

Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance

This paper describes a real-time prototype computer vision system for monitoring driver vigilance. The main components of the system consists of a remotely located video CCD camera, a specially designed hardware system for real-time image acquisition and for controlling the illuminator and the alarm system, and various computer vision algorithms for simultaneously, real-time and non-intrusively monitoring various visual bio-behaviors that typically characterize a driver's level of vigilance. The visual behaviors include eyelid movement, face orientation, and gaze movement (pupil movement). The system was tested in a simulating environment with subjects of different ethnic backgrounds, different genders, ages, with/without glasses, and under different illumination conditions, and it was found very robust, reliable and accurate.

© 2002 Elsevier Science Ltd. All rights reserved.

Qiang Ji¹ and Xiaojie Yang²

¹*Department of Electrical, Computer,
and System Engineering Rensselaer Polytechnic Institute,
Troy, NY 12180, USA.*

E-mail: qji@ecse.rpi.edu

²*Department of Computer Science,
University of Nevada, Reno, NV 85977 USA.*

E-mail: xyang@cs.unr.edu

Introduction

The ever-increasing number of traffic accidents in the US due to a diminished driver's vigilance level has become a problem of serious concern to society. Drivers with a diminished vigilance level suffer from a marked decline in their abilities of perception, recognition, and vehicle control, and therefore pose serious danger to their own life and the lives of other people. Statistics show that a leading cause for fatal or injury-causing traffic accidents is due to drivers with a diminished vigilance level. In the trucking industry, 57% fatal truck

accidents are due to driver fatigue. It is the number 1 cause for heavy truck crashes. Seventy percent of American drivers report driving fatigued. With the ever-growing traffic conditions, this problem will further deteriorate. For this reason, developing systems actively monitoring a driver's level of vigilance and alerting the driver of any insecure driving conditions is essential to prevent accidents.

Many efforts [1–9] have been reported in the literature for developing active safety systems intended for reducing the number of automobile accidents due to

reduced vigilance. Among different techniques, the best detection accuracy is achieved with techniques that measure physiological conditions like brain waves, heart rate, and pulse rate [8,10]. Requiring physical contact with drivers (e.g. attaching electrodes) to perform, these techniques are intrusive, causing annoyance to drivers. Good results have also been reported with techniques that monitor eyelid movement and gaze with a head-mounted eye tracker or special contact lens. Results from monitoring head movement [11] with a head-mount device are also encouraging. These techniques, though less intrusive, are still not practically acceptable. A driver's state of vigilance can also be characterized by the behaviors of the vehicle he/she operates. Vehicle behaviors including speed, lateral position, turning angle, and moving course are good indicators of a driver's alertness level. While these techniques may be implemented non-intrusively, they are, nevertheless, subject to several limitations including the vehicle type, driver experiences, and driving conditions [2].

People in fatigue exhibit certain visual behaviors easily observable from changes in their facial features like the eyes, head, and face. Typical visual characteristics observable from the image of a person with reduced alertness level include slow eyelid movement [12, 13], smaller degree of eye openness (or even closed), frequent nodding [14], yawning, gaze (narrowness in the line of sight), sluggish in facial expression, and sagging posture. To make use of these visual cues, another increasingly popular and non-invasive approach for monitoring fatigue is to assess a driver's vigilance level through visual observation of his/her physical conditions using a camera and state-of-the-art technologies in computer vision. Techniques using computer vision are aimed at extracting visual characteristics that typically characterize a driver's vigilance level from his/her video images. In a recent workshop [15] sponsored by the Department of Transportation (DOT) on driver's vigilance, it is concluded that computer vision represents the most promising non-invasive technology to monitor driver's vigilance.

Many efforts have been reported in the literature on developing active real-time image-based fatigue monitoring systems [1–5, 7, 8, 13, 16–19]. These efforts are primarily focused on detecting driver fatigue. For example, Ishii *et al.* [7] introduced a system for characterizing a driver's mental state from his facial expression. Saito *et al.* [1] proposed a vision system to detect a driver's physical and mental conditions from

line of sight (gaze). Boverie *et al.* [3] described a system for monitoring driving vigilance by studying the eyelid movement. Their preliminary evaluation revealed promising results of their system for characterizing a driver's vigilance level using eyelid movement. Ueno *et al.* [2] described a system for drowsiness detection by recognizing whether a driver's eyes are open and closed, and, if open, computing the degree of openness. Their study showed that the performance of their system is comparable to those of techniques using physiological signals.

Despite the success of the existing approaches/systems for extracting characteristics of a driver using computer vision technologies, current efforts in this area, however, focus on using only a single visual cue such as eyelid movement or line of sight or head orientation to characterize driver's state of alertness. The system relying on a single visual cue may encounter difficulty when the required visual features cannot be acquired accurately or reliably. For example, drivers with glasses could pose serious problem to those techniques based on detecting eye characteristics. Glasses can cause glare and may be total opaque to light, making it impossible for camera to monitor eye movement. Furthermore, the degree of eye openness may vary from people to people. Another potential problem with the use of a single visual cue is that the obtained visual feature may not always be indicative of one's mental conditions. For example, the irregular head movement or line of sight (like briefly look back or at the mirror) may yield false alarms for such a system.

All those visual cues, however imperfect they are individually, if combined systematically, can provide an accurate characterization of a driver's level of vigilance. It is our belief that simultaneous extraction and use of multiple visual cues can reduce the uncertainty and resolve the ambiguity present in the information from a single source. The system we propose can simultaneously, non-intrusively, and in real-time monitor several visual behaviors that typically characterize a person's level of alertness while driving. These visual cues include eyelid movement, pupil movement, and face orientation. The fatigue parameters computed from this visual cues are subsequently combined probabilistically to form a composite fatigue index that can robustly, accurately, and consistently characterize one's vigilance level. Figure 1 gives an overview of our driver vigilance monitoring system.

The paper focuses on the development of computer vision algorithms and the necessary hardware compo-

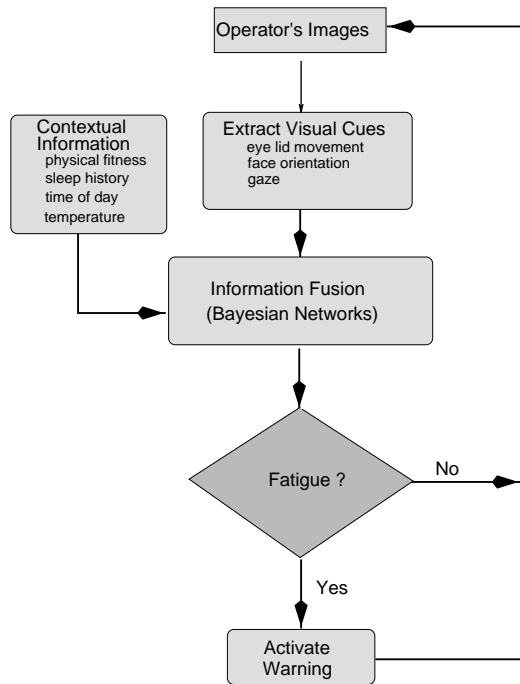


Figure 1. A flowchart of the proposed vigilance monitoring system.

nents to extract the needed visual cues. The issue of sensory data fusion will be dealt with in a separate paper. Figure 2 gives an overview of our visual cues extraction system for driver fatigue monitoring. The system starts with pupil detection and tracking, which is then used for eyelid movement monitoring, gaze

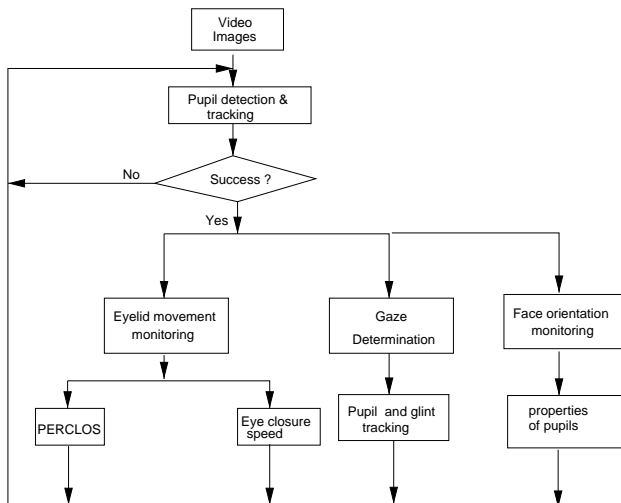


Figure 2. Overview of the driver vigilance monitoring system.

estimation, and face orientation determination, respectively.

Image Acquisition System

Introduction

Image understanding of visual behaviors starts with image acquisition. The purpose of image acquisition is to acquire the video images of the driver face in real time. The acquired images should have relatively consistent photometric property under different climatic/ambient conditions and should produce distinguishable features that can facilitate the subsequent image processing. To this end, the person's face is illuminated using a near-infrared (NIR) illuminator. The use of infrared (IR) illuminator serves three purposes: first, it minimizes the impact of different ambient light conditions, therefore ensuring image quality under varying real-world conditions including poor illumination, day, and night; second, it allows to produce the bright pupil effect, which constitutes the foundation for detection and tracking the proposed visual cues. Third, since NIR is barely visible to the driver, this will minimize any interference with the driver's driving.

According to the original patent from Hutchinson [20], a bright pupil can be obtained if the eyes are illuminated with a NIR illuminator beaming light along the camera optical axis at certain wavelength. At the NIR wavelength, pupils reflect almost all IR light they receive along the path back to the camera, producing the bright pupil effect, very much similar to the red eye effect in photography. If illuminated off the camera optical axis, the pupils appear dark since the reflected light will not enter the camera lens. This produces the so-called dark pupil effects. Figure 3 illustrates the principle of bright and dark pupil effects. An example of the bright/dark pupils is given in Figure 4.

To produce the desired bright pupil effect, the IR illuminator theoretically must locate along the optical axis of the lens. Our experiment reveals that it is physically difficult to place IR light-emitting diodes (LEDs) along the optical axis since it may block the view of the camera, limiting the camera's operational field of view. Others [11] suggested the use of a beam splitter to achieve this purpose without blocking camera view. The set-up, however, is complex.

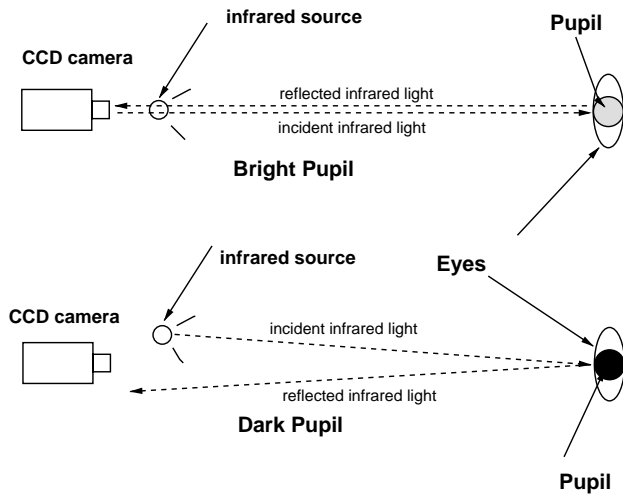


Figure 3. Principle of bright and dark pupil effects.

