be calculated from the camera calibration, $X$ is the size of the object, $d$ is the distance from the camera to the object. We use the distance between two eyes as $X$, which is obtained by taking pictures from a bunch of people under fixed $f$ and $d$ and averaging all $X$ as the ground truth.

**Yawn Frequency:** We use template matching(open mouth, closed mouth) to detect the frequency of yawn. The yawn frequency is then estimated by the number of times of open mouth during the 10s window: $y = \text{count(open mouth)}/10$.

3.1.2 Eye Features

![Figure 3: The pipeline of the eye features extraction.](image)

Eye features contains the eye gaze direction and the eye blinking rate. Figure 3 shows the pipeline to extract eye features from the video stream.

**Eye Gaze Direction:** We first extract the eye patch from the detected face. This is done by drawing a bounding box around the detected eye landmarks. The position of the pupil is localized using the template matching. We divide the bounding box of an eye into $3 \times 3$ grids as in Figure 4. We use a histogram $g$ to record the position of the pupil. There are five bins representing five directions: left, right, center, up and down. Each bin in the histogram represents the probability that the pupil is in the direction. Each direction covers three grids. For example, the probability of being left is the sum of probability that the pupil is in one of three left grids.

**Eye Blinking Rate:** We use two templates(open eyes and closed eyes) matching to detect the eye blinking rate. The blinking rate is estimated by the number of closed eyes
within the 10 second time window: $r = \frac{\text{count(closed eyes)}}{10}$.

### 3.2 Heuristic Recommendation Algorithm

Given the feature vector $x_t = (p_t^i, d, y, g_t^i, r)^T$ generated through feature extraction every 10 seconds, every minute, we compute the feature of this minute as the average of

\[x_{\text{minute}} = \frac{1}{t} \sum_{i=1}^{t} (x_i)\]

where $t$ represents every 10 seconds.

We set the check point to be every past 15 minutes including the current minute. For the face distance, we store 15 values of the average distance in a minute, and put them in a vector $V_d$. We estimate the overall face distance in the past 15 minutes by its mean and standard deviation:

\[\bar{d} = \mu(V_d) + \sigma(V_d)\]

If $\bar{d} < \delta_d$, we will generate a suggestion that “During past 15 minutes, you have been less than $\delta$ meters away from the screen, we suggest you to sit back to protect your eyes”.

We do the same computation for the eye blinking rate and the yawning frequency, and compute $\bar{\hat{r}}$ and $\bar{\hat{y}}$. If $\bar{\hat{r}} > \delta_r$ and $\bar{\hat{y}} > \delta_y$, we assume the user is tired. A suggestion that “You yawned and blinked frequently. It might be better to take a break or have a cup of coffee” will prompt out to the user.

We compose a training set with plenty of labeled face images, which contains samples in 9 directions of the face on the image: upper-left, upper, upper-right, left, right, lower-left, lower, lower-right, lower and center. We then train a classifier which classifies each $p$ to the correct face orientation. This is prepared offline. When the system starts, we use the classifier to identity which orientation the current $p_t$ belongs to. During each 15-minute period, we obtains a histogram that each bin is a direction, and the value of a bin is the number of times the user has been in this direction. We remove the bin for being at center because that is the required for good posture. Let $n_i$ be the value of $i$th bin, and put them into vector $V_p$. We compute $\bar{\hat{p}}$ as the variance of $V_p$. The small value of $\bar{\hat{p}}$ means all directions are equally covered. The big value of $\bar{\hat{p}}$ means that one direction might dominate over this period. We therefore use a threshold for $\bar{\hat{p}}$. If $\bar{\hat{p}} > \delta_u$, the assistant system will recommend the user “You have been facing one direction for a long time. We suggest you to sit straight and protect your neck”.

For eye gaze direction, each time $t$, we also obtain a histogram $g_t$, where each value of a bin means the probability of eyes gazing in that direction. There are five directions in total. For the 15-minute period, we sum up all $g_t$s by bin and put the result into a vector $V_g$. The center is also excluded. The value of $i$th item in $V_g$ now is the probability that the eyes were in the direction. Similarity, we compute the variance of $V_g$ to measure the eye gaze of this period $\bar{\hat{g}}$. If the variance is small, the user moved his pupil often and stayed active. If the variance is big, the user kept staring at one direction rather than the center for a long time. Therefore, if $\bar{\hat{g}} > \delta_g$, we suggest the user that “You stared a position for a long time, are you tried? Maybe you can try taking a rest”.

For all thresholds used for the recommendation, the system will provide an initial setting and the user is able to modify the threshold. The user can also modify the check points, by making it longer than 15 minutes or shorter. We take a conservative approach to make recommendation to the user. If any criteria is satisfied, the suggestion will be made. If more than one criterion are satisfied, we will generate the recommendation text by combining all cases.

### 4. CONCLUSION AND FUTURE WORK

We propose a novel framework to assist users for correcting their bad postures and habits. This is useful for the health and work efficiency of human beings. It can help users avoid the symptoms of diseases caused by using computers too long.

In addition to the real-time recommendation, the framework has rich potential applications. The system records continuous video streams and identify user’s status over time. Thus it can generate a daily, weekly or monthly reports for each user. The target users can be people who suffer from repetitive stress injury caused by non-ergonomic postures and who want to prevent such problem. The report can summarize important information such as the total time of being facing left, or other directions, as well as the histogram of face distance to the screen. One future work is to detect the heights of each shoulder of the user. The report can include the time that left shoulder is higher than the right, and vice versa. Face orientation, distance to the screen and the shoulder imbalance are crucial for indicating non-ergonomic postures. Users can aware how they behave over time.

Another type of target users are students who take online classes. The report can add some important information to let them understand their learning process. One is to estimate when the student studies the best. For instance, a student has the lowest yawn rate and steadiest eye blinking rate at 10 AM for most time. If the user can provide some extra information, like which class he is taking right now, the system is able to tell the student that which class he likes the most, by calculating the posture changing rate and the fatigue frequency.

### 5. REFERENCES


