Real Time Visual Cues Extraction for Monitoring Driver Vigilance

Qiang Ji[1] and Xiaojie Yang[2]

[1] Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
[2] Department of Computer Science
University of Nevada at Reno

Abstract. This paper describes a real-time prototype computer vision system for monitoring driver vigilance. The main components of the system consists of 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 real time eye tracking, eyelid movement monitoring, face pose discrimination, and gaze estimation. Specific contributions include the development of an infrared illuminator to produce the desired bright/dark pupil effect, the development a digital circuitry to perform real time image subtraction, and the development of numerous real time computer vision algorithms for eye tracking, face orientation discrimination, and gaze tracking. The system was tested extensively 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.

1 Introduction

The ever-increasing number of traffic accidents in the U.S. 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. 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 accident prevention.

People in fatigue exhibit certain visual behaviors easily observable from changes in facial features like the eyes, head, and face. To make use of these visual cues, an 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. Several studies have shown the feasibility and promise of this approach [9], [13], [3], [14]. For example, study by Ueno et al [14] showed that the performance of their system is comparable to those of techniques using physiological signals. The 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. It is therefore important to simultaneously use multiple visual cues to improve the detection accuracy and robustness.

We develop a real-time prototype computer vision system that can simultaneously extract multiple visual cues in real time to characterize one's level of vigilance. Specifically, our system monitors in real time, from video images of the subject's face obtained from a remotely installed camera, certain visual behaviors that typically characterize a person's level of fatigue. The specific visual behaviors we monitor include eyelid movements, face orientation, and gaze.

Figure 1 gives an overview of our visual cues extraction system for driver fatigue monitoring. The system starts with an initialization to perform certain calibration and to compute some nominal parameters. The system then performs pupil detection and tracking, which is then used for eyelid movement monitoring, gaze estimation, and face orientation determination respectively.

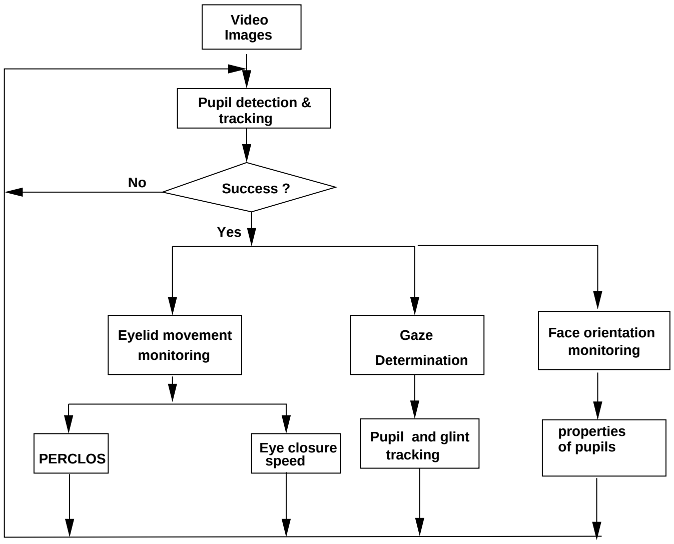


Fig. 1. Overview of the driver vigilance monitoring system

2 Image Acquisition

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 meet these requirements, we built a special infrared (IR) illuminator and choose a IR-sensitive CCD camera for image acquisition. According to the original patent from Hutchinson [7], the use of IR illuminator allows to produce the bright-pupil effect, which will be used for pupil detection and tracking.

Our IR illuminator consists two sets of IR LEDs, distributed evenly around two concentric rings (8LED's in each ring) as shown in Figure 2. The center of both rings coincides with the camera optical axis. Both rings are in the same plane and mounted on the front of camera.