We present a simple geometric disposition of the IR LEDs similar to that of morimoto et al. [21] that can achieve the same bright pupil effect but with minimal reduction in camera operational view. Specifically, our IR illuminator consists two sets of IR LEDs, distributed evenly and symmetrically along the circumference of two coplanar concentric rings as shown in Figure 5. The center of both rings coincides with the camera optical axis.

Mounted in front of the lens, the optimal sizes of the LED rings are determined empirically so that a dark image is produced if the outer ring is turned on and a bright pupil image is produced if the inner ring is turned on. It represents a trade-off between producing the ideal bright/dark pupil images and minimizing the occlusion of the camera view. The ring configuration achieves the same bright-pupil effect as with LEDs mounted along the optical axis. Besides the advantages of easy installation and a minimal reduction in camera's operational field of view, since LEDs are arranged symmetrically around the camera optical axis, they can

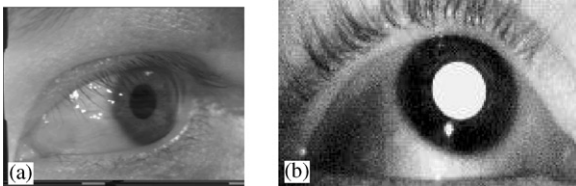


Figure 4. IR illuminated eyes: (a) dark pupil image generated by IR LEDs off the camera optical axis; (b) bright pupil image generated by IR LEDs along the camera optical axis.

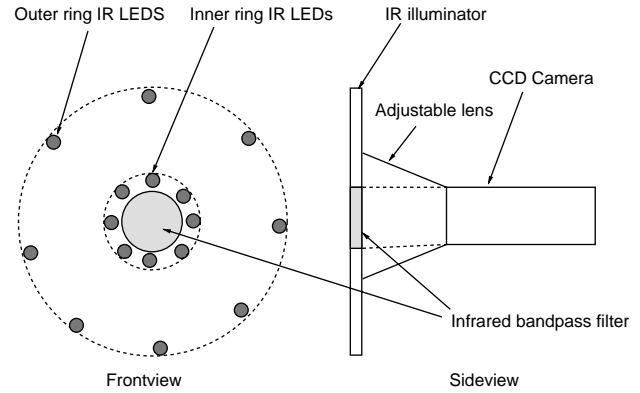


Figure 5. IR light source configuration.

cancel shadows generated by LEDs. Furthermore, the use of multiple IR LEDs can generate a strong light such that the IR illumination from the illuminator dominates the IR radiation exposed to the driver's face, therefore greatly minimizing the IR effect from other sources. This ensures the bright pupil effect under different climatic conditions. The use of more than one LED also allows to produce the bright pupil for subjects far away ($3f$) from camera. To further minimize interference from light sources beyond IR light and to maintain uniform illumination under different climatic conditions, a narrow bandpass NIR filter is attached to the front of the lens to attenuate light beyond NIR range (700–900 nm). A physical set-up of the IR illuminator is shown in Figure 6.

The IR light source illuminates the user's eye and generates two kinds of pupil images: bright and dark

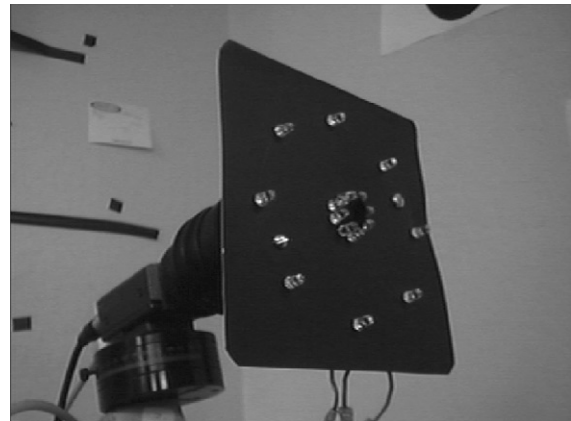


Figure 6. An actual photograph of the two rings IR illuminator configuration.

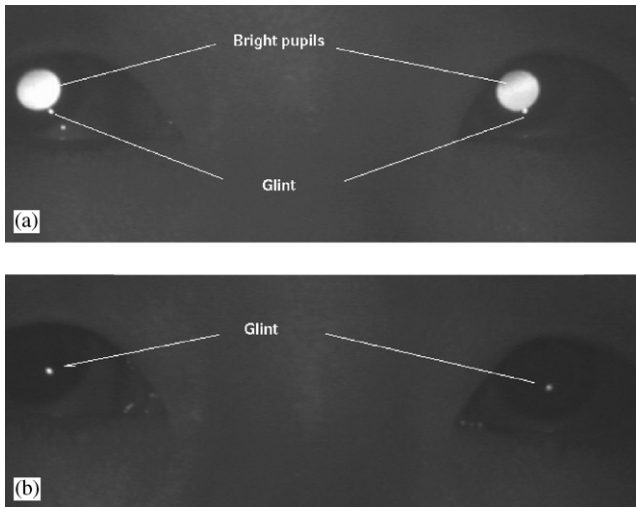


Figure 7. (a) Bright and (b) dark pupil images with glints.

pupil images as shown in Figure 7. The bright pupil image is produced when the inner ring of IR LEDs is turned on and the dark image is produced when the outer ring is turned on. Note, the glint* appears on both the dark and bright pupil images. Figure 8 presents additional examples of the acquired images using the image acquisition system described above. These images demonstrate the robustness of the system in that the desired bright pupil effect is clear for images at different distances, orientations, magnifications, with and without glasses. It even works to certain degree with sunglasses.

For implementation in vehicle, we propose to use two CMOS miniature CCD cameras embedded on the dashboard of the vehicle as shown in Figure 9. The first camera is a narrow angle camera, focusing on the driver's eyes to monitor eyelid movement while the second camera is a wide angle camera that focuses on the driver's head to track and monitor head movement. Two cameras may be motorized so that their movements can be adjusted through a controller to obtain the best picture of the driver. Both cameras should have a rather large depth of field, so that they can stay focused from a working distance of 0.8–1.5 m. The field of angle is between $\pm 45^\circ$ on both the horizontal and vertical directions.

*The small bright spot near the pupil, produced by corneal reflection of the IR light.

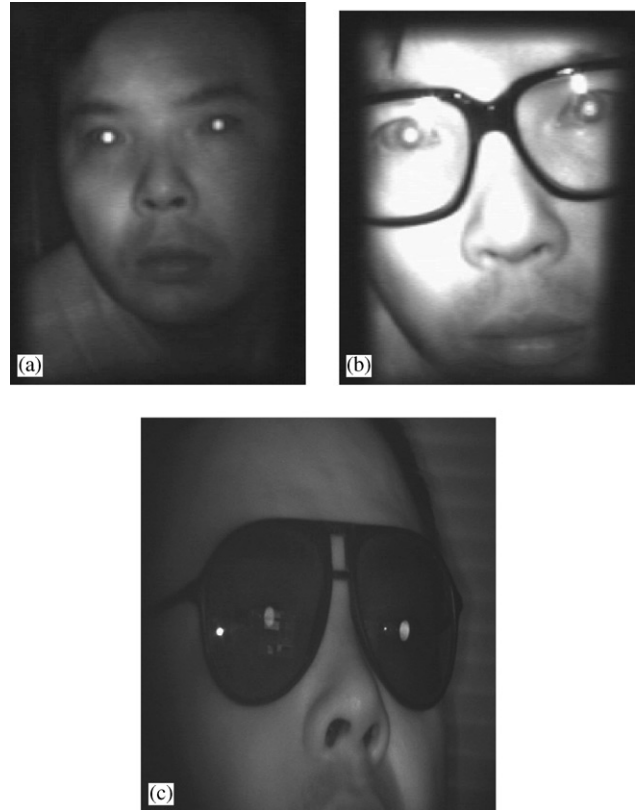


Figure 8. Examples of the acquired images with the desired bright pupil effect. (a) without glasses, (b) with glasses; (c) with sunglasses.

Pupil Detection and Tracking

The goal of pupil detection and tracking is for subsequent eyelid movements monitoring, gaze determination, and face orientation estimation. A robust, accurate, and real-time pupil detection is therefore crucial. Pupil detection and tracking starts with pupil detection.

For this research, we detect pupils based on their intensity, shape, and size. Due to the use of special IR illumination, pupils appear distinctively brighter than the rest of the face. Pupil intensity is, therefore, the primary feature employed to detect pupils. To further separate pupils from other bright objects in the images, additional properties of the pupils are used. These include pupils size, shape, the spatial relationships between pupils, and their motion characteristics. Given the detected pupils, pupils are then tracked efficiently from frame to frame in real time based on Kalman

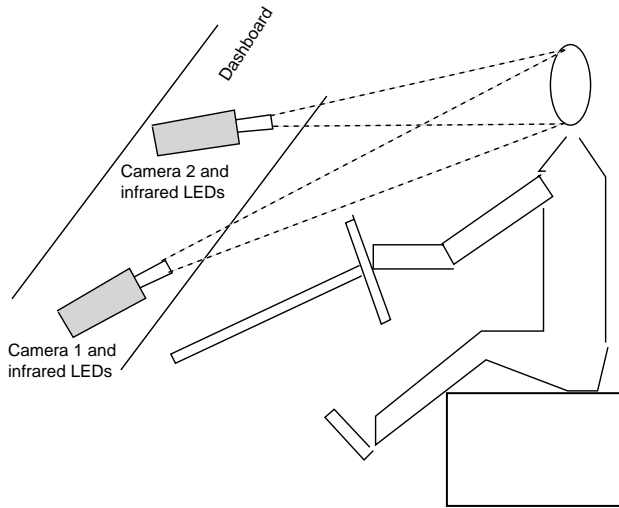


Figure 9. Camera configuration inside a vehicle.

filtering. Figure 10 gives an overview of the eye tracking system.

The system has two phases-pupil detection and pupil tracking. Pupil detection starts with a pre-processing to remove external illumination interference, followed by a global search of the whole image to locate the first pair of pupils in the image. Pupil tracking locally searches the pupils based on the pupils' positions in prior frames. In the sections to follow, we discuss in detail each aspect.

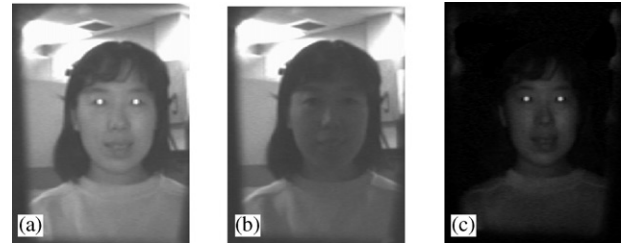


Figure 11. Background illumination interference removal (a) the even image field obtained with both ambient and IR light; (b) the odd image field obtained with only ambient light; (c) the image resulting from subtraction of (b) from (a).

Pupil detection

Pupil detection involves locating pupils in the image. It consists of two steps: illumination interference removal and pupil detection.

Illumination interference removal via image subtraction. The detection algorithm starts with a pre-processing to minimize interference from illumination sources other than the IR illuminator. This includes sunlight and ambient light interference. Figure 11(a) shows an image where parts (upper left) of the background look very bright, almost as bright as the pupil. A simple global thresholding of the image solely based on intensity may not always uniquely detect the pupils due to the presence of other bright regions in the image. To uniquely detect pupils, other bright areas in the image must be removed

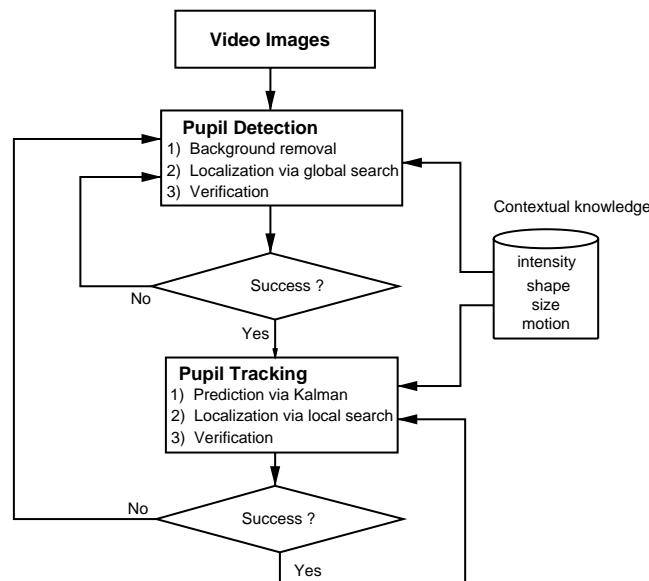


Figure 10. Pupil detection and tracking system flowchart.

or they may adversely affect pupil detection. The background clusters removal is accomplished by subtracting the image with only ambient light illumination from the one illuminated by both the IR illuminator and the ambient light. The resultant image contains the illumination effect from only the IR illuminator, therefore with bright pupils and relatively dark background. This method has been found very effective in improving the robustness and accuracy of our eye tracking algorithm under strong external illumination interference, even under strong incoming light or another IR source nearby, as shown in Figures 11–13.

For real eye tracking, the image subtraction must be implemented efficiently in real time. To achieve this, we design a video decoder that detects from each interlaced image frame (camera output) the even and odd field signals, which is then used to alternately turn the outer and inner IR rings on to produce the dark and bright pupil image fields. A program was then written to separate each frame into two image fields (even and odd), representing the bright and dark pupil images separately. The even image field is then digitally subtracted from the odd image field to produce the difference image. Images in Figures 11–13 are such produced. Figure 14 depicts the block diagram of the image subtraction controller.

Determining the initial pupils positions. Given the image resulted from the removal of the external illumination disturbance, pupils may be detected by searching the

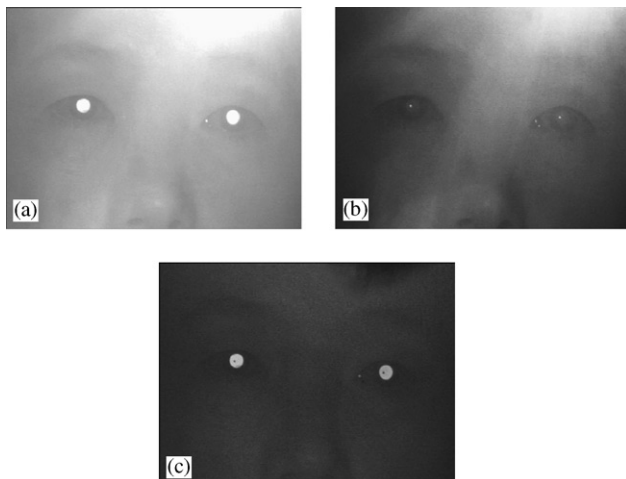


Figure 12. Strong incoming illumination interference removal (a) the image field obtained with both incoming light and IR light; (b) the image field obtained with only the incoming light; (c) the image resulting from subtraction of (b) from (a).

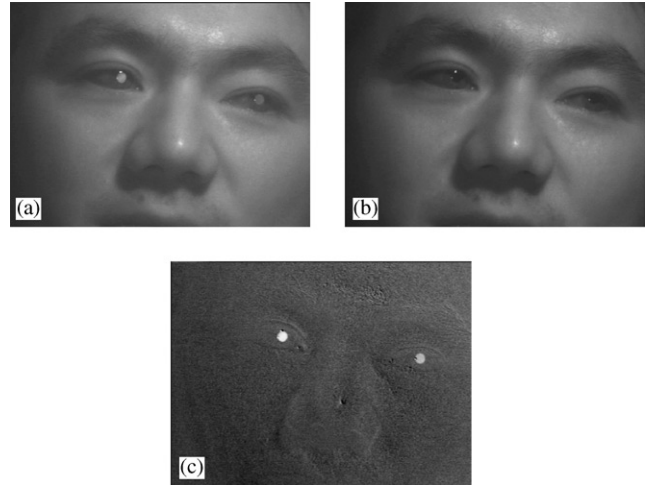


Figure 13. External IR Illumination interference removal: (a) the image field obtained with both the external and IR internal lights; (b) the image field obtained with only the external IR and the ambient light; (c) the image resulting from subtraction of (b) from (a).

entire image to locate two bright regions that satisfy certain size, shape, and distance constraints. To do so, a search window scans through the image. At each location, the portion of the image covered by the window is examined to determine the number of modality of its intensity distribution. It is assumed that the intensity distribution follows an unimodal distribution if the pupil is not covered by the window and follows a bimodal intensity distribution if the window includes the pupil. A thresholding is then applied to the window image if its intensity distribution is determined to be bimodal. The threshold is determined automatically by minimizing Kullback information distance [22]. This yields a binary image consisting of binary blob that may contain a pupil. The binary blob is then validated based on its shape, size, its distance to the other detected pupil, and its motion characteristics to ensure it is a pupil. The validation step is critical since some regions of the image such as the glares of the glasses (see Figure 15, glares cannot be removed by the image subtraction procedure) are equally bright. They may be mistaken for pupils without the verification procedure. The window moves to next position if the validation fails. The centroids of the blob are returned as the position of the detected pupil if the validation succeeds. This then repeats to detect another pupil.

Pupil tracking via Kalman filtering

To continuously monitor the person, it is important to track his/her pupils from frame to frame in real time.

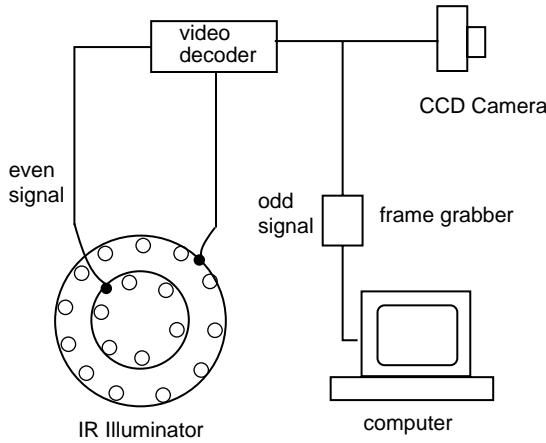


Figure 14. Block diagram of the image subtraction circuitry.

This can be done by performing a pupil detection in each frame. This brutal force method, however, will significantly slow down the speed of pupil tracking, making real-time pupil tracking impossible since it needs to search the entire image for each frame. This can be done more efficiently by using scheme of prediction and localization. Prediction involves determining the approximate locations of pupils in next frame based on its current location. Localization determines the exact location via a local search. The first step ensures efficient accurate localization for the second step since it limits search area in subsequent frames to a small area.

Two important factors need be considered when implementing this scheme. The first one is the search window size for the second step. A large search window results in unnecessary search and is time consuming while a too small search window may easily lose the pupil information. Several factors may influence the

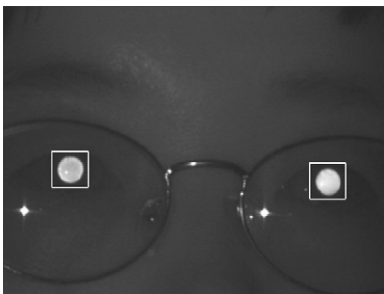


Figure 15. The glares on the eye frame are equally as bright as the pupils. Verification can eliminate them from being considered as pupils.

search window size including pupil size and uncertainty with the predicted position. Pupil size varies with distance and subjects, and positional uncertainty depends on the feature detector and the noise characteristics of the image. An effective way is to use an adaptive search window, whereby the search area is determined automatically based on pupil size and location error. Kalman filtering [23] provides a mechanism to accomplish this. A Kalman filter is a set of recursive algorithms which estimate the position and uncertainty of moving objects in the next time frame, that is, where to look for the pupils, and how large a region should be searched in the next frame, around the predicted position, to be sure to find the pupils with certain confidence. It recursively conditions current estimate on all of the past measurements and the process is repeated with the previous *posterior* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the very appealing feature of the Kalman filter—it makes practical implementation much more feasible.

Our pupil tracking method based on Kalman filtering can be formalized as follows. A sequence of image frames is captured. The image sequence is sampled at each frame t , which is then processed to determine pupil position. The state of a pupil at each time instance (frame) can be characterized by its position and velocity. Let $(\mathbf{c}_t, \mathbf{r}_t)$ represent the pupil pixel position (its centroid) at time t and $(\mathbf{u}_t, \mathbf{v}_t)$ be its velocity at time t in \mathbf{c} and \mathbf{r} directions, respectively. The state vector at time t can, therefore, be represented as $\mathbf{x}_t = (\mathbf{c}_t \ \mathbf{r}_t \ \mathbf{u}_t \ \mathbf{v}_t)^t$.

According to the theory of Kalman filtering [24], \mathbf{x}_{t+1} , the state vector at the next time frame $t+1$, linearly relates to current state \mathbf{x}_t by the system model as follows:

$$\mathbf{x}_{t+1} = \Phi \mathbf{x}_t + \mathbf{w}_t \quad (1)$$

where Φ is the state transition matrix and \mathbf{w}_t represents system perturbation, normally distributed as $\mathbf{w}_t \sim N(0, \mathbf{Q})$.

If we assume pupil movement between two consecutive frames is small enough to consider the motion of pupil positions from frame to frame as uniform, the state transition matrix can be parameterized as

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We further assume that a fast feature detector estimates $\mathbf{z}_t = (\hat{\mathbf{c}}_t, \hat{\mathbf{r}}_t)$, the estimated pupil position at time t .