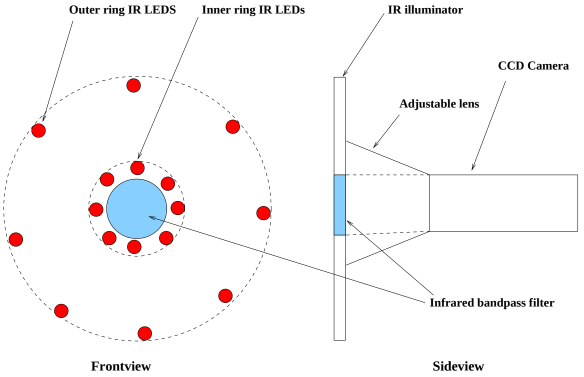


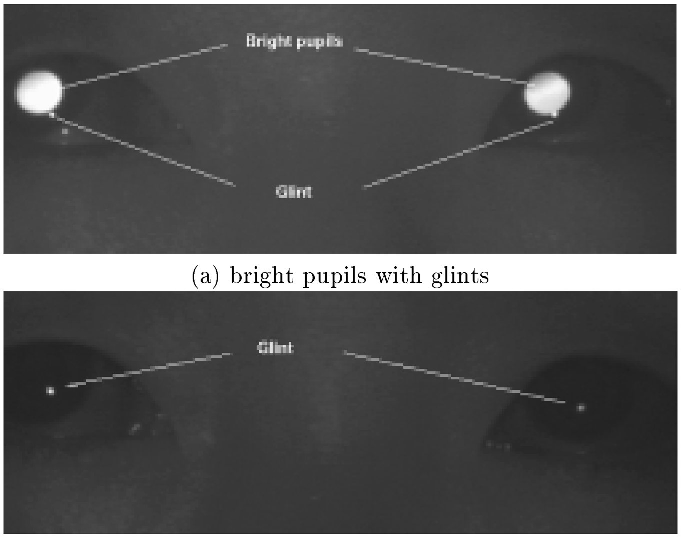
Fig. 2. IR light Source Configuration

The IR light source illuminates the user's eye and generates two kinds of pupil images: bright and dark pupil images as shown in Figure 3. 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 [3] appears on both the dark and bright pupil images. Figure 4 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 Sun glasses.

3 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. Figure 5 gives an overview of the pupil tracking system. Pupil tracking can be divided into two stages: pupil detection and pupil tracking as discussed below.

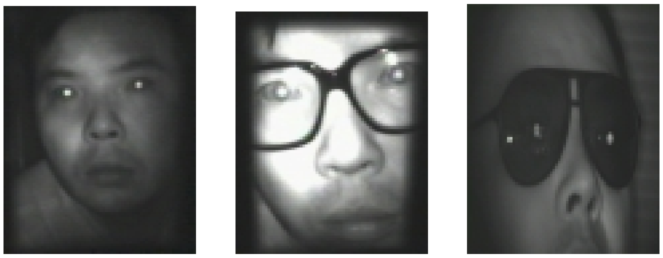
[3] the small bright spot near the pupil, produced by corneal reflection of the IR illuminator.



(a) bright pupils with glints

(b) dark pupils with glints

Fig. 3. Bright and dark pupil image



(a) (b) (c)

Fig. 4. Examples of the acquired images with the desired bright pupil effect. (a) without glasses, (b) with glasses; (c) with Sun glasses.

3.1 Pupil Detection

Pupil detection involves locating pupils in the image. The detection algorithm starts with an initialization to obtain the background image and to record the nominal pupil size and intensity. Following the initialization, a preprocessing is applied to minimize interference from illumination sources other than IR illuminator. This includes Sun light and ambient light interference. Figure 6 shows an image where parts of the background look very bright, almost as bright as the pupil. This must be eliminated or they may adversely affect pupil detection. Their removal is accomplished by subtracting the image with only ambient light from the one illuminated by both the infrared illuminator and the ambient light. The resultant image, as shown in Figure 6 contains the illumination effect from