Therefore, the measurement model in the form needed by the Kalman filter is

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \quad (2)$$

where matrix \mathbf{H} relates current state to current measurement and \mathbf{v}_t represents measurement uncertainty, normally distributed as $\mathbf{v}_t \sim N(0, \mathbf{R})$. For simplicity and since \mathbf{z}_t only involves position, \mathbf{H} can be represented as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The feature detector (e.g. thresholding or correlation) searches for the region determined by the covariance matrix \sum_{t+1}^{-1} to find the feature point at time $t + 1$. The detected point is then combined with the prediction estimation to produce the final estimate. Search region automatically changes based on \sum_{t+1}^{-1} .

For subsequent discussion, let us define a few more variables. Let \mathbf{x}_{t+1}^{-1} be the estimated state at time $t + 1$, resulted from using the system model. It is often referred to as the *prior* state estimate. \mathbf{x}_{t+1} differs from \mathbf{x}_{t+1}^{-1} in that it is estimated using both the system model (Eqn (1)) and the measurement model (Eqn (2)). \mathbf{x}_{t+1} is usually referred as *posterior* state estimate. Let \sum_{t+1}^{-1} and \sum_{t+1} be the covariance matrices for the state estimates \mathbf{x}_{t+1}^{-1} and \mathbf{x}_{t+1} , respectively. They characterize the uncertainties associated with the prior and posterior state estimates. The goal of Kalman filtering is therefore to estimate \mathbf{x}_{t+1} and \sum_{t+1} given \mathbf{x}_t , \sum_t , \mathbf{z}_t , the system, and measurement models. The Kalman filtering algorithm for state prediction and updating may be summarized below.

State prediction. Given current state \mathbf{x}_t and its covariance matrix \sum_t , state prediction involves two steps: state projection (\mathbf{x}_{t+1}^{-1}) and error covariance estimation (\sum_{t+1}^{-1}) as summarized below:

$$\mathbf{x}_{t+1}^{-1} = \Phi\mathbf{x}_t \quad (3)$$

$$\sum_{t+1}^{-1} = \Phi\sum_t\Phi^T + \mathbf{Q}_t \quad (4)$$

State updating. Given the prior estimate \mathbf{x}_{t+1}^{-1} , its covariance matrix \sum_{t+1}^{-1} , and current measurement \mathbf{z}_{t+1} resulted from a feature detection (via either a simple thresholding or correlation method) in the neighborhood determined by \sum_{t+1}^{-1} , state updating can be performed to derive the posterior state and its covariance matrix.

The first task during the measurement update is to compute the Kalman gain \mathbf{K}_{t+1} . It is performed as follows:

$$\mathbf{K}_{t+1} = \frac{\sum_{t+1}^{-1}\mathbf{H}^T}{\mathbf{H}\sum_{t+1}^{-1}\mathbf{H}^T + \mathbf{R}} \quad (5)$$

The gain matrix \mathbf{K} can be physically interpreted as a weighting factor to determine the contribution of measurement \mathbf{z}_{t+1} and prediction $\mathbf{H}\mathbf{x}_{t+1}^{-1}$ to the posterior state estimate \mathbf{x}_{t+1} . The next step is to actually measure the process to obtain \mathbf{z}_t , and then to generate a *posteriori* state estimate \mathbf{x}_{t+1} by incorporation the measurement into Eqn (1). \mathbf{x}_{t+1} is computed as follows:

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^{-1} + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}\mathbf{x}_{t+1}^{-1}) \quad (6)$$

The final step is to obtain a *posteriori* error covariance estimate. It is computed as follows:

$$\sum_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H})\sum_{t+1}^{-1} \quad (7)$$

After each time and measurement update pair, the Kalman filter recursively conditions current estimate on all of the past measurements and the process is repeated with the previous *posterior* estimates used to project or predict a new *a priori* estimate.

In order for the Kalman filter to work, the Kalman filter needs be initialized. First, we need to specify an initial state. We start Kalman filter tracker after we detected the pupils successfully in two consecutive frames. Let the two frames be t and $t + 1$. The initial state vector \mathbf{x}_0 can be specified as

$$\mathbf{c}_0 = \mathbf{c}_{t+1}$$

$$\mathbf{r}_0 = \mathbf{r}_{t+1}$$

$$\mathbf{u}_0 = \mathbf{c}_{t+1} - \mathbf{c}_t$$

$$\mathbf{v}_0 = \mathbf{r}_{t+1} - \mathbf{r}_t$$

We also need to specify the initial covariance matrix \sum_0 for the initial state \mathbf{x}_0 . Since \sum_t is updated iteratively as we acquire more images, we can initialize it to large values. Suppose the predicted position has ± 10 pixels error from the true value in both \mathbf{u} and \mathbf{v} directions, and the speed has ± 5 pixels error from the true value in \mathbf{u} and \mathbf{v} directions, therefore, the estimator error covariance \sum_0 is defined as

$$\sum_0 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

Besides \mathbf{x}_0 and Σ_0 , we also estimate the system and measurement error covariance matrices \mathbf{Q} and \mathbf{R} . Based on observation of the pupil motion, we can safely assume the system perturbation model as follows. The standard deviation from positional system error to be 4 pixels for both \mathbf{u} and \mathbf{v} directions. We further assume that the standard deviation for velocity error to be 2 pixels/frame. Therefore, the state covariance matrix can be quantified as

$$\mathbf{Q} = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Similarly, we can also assume the error for measurement model as 2 pixels for both \mathbf{x} and \mathbf{y} directions. Thus,

$$\mathbf{R} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

Both \mathbf{Q} and \mathbf{R} are assumed be stationary (constant).

Using the state prediction and updating equations, in conjunction with the initial conditions, the state vector and its covariance matrix at each frame is estimated. We observe the covariance matrix Σ_t gradually stabilize as expected after a few image frames. The state covariance matrix Σ_t gives the uncertainty of pupils position in t th frame. The search area can be determined by Σ_t as shown in Figure 16, where the ellipse represents the search region and the major and minor axes of the ellipse is determined by the two eigenvectors of Σ_t . In practice, to speed up the computation, the values of $\Sigma_t[0][0]$ and $\Sigma_t[1][1]$ are used to compute the search

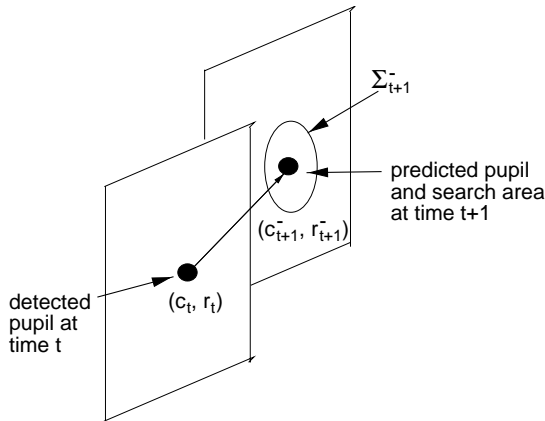


Figure 16. Pupil detection and tracking using Kalman filtering, where $(\hat{\mathbf{c}}_{t+1}, \hat{\mathbf{r}}_{t+1})$ are the predicted position and Σ_{t+1} are the uncertainty associated with $(\hat{\mathbf{c}}_{t+1}, \hat{\mathbf{r}}_{t+1})$.

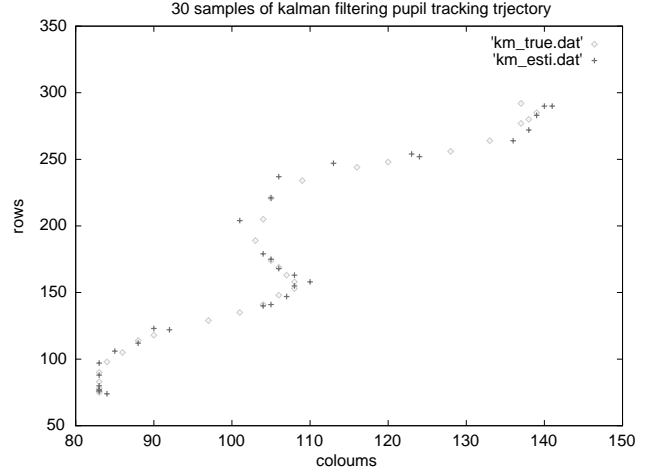


Figure 17. Trajectory of the real and estimated pupil positions in a 30-frame sequence. Crosses indicate the estimated pupil positions via Kalman filter. Small circles indicate the actual tracked pupil positions. It is apparent that the two pupil positions match very well.

area size. Specifically, the search area size is chosen as $50 + 2 \cdot \Sigma[0][0]$ and $50 + 2 \cdot \Sigma[1][1]$, where 50×50 is the basic window size. This means the larger the $\Sigma[0][0]$ and $\Sigma[1][1]$, the more the uncertainty of the estimation, and the larger the search area is. The search area is, therefore, adaptively adjusted. The pupil detection procedure will be reactivated if tracking fails. Tracking fails if the tracked region has a unimodal intensity distribution (only background) as indicated by its variance or the binary blob detected in the tracked region does not satisfy certain geometric constraints such as size and shape.

To study the validity of the Kalman filter for pupil tracking, we study the differences between the predicted and the actual pupil locations as shown in Figure 17 which shows the trajectory of the actual and estimated pupil position in a 30-frame image sequence using Kalman filter. It is clear from this figure that the predicted and the actual pupil positions match well.

Tracking results. The Kalman filter tracker has been implemented and tested on a 300 MHz Ultra 30 Sun workstation, with image resolution of 640×480 . The pupil tracking program and algorithm is found to be rather robust under different face orientations and distances. Specifically, our pupil tracking program allows the subject to freely move in the view of the



Figure 18. The pupil tracking program tracks the pupils as the face rotates.

camera under the illumination of both an IR source and ambient light. It automatically finds and tracks the pupils and can recover from tracking-failures. For pupils temporarily out of the camera view, it can instantly relocate the pupils as soon as they reappear in the camera view. The system runs at a frame-rate of about 20 frames/s. To demonstrate the pupil tracking program, Figure 18 shows a sequence of consecutive frames with rectangles indicating the locations of the detected pupils. Additional images are shown in Figures 19 and 20, which shows pupil tracking for people with glasses. Figure 21 shows the pupil tracking result under strong external illumination interference from a light in front of the subject. Video demonstration of our eye tracking system may be found at <http://www.cs.unr.edu/qiangji/fatigue.html>.

The major advantages of the eye detection and tracking approach include its simplicity, robustness, and accuracy. It is non-invasive and can, therefore, be achieved without even the knowledge of the subject. Furthermore, the technique does not require any physical or geometric model. It is based on the spectral (reflective) properties of pupil. Although special illumination is required, the illumination set-up is simple and the scene background is irrelevant. Pupils can be detected and tracked in real time in a wide range of scales and illumination conditions.

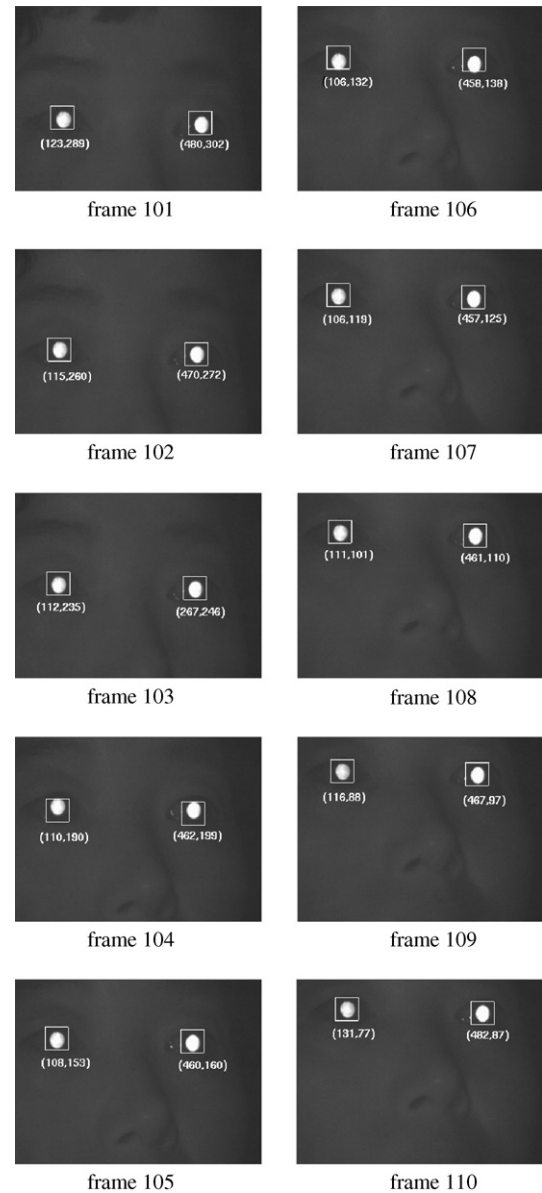


Figure 19. Tracking result in 10 consecutive frames with Kalman filtering track scheme.

Computation of Eyelid Movement Parameters

Introduction

Eyelid movement is one of the visual behaviors that reflect a person's level of fatigue. The primary purpose of pupils tracking is to monitor eyelid movements and to compute the relevant eyelid movement parameters. There are several ocular measures to characterize eyelid

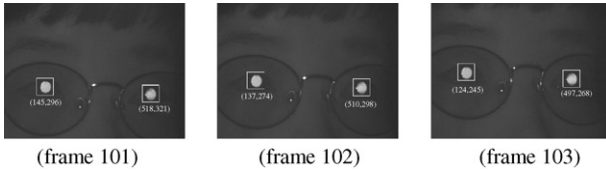


Figure 20. Kalman filtering tracking result with glasses.

movement such as eye blink frequency, eye closure duration, eye closure speed, and the recently developed parameter PERCLOS. PERCLOS measures percentage of eye closure over time, excluding the time spent on normal closure. It has been found to be the most valid ocular parameter for characterizing driver fatigue [13]. Study performed by Wierwillw et al. [25] shows that alert drivers were reported to have much lower PERCLOS measurement than a drowsy driver. Another ocular parameter that could potentially be a good indicator of fatigue is eye closure/opening speed, i.e. the amount of time needed to fully close the eyes and to fully open the eyes. Our preliminary study indicates that the eye closure speed is distinctively different for a drowsy and an alert subject. For this research, we focus on real-time computation of these two parameters to characterize eyelid movement.

To obtain these measurements (PERCLOS and eye closure speed), we propose to continuously track the subject's pupils and determine in real time the amount of eye closure based on the area of the pupils that have been occluded by the eyelids. Specifically, an eye closure



Figure 21. Examples of pupil tracking under strong external illumination interference.

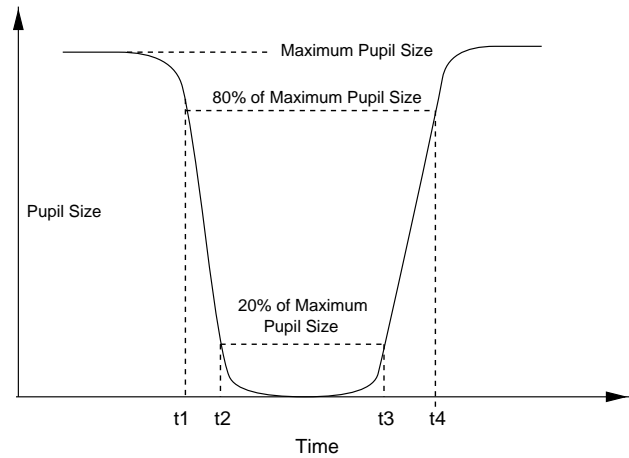


Figure 22. Definition of eye closure duration and eye open/close speed.

occurs when the size of detected pupil shrinks to a fraction (say 20%) of its nominal size. As shown in Figure 22, an individual eye closure duration is defined as the time difference between two consecutive time instants, t_2 and t_3 , between which the pupil size is 20% or less of the maximum pupil size. And an individual eye closure speed is defined as the time period of t_1 to t_2 or t_3 to t_4 , during which pupil size is between 20% and 80% of nominal pupil size, respectively.

Computation of PERCLOS and average eye closure speed (AECS)

Individual eye closure duration and eye closure speed can reflect driver's level of alertness. Both PERCLOS and AECS at a particular time instance are computed over a fixed time interval (30 s). Average eye closure/opening speed is computed as arithmetic average of all eye closure speed over a fixed time period (30 s). A one-time measurement of either parameter, however, cannot accurately quantify a person's alertness because of its random nature. A more accurate and robust way is to compute the running average (time tracking) of each parameter. Running average computes the two parameters at a time using current data and the data at previous time instances. To obtain running average of PERCLOS measurement for example, the program continuously tracks the person's pupil size and monitors eye closure at each time instance. The cumulative eye closure duration over time is used to compute PERCLOS. We compute the PERCLOS measure based on

the percentage of eye closure in 30 s. This time limit is arbitrary and can be adjusted by the user.

Eye closure simulation result analysis

In this section, we describe the experiments we performed aimed at studying the validity of the two eyelid movement parameters. To produce realistic real data, a human subject deliberately blinks her eyes differently in front of our system to simulate different eyelid movement patterns possibly associated with different levels of fatigue. Note we are not measuring fatigue itself, rather different eyelid movement patterns. The first experiment studied the difference in eye closure speed between an alert individual and a drowsy individual. Figure 23 shows the average time needed for an alert individual to close eyes vs. the time needed for a drowsy person to close eyes. It is clear from the simulation data that the eye closure speed for a fatigue individual is much slower (longer time) than that of an vigilant individual. This reveals that eye closure speed could potentially be used as a metric to quantify the level of fatigue subject to more rigorous validation.

The second experiment lasted for 6 min, with the subject being alert during the first 2.5 min and being fatigued afterwards. The goal of this experiment is to study: (1) whether the change in fatigue level can be detected by both parameters; and (2) whether the two parameters correlated. Our system recorded the eyelid movement and computed the two parameters over the

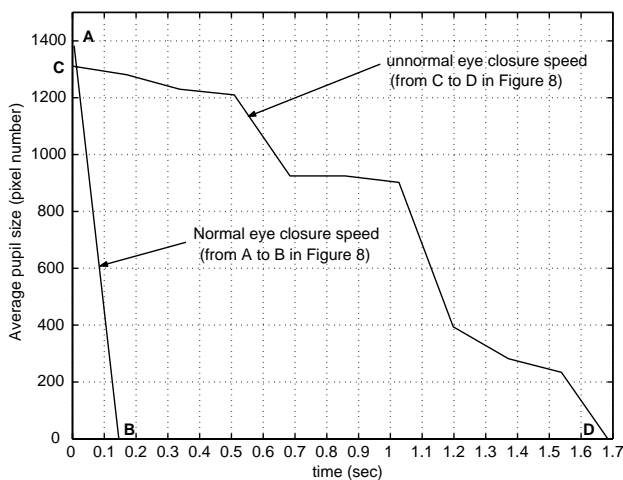


Figure 23. Eye closure speed comparison. The normal eye closure only takes 0.17 s, but abnormal eye closure takes as long as 1.68 s.

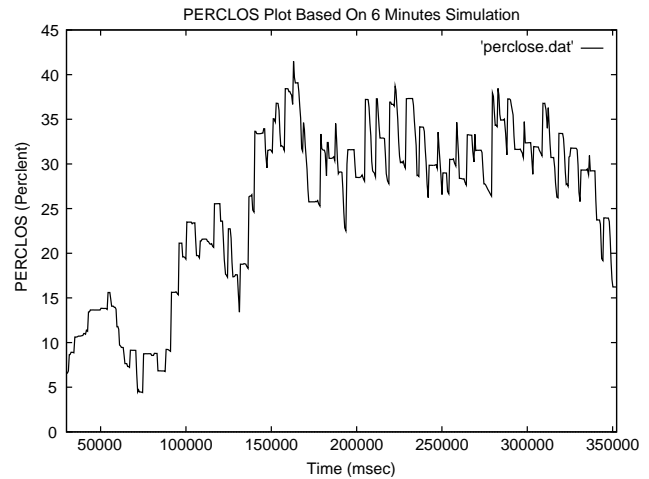


Figure 24. Running average PERCLOS measurement over 6 min.

period in real time. Figures 24 and 25 plot PERCLOS and AECS parameters over the entire 6-min period.

Figure 24 shows that in the early session of simulation (before 150,000 ms or 2.5 min), PERCLOS measurement is below 30%, which represents the alert state. However, beyond 150,000 ms, PERCLOS measures over 40%, a significant increase, representing the fatigue state. Interestingly enough, Figure 25 follows the similar trend, i.e. it takes much shorter time to close eyes when the subject is alert (before 150,000 ms) and much longer time to close eyes when the subject is drowsy (after

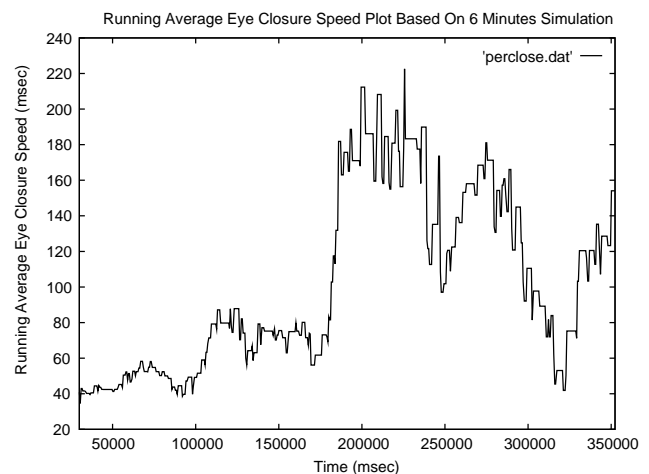


Figure 25. Running average eye closure speed measurements over 6 min.

150,000 ms). We can conclude from this experiment: (1) both parameters can detect different levels of vigilance; (2) PERCLOS and AECS covariate, which demonstrates that the two parameters correlate to certain degree. The covariation between the two parameters, however, needs be further validated using human subjects with real fatigue.