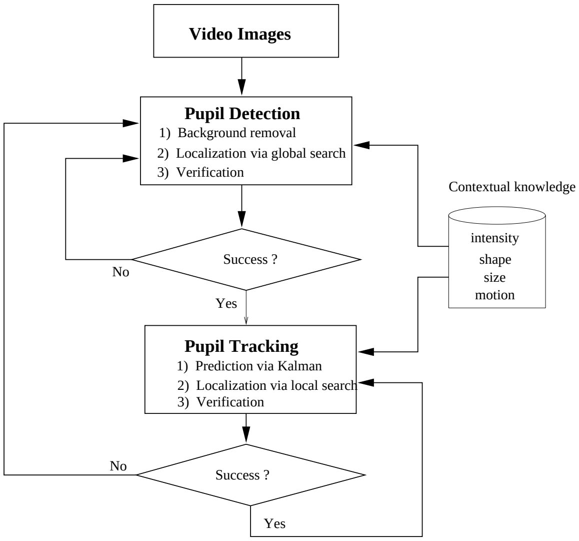


Fig. 5. Pupil detection and tracking system flowchart

only the infrared illuminator. A micro-controller with a video decoder has been built to perform real image subtraction. The controller separates each incoming interlaced image frame from the camera into even and odd fields and alternately turns the inner ring on for the even field and off for the odd field. The difference image is then produced by subtracting the odd field from the even field. The images shown in Figure 6 are such produced.

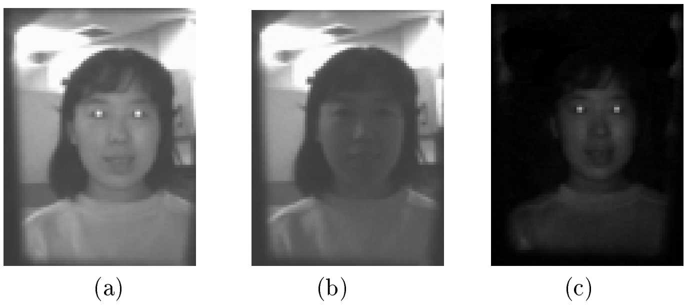
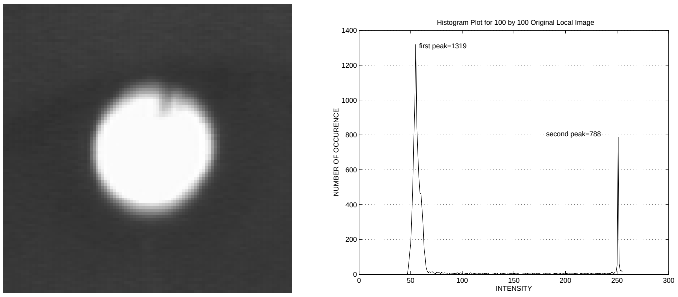


Fig. 6. Background Removal via Image Subtraction: (a) the image obtained with both ambient and IR light (even field); (b) the image obtained with only ambient light (odd field); and (c) the image resulted from subtraction (b) from (a)

3.2 Determining the Initial Pupils Positions

Given the image resulted from the background removal procedure, pupils may be detected by searching the 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 as shown in Figure 7 (a) and its intensity histogram in (b), where there are two distinctive peaks, one representing background and the other representing pupil. The two peaks must also be sufficiently apart from each other.



(a) (b)

Fig. 7. An original subimage containing a pupil (a); and its histogram plot (b), which follows a bimodal distribution.

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 [10]. 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 8)) 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.

3.3 Pupil Tracking via Kalman Filtering

Given the detected pupils in the initial frames, pupils can then be tracked from frame to frame in real time. Tracking can be done more efficiently by using the location of the pupil in previous frames to predict the location of the face in

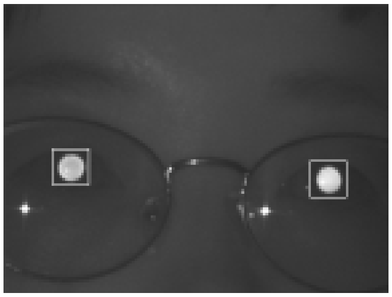


Fig. 8. The glares on the eye frame are equally as bright as the pupils. Verification can eliminate them from being considered as pupils.

future frames based on Kalman filtering [2], assuming that the person's pupil will not undergo significant locational change in two consecutive frames.