In summary, our experiments show there is a significant difference in eyelid movement measured in terms of PERCLOS and AECS for different levels of vigilance. If a driver is alert while driving, the parameters for PERCLOS and running average eye closure speed should be less than 30% and 0.5 s, respectively.

We have successfully developed a prototype hardware and software system that tracks pupils and computes PERCLOS and AECS parameters in real time. The system will issue an alarm (a repeated and loud beeping) if the computed PERCLOS and AECS measures exceed certain given threshold (e.g. $>30\%$ and >0.5 s) to alert the driver. We have tested this system with the subject's face at different orientations, distances, and substantial face movements. It works rather robustly under each situation.

This is a very significant progress in that (1) it is real time; (2) it is robust; (3) it computes the most valid fatigue measure PERCLOS, a measure recommended by the US DOT for monitoring driver fatigue [13] (4) it is non-invasive, it can be executed without the knowledge of the user due to its use of IR illumination. It can also find applications such as in human-computer interaction, assisting people with disability, and studying people's eye blink patterns.

Face (head) Orientation Estimation

Face (head) pose contains information about one's attention, gaze, and level of fatigue. Face pose determination is concerned with computation of the three dimensional (3D) face orientation and position to detect such head behaviors as head tilts. The nominal face orientation while driving is frontal. If the driver's face orientation is in other directions (e.g. down or sideway) for an extended period of time or occurs frequently (e.g. various head tilts), this is either due to fatigue or inattention. Face pose estimation, therefore, can detect both fatigue and inattentive drivers.

Methods for face pose estimation can be classified into three main categories: *model-based*, *appearance-based*, and *feature-based*. Model-based approaches typically recover the face pose by establishing the relationship between 3D face model and its two-dimensional (2D) projection [26–28]. Appearance-based approaches are based on view interpolation and their goal is to construct an association between appearance and face orientation [29–31]. Although appearance-based methods are simpler, they are expected to be less accurate than model-based approaches and are mainly used for pose discrimination. Feature-based approach determines face pose using some facial features and their image, and then determine the pose using the conventional point-based pose estimation method. The major challenge with feature-based approach is to robustly detect and track the required facial features from frame to frame under varying illumination conditions, facial expressions, and different head orientations.

In our application, the precise degree of head orientation is not necessary, what we are interested is to detect if the driver head deviates from its nominal position and orientation for an extended time or too frequently.

Face orientation determination

We propose a new model-based approach. Our approach recovers 3D face pose from a monocular view of the face with full perspective projection. Our study shows that there exists a direct correlation between 3D face pose and properties of pupils such as pupils size, inter-pupil distance, and pupils shape. Figure 26 shows pupil measurements under different head orientations. The followings are apparent from these images:

- The inter-pupil distance decreases as the face rotates away from the frontal orientation.
- The ratio between the average intensity of two pupils either increases to over one or decreases to less than one as face rotates away or rotates up/down.
- The shapes of two pupils become more elliptical as the face rotates away or rotates up/down.
- The sizes of the pupils also decrease as the face rotates away or rotates up/down.

The above observations serve as the basis for estimating face orientation from pupils.

Based on the above observations, we can develop a face pose estimation algorithm by exploiting the relationships between face orientation and these pupil

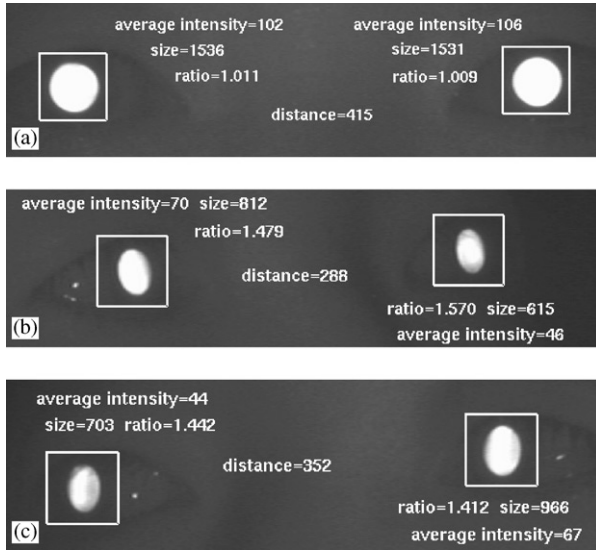


Figure 26. Pupil images for different head orientations. (a) frontal head position, (b) turn head left, (c) turn head right. It is clear that pupil properties such as size, intensity, and shape vary with face orientations.

parameters. We build a so-called *pupil feature space* (PFS) which is constructed by seven pupil features: inter-pupil distance, sizes of left and right pupils, intensities of left and right pupils, and ellipse ratios of left and right pupils. To make those features scale invariant, we further normalize those parameters by dividing over corresponding values of the front view. Figure 27 shows sample data projections in PFS, from which we can see clearly that there are five distinctive clusters corresponding to five face orientations (5 yaw angles). Note that although we can only plot 3-D space here, PFS is constructed by seven features, in which the

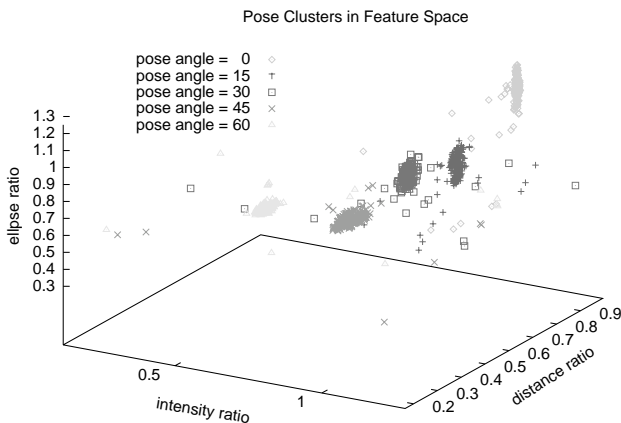


Figure 27. Face pose clusters in pupil feature space.

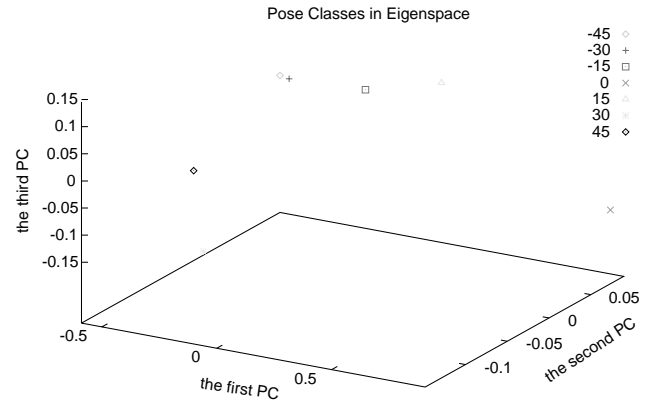


Figure 28. Projection of pose classes in eigen-PFS.

clusters will be more distinctive. So a pose can be determined by the projection of pupil properties in PFS.

To maximize the separation of different clusters, we need to find a representation of the PFS by which different pose classes are most apart from each other. A well-known method to achieve this goal is principal component algorithm (PCA), or eigenspace algorithm, which is to find the principal components of the distribution of poses, or the eigenvectors of the covariance matrix of the set of poses. The eigenvectors are ordered, each one accounting for a different amount of the variation among the poses, and each individual pose can be represented exactly in terms of a linear combination of the eigenvectors. Training data are collected to build the eigen-PFS, and store several models representing typical poses, which are, in our experiments, vary between -45° and 45° . Figure 28 shows the distribution of the models in eigen-PFS, where a 3-D projection is used while the actual dimensions of the eigen-PFS are seven. The face orientation of an input face can then be mapped to one of the clusters based on its Euclidean distance to the center of each cluster.

Experimental results

Here we present results of experiments with real data. The experiment involve a human subject rotating her face left and right to produce difference face orientations. Our face pose estimation technique is used to estimate each pose. The face orientation is quantized into seven angles: -45° , -30° , -15° , 0° , 15° , 30° , and 45° . A total of 300 data points were collected for each face orientation, representing the seven face orienta-

Table 1. Overall performance: statistics of pose estimation output. Seven feature ratios, median filtering, leave-15-out training

Ground truth	Total no. of data	No. of test data	Overall correct estimation	Correctness
-45	300	15	15	100.00
-30	300		14.95	99.67
-15	300		14.9	99.33
0	300		14.8	98.67
15	300		14.45	96.33
30	300		15	100.00
45	300		14.2	94.67

tions. Each data point is comprised of seven raw features (inter-pupil distance, sizes of left and right pupils, intensities of left and right pupils, and ellipse ratios of left and right pupils), which are collected and pre-processed through a median filtering to remove outliers. The pre-processed data are subsequently normalized by dividing each feature data by the according data of the front view. The leave- N -out technique was used to train and test the algorithm since there are not enough training data available. It works by repeatedly using part of the data for training and the other part for test. This repeats until all data have been training and testing data. In our case, 95% data are used for training and 5% for test in each iteration. The experimental results are summarized in Table 1.

To further validate the face pose estimation method, we apply it to monitor the drivers attention while driving. Figure 29 shows the running average face pose

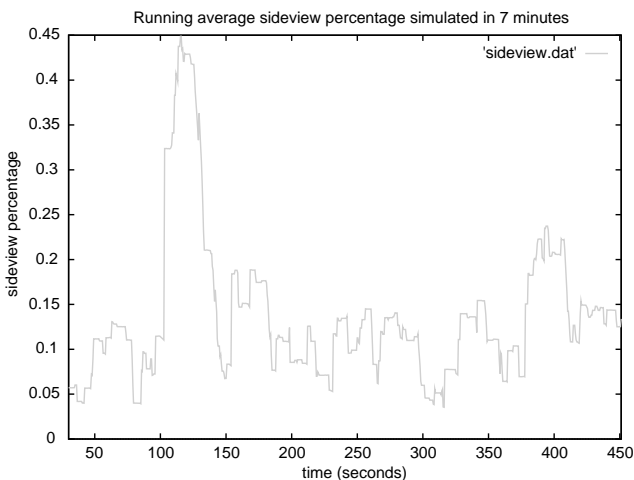


Figure 29. Face orientation monitoring over time.

estimation for a period of 6 min. As can be seen, most times during this period, face pose is frontal. But there are times when an extended period of time is spent on other directions (left or right), representing inattention. For a real-time demo of the face orientation determination, refer to

<http://www.cs.unr.edu/~qiangji/fatigue.html>.

Similar results were obtained for head rotating up and down (head tilt).

Eye-gaze Determination and Tracking

Gaze has the potential to indicate a person's level of vigilance. A fatigued individual tends to have a narrow gaze. Gaze may also reveal one's needs and attention. Gaze estimation is important not only for fatigue detection but also for identifying a person's focus of attention which can be used in the area of human-computer interaction.

Of the numerous techniques proposed for gaze estimation [31–33], the one proposed by Ebisawa [34] appears very promising and is directly applicable to this project. Their technique estimates the gaze direction based on the relative position between pupil and the glint. Based on an improvement of this technique, we have developed a video-based, contact-free eye-gaze estimation algorithm that can: (1) track and detect the eye position from the face image in real time; (2) estimate the gaze direction by computing the Cartesian coordinates difference of the glint center and the pupil center; (3) map the pixel coordinates difference into gaze.

Pupil and glint detection and tracking

The gaze estimation algorithm consists of three parts: pupil–glint detection and tracking, calibration, and gaze mapping. For this research, the gaze of a driver can be quantized into nine areas: frontal, left, right, up, down, upper left, upper right, lower left and lower right, as shown in Figure 30. Gaze estimation starts with pupil–glint detection and tracking. For gaze estimation, we continue using the IR illuminator as shown in Figure 5. To produce the desired pupil effects, the two rings are turned on and off alternately via the micro-controller described in the Pupil detection section to produce the so-called bright and dark pupil effects as shown in Figure 7(a) and (b). The pupil looks bright when the

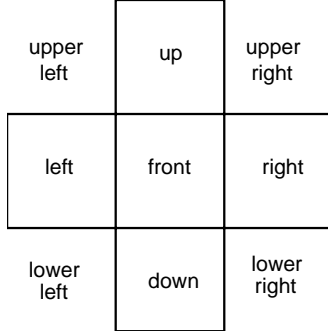


Figure 30. Quantized eye gaze regions.

inner ring is turned on as shown in Figure 7(a) and the pupil looks dark when the outer ring is turned on as shown in Figure 7(b). Note glint appears on both images. Algorithmwise, glint can be detected much more easily from the dark image since both glint and pupil appear equally bright and sometimes overlap on the bright pupil image. This explains why we need both the dark and bright pupil images. Unlike pupil tracking, gaze tracking requires a close-up view of the eyes to ensure accuracy and robustness in glint and pupil detection and the accuracy of the gaze estimate. The field of the view of camera, therefore, focuses on the region of the face near the eyes.

Given a bright pupil image, the pupil detection and tracking technique described earlier can be directly applied. The location of a pupil at each frame is characterized by its centroid. Only one pupil needs to be tracked since both pupils give the same gaze direction. Given the dark pupil image, the pupil detection and tracking technique can be adapted to detect and track glint. The center of glint can be computed and is used to specify the location of the glint. Figure 31 gives an example of bright pupils (a); dark pupils with glint (b); and the detected pupil and glint (c).

Gaze mapping

Given the relative position between pupil and glint, the screen (actual) coordinates of the gaze can be determined via a linear mapping procedure. The conventional approach for gaze mapping only uses coordinates displacement of pupil center and glint position [35, 36] as a pupil–glint vector. The main drawback with this method is that the subject must keep his or her head stationary, or the glint position in the image will change. In practice, it is difficult to keep head still and the

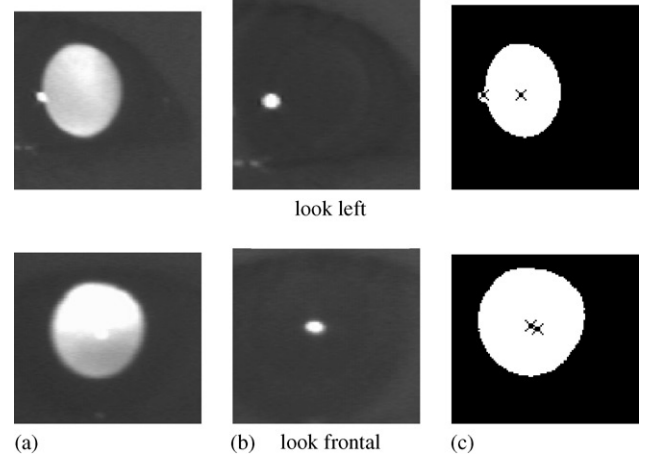


Figure 31. Relative spatial relationship between glint and bright pupil center used to determine eye-gaze position: (a) bright pupil images, (b) glint images; (c) pupil–glint relationship generated by superimposing glint to the thresholded bright pupil images. The relative positions between glint and pupil determines gaze.

existing gaze tracking methods will produce incorrect result if the head moves, even slightly. Head movement must, therefore, be incorporated in the gaze estimation procedure. In this section, we introduce a new gaze estimation procedure that tolerates slight translational head movement.

According to our mapping procedure, the pupil–glint vector is represented by

$$\mathbf{g} = [\Delta x \ \Delta y \ g_x \ g_y \ 1]^T$$

where Δx and Δy are the pupil–glint displacement, g_x and g_y are the glint image coordinates. Unlike the existing methods which only uses Δx and Δy , our procedure also includes the glint position. This effectively reduces the head movement influence. The coefficient vector \mathbf{c} is represented by

$$\mathbf{c} = [\alpha \ \beta \ \gamma \ \lambda \ \theta]^T$$

Assuming the gaze point is located at one of the nine locations on the screen as illustrated in Figure 30. The pupil–glint vector measured during runtime can be mapped to the image screen locations through the following equations:

$$i = \mathbf{g} \cdot \mathbf{c} = \alpha \Delta x + \beta \Delta y + \gamma g_x + \lambda g_y + \theta$$

where i is the gaze region index from 1 to 9 representing one of nine directions. The coefficients α , β , γ , λ , and θ

are determined via a simple calibration procedure as discussed below.

Calibration procedure. The calibration procedure is very simple and brief. We divided calibration panel of into nine regions to represent the nine gaze directions corresponding to Figure 30. During calibration, the user is asked to fixate his/her gaze from target 1 to 27. After each target is finished, the system will sound a beeper to remind user to move his gaze to next target. On each fixation, three sets of pupil-glint vectors are computed and stored, so that a 27×5 pupil-glnt matrix \mathbf{A} is obtained. The transformation from pupil-glnt matrix \mathbf{A} to the target vector \mathbf{B} is given by

$$\mathbf{A} \cdot \mathbf{c} = \mathbf{B} \quad (8)$$

where \mathbf{A} is 27×5 matrix represented as

$$\mathbf{A} = \begin{bmatrix} \Delta x_1 & \Delta y_1 & gx_1 & gy_1 & 1 \\ \Delta x_2 & \Delta y_2 & gx_2 & gy_2 & 1 \\ \Delta x_3 & \Delta y_3 & gx_3 & gy_3 & 1 \\ \Delta x_4 & \Delta y_4 & gx_4 & gy_4 & 1 \\ \Delta x_5 & \Delta y_5 & gx_5 & gy_5 & 1 \\ \Delta x_6 & \Delta y_6 & gx_6 & gy_6 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_{25} & \Delta y_{25} & gx_{25} & gy_{25} & 1 \\ \Delta x_{26} & \Delta y_{26} & gx_{26} & gy_{26} & 1 \\ \Delta x_{27} & \Delta y_{27} & gx_{27} & gy_{27} & 1 \end{bmatrix}$$

and \mathbf{B} is 27×1 vector represented by

$$\mathbf{B} = [1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 3 \ \dots \ 9 \ 9 \ 9]^T$$

Coefficient vector \mathbf{c} can be acquired by solving Eqn (8) using the least-squares method.

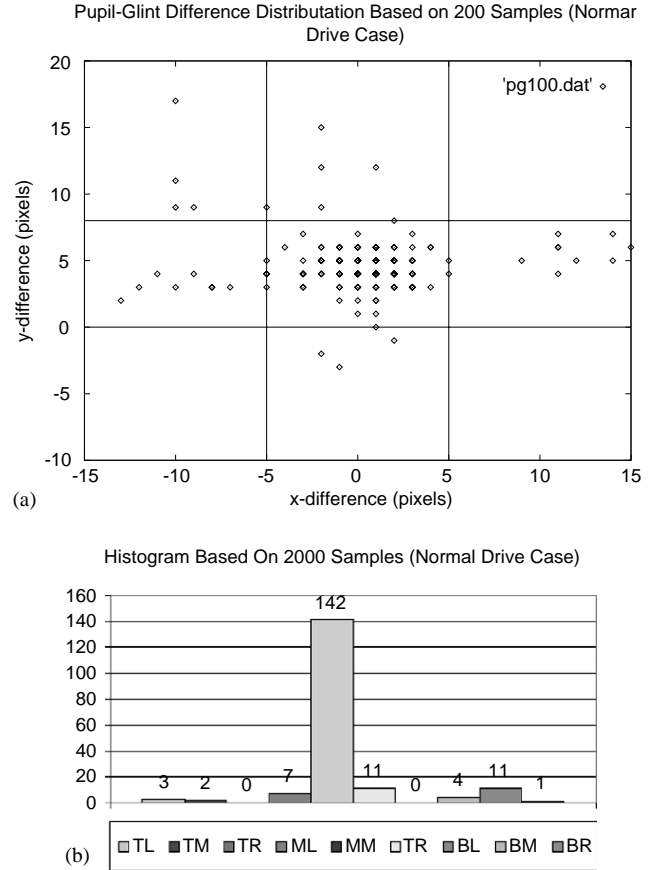


Figure 32. Normal driving case: Look front. (a) Gaze distribution under normal driving. Gaze most times is frontal. (b) Histogram plot for normal driving.

Experimental Results and Analysis

The gaze tracker is currently running on Sun Ultra10 (300 MHz) in near real time (15 frames/s). In operation, the subject faces directly to camera and changes his or her gaze direction after finishing calibration. Table 2 is

Table 2 Confusion table for gaze estimation with 100 gaze samples

Ground truth (target no.)	Estimation result (mapping target no.)									Correctness rate (%)
	1	2	3	4	5	6	7	8	9	
1	95	1	1	0	0	0	0	1	2	95
2	7	84	5	1	3	0	0	0	0	84
3	0	3	90	0	0	5	0	2	0	90
4	0	0	1	98	0	0	1	0	0	98
5	1	0	0	0	94	0	1	4	0	94
6	0	0	1	0	6	93	0	0	0	93
7	0	0	0	0	0	1	98	1	0	98
8	0	0	0	0	0	0	0	100	0	100
9	1	0	0	0	0	1	0	1	97	97

the confusion table showing the correction rate for each direction based on 100 samples. The result is very promising. Another experiment was conducted to simulate different driving state: normal, drowsy and inattention. Two hundred data samples were taken for each case. The gaze distribution and its histogram plots are shown in Figures 32–34. Figure 32 shows most gaze points are located in region 5, which reflects normal driving case. Figure 34 shows most gaze points are located in region 8, which reflects drowsy driving case. Figure 33 shows most gaze points are distributed in regions 4–6, which reflects inattentive driving case.

Conclusions

Through research presented in this paper, we developed a non-intrusive prototype computer vision system for

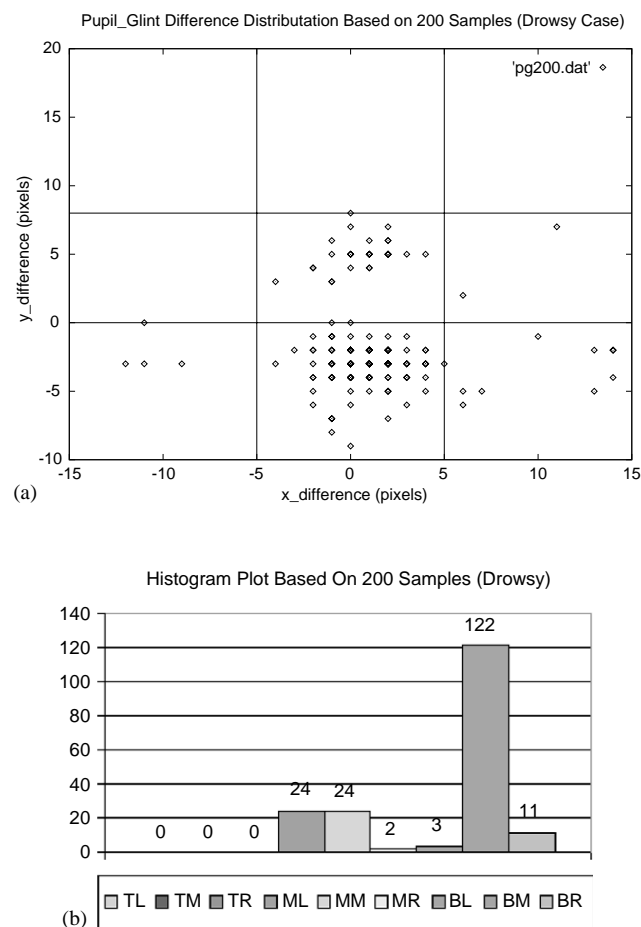


Figure 33. Drowsy: look down. (a) Gaze distribution for fatigued driving. Most gaze is down. (b) Histogram plot for fatigued driving.

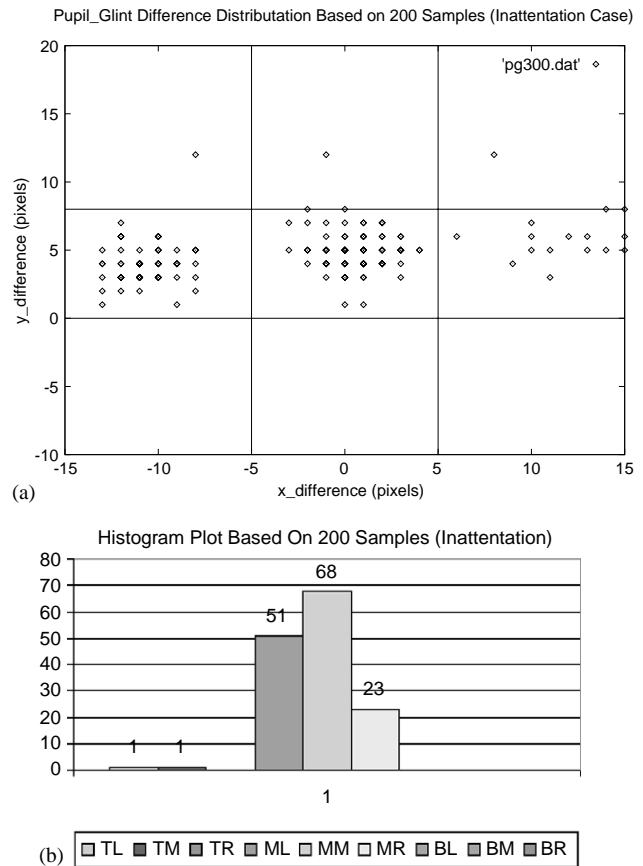


Figure 34. Inattention: look left or right. (a) Gaze distribution for inattentive driving. Significant gaze are toward left and right. (b) Histogram plot for inattentive driving.

real-time monitoring a driver's vigilance. We focus on developing the necessary hardware and imaging algorithms that can simultaneously extract multiple visual cues that typically characterize a person's level of fatigue. These visual cues include eyelid movement, gaze, and face orientation. The main components of the system consists of a hardware system for real time acquisition of video images of the driver and various computer vision algorithms and their software implementations for real-time eye tracking, eyelid movement parameters computation, face pose discrimination, and gaze estimation.

Each part of our fatigue monitor system was tested in a simulating environment with subjects of different ethnic backgrounds, different genders, ages, and under different illumination conditions. The system was found very robust, reliable and accurate. We are now collaborating with Honda to port the codes to PC and to install it system on a vehicle to evaluate its

performance under real driving conditions. Another future task is to develop new gaze estimation algorithm to tolerate large head movement. This requires improving the calibration procedure and to incorporate face pose into gaze estimation to compensate for head movement.

With our active hardware system, we can achieve the followings that are difficult to achieve with the conventional passive appearance-based methods:

- fast and real-time eye and face tracking;
- less sensitive to external illumination interference;
- more robust and accurate;
- allow fast head/face movement.

Acknowledgements

The research is partially supported by a grant from the Air Force Office of Scientific Research and a grant from Honda R & D Americas.

References

1. Saito, H., Ishiwaka, T., Sakata, M. & Okabayashi, S. (1994) Applications of driver's line of sight to automobiles – what can driver's eye tell. *Proceedings of 1994 Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, August 1994, pp. 21–26.
2. Ueno, H., Kaneda, M. & Tsukino, M. (1994) Development of drowsiness detection system. *Proceedings of 1994 Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, August 1994, pp. 15–20.
3. Boverie, S., Leqellec, J.M. & Hirl, A. (1998) Intelligent systems for video monitoring of vehicle cockpit. *1998 International Congress and Exposition ITS: Advanced Controls and Vehicle Navigation Systems*, pp. 1–5.
4. Kaneda, M. *et al.* (1994) Development of a drowsiness warning system. *The 11th International Conference on Enhanced Safety of Vehicle*, Munich.
5. Onken, R. (1994) Daisy, an adaptive knowledge-based driver monitoring and warning system. *Proceedings of 1994 Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, August 1994, pp. 3–10.
6. Feraric, J., Kopf, M. & Onken, R. (1992) Statistical versus neural net approach for driver behavior description and adaptive warning. *The 11th European Annual Manual*, pp. 429–436.
7. Ishii, T., Hirose, M. & Iwata, H. (1987) Automatic recognition of driver's facial expression by image analysis. *Journal of the Society of Automotive Engineers of Japan*, **41**: 1398–1403.
8. Yamamoto, K. & Higuchi, S. (1992) Development of a drowsiness warning system. *Journal of the Society of Automotive Engineers of Japan*, **46**: 127–133.
9. Smith, P., Shah, M. & da Vitoria Lobo, N. (2000) Monitoring head/eye motion for driver alertness with one camera. *The 15th International Conference on Pattern Recognition*, Vol. 4, pp. 636–642.
10. Saito, S. (1992) Does fatigue exist in a quantitative of eye movement? *Ergonomics*, **35**: 607–615.
11. Anon. (1999) Perclos and eyetracking: Challenge and Opportunity. *Technical Report*, Applied Science Laboratories, Bedford, MA.
12. Wierville, W.W. (1994) Overview of research on driver drowsiness definition and driver drowsiness detection. *ESV*, Munich.
13. Dinges, D.F., Mallis, M., Maislin, G. & Powell, J.W. (1998) Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management. *Department of Transportation Highway Safety Publication* 808 762, April 1998.
14. Anon. (1998) Proximity array sensing system: head position monitor/metric. *Advanced Safety Concepts, Inc., Santa Fe*, NM87504.
15. Anon. (1999) *Conference on Ocular Measures of Driver Alertness*, Washington, DC, April, 1999.
16. Grace R. (1999) A drowsy driver detection system for heavy vehicles. *Conference on Ocular Measures of Driver Alertness*, April 1999.
17. Cleveland, D. (1999) Unobtrusive eyelid closure and visual of regard measurement system. *Conference on Ocular Measures of Driver Alertness*, April 1999.
18. Fukuda, J., Adachi, K., Nishida, M. & Akutsu, E. (1995) Development of driver's drowsiness detection technology. *Toyota Technical Review*, **45**: 34–40.
19. Richardson, J.H. (1995) The development of a driver alertness monitoring system. In: Hartley, L. (ed.) *Fatigue and Driving: Driver Impairment, Driver Fatigue and Driver Simulation*. London: Taylor & Francis.
20. Hutchinson, T.E. (1990) Eye movement detection with improved calibration and speed. *U.S. Patent* 4950069, April.
21. Morimoto, C.H., Koons, D., Amir, A. & Flickner, M. (2000) Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, **18**: 331–335.
22. Kittler, J., Illingworth, J. & Foglein, J. (1985) Threshold selection based on simple image statistic. *Computer Vision, Graphics, and Image Processing*, **30**: 125–147.
23. Blake, A., Curwen, R. & Zisserman, A. (1993) A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision*, **11**: 127–145.
24. Maybeck, P.S. (1979) *Stochastic Models, Estimation and Control*, Vol. 1. New York: Academic Press, Inc.
25. Wierville, W.W. & Ellsworth, L.A. *et al.* (1994) Research on vehicle based driver status/performance monitoring: development, validation, and refinement of algorithms for detection of driver drowsiness. *National Highway Traffic Safety Administration Final Report*: DOT HS 808 247.
26. Tsukamoto, A., Lee, C. & Tsuji, S. (1994) Detection and pose estimation of human face with synthesized image models. *ICPR-94*, pp. 754–757.
27. Gee, A. & Cipoll, R. (1994) Determining the gaze of faces in images. *Image and Vision Computing*, **30**: 639–647.
28. Horprasert, T., Yacoob, Y. & Davis, L. (1996). Computing 3D head orientation from a monocular image.

- Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 242–247.
29. Gong, S., Ong, E. & McKenna, S. (1998) Learning to associate faces across views in vector space of similarities of prototypes. *Proceedings of British Machine Vision*. Southampton, UK, 1998, 300–305.
 30. Pentland, A., Moghaddam, B. & Starner, T. (1994) View-based and modular eigenspaces for face recognition. *CVPR-94*, pp. 84–91.
 31. Rae, R. & Ritter, H. (1998) Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks*, **9**: 257–265.
 32. Baluja, S. & Pomerleau, D. (1994) Non-intrusive gaze tracking using artificial neural networks. *Technical Report CMU-CS-94-102*, Carnegie Mellon University.
 33. Yang, G. & Waibel, A. (1996) A real-time face tracker. *Workshop on Applications of Computer Vision*, pp. 142–147.
 34. Ebisawa, Y. (1989) Unconstrained pupil detection technique using two light sources and the image difference method. *Visualization and Intelligent Design in Engineering*, pp. 79–89.
 35. Hutchinson, T.E. (1988) Eye movement detection with improved calibration and speed. *United States Patent* [19], (4,950,069).
 36. Hutchinson, T.E., Preston White, K., Worthy, J.R., Martin, N., Kelly, C., Lisa, R. & Frey, A. (1989) Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, man, and Cybernetics*, **19**: 1527–1533.