Briefly, 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 motion of a pupil at each time instance (frame) can be characterized by its position and velocity. Let (c_t, r_t) represent the pupil pixel position (its centroid) at time t and (u_t, v_t) be its velocity at time t in c and r directions. The state vector at time t can therefore be represented as $x_t = (c_t\ r_t\ u_t\ v_t)^t$.

The system can therefore be modeled as

$$\mathbf{x}_{t+1} = \phi \mathbf{x}_t + \mathbf{w_t} \tag{1}$$

where $\mathbf{w_t}$ represents system perturbation.

If we assume pupil movement between two consecutive frames is small enough to consider the motion of pupil positions from frame to frame 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 extractor estimates $\mathbf{z}_t = (\hat{c}_t, \hat{r}_t)$, the pupil position at time t. Therefore, the measurement model in the form needed by the Kalman filter is

$$\mathbf{z}_t = H\mathbf{x}_t + v_t \tag{2}$$

For simplicity and since z_t only involves position, we have

$$H = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{bmatrix}$$

and v_t represents measurement uncertainty. Specifically, the position of current frame t, $z_t = (c_t, r_t)$, is estimated as

$$\hat{c}_t = c_{t-1} + (c_{t-1} - c_{t-2}) \tag{3}$$
$$\hat{r}_t = r_{t-1} + (r_{t-1} - r_{t-2}) \tag{4}$$

where (c_{t-1}, r_{t-1}) and (c_{t-2}, r_{t-2}) are the actual pupil pixel coordinates estimated at frames t-1 and t-2.

Given the state model in equation 1 and measurement model in equation 2 as well as some initial conditions, the state vector x_{t+1}, along with its covariance matrix Σ_{t+1}, can be updated from the measurement images using Kalman filter [11]. While x_{t+1} gives predicted pupil position, Σ_{t+1} gives the uncertainty of the estimated pupil position, which determines local search area. Figure 9 graphically illustrates the principle of Kalman filtering.

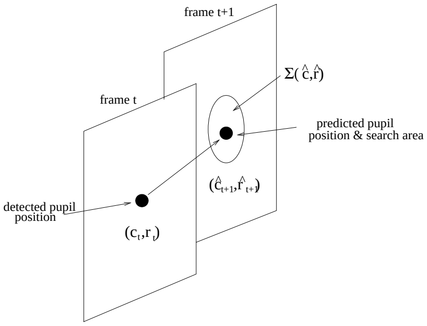

frame t+1
frame t
$\Sigma(\hat{c},\hat{r})$
predicted pupil position & search area
$(\hat{c}_{t+1},\hat{r}_{t+1})$
detected pupil position
(c_t,r_t)

Fig. 9. Pupil detection and tracking using Kalman filtering

3.4 Tracking Results

The Kalman filter tracker has been implemented and tested. A tracking speed as high as 15-18 frames per second for an image size of 640 × 480 is achieved with a SUN 300 MHZ workstation (Ultra 30). The pupil detection and tracking software is found to be rather robust under different face orientations and distances and can quickly 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 It also works well for pupil with glasses. For a real time video demo of our pupil tracking system, please refer to *http://www.cs.unr.edu/~ qiangji/fatigue.html.* Figure 10 shows the trajectory of the real and estimated pupil position in 30 consecutive frames using Kalman filter.

4 Eyelid Movement Parameters

The primary purpose of pupils tracking is to monitor eyelid movements and to compute two ocular parameters: PERCLOS and AECS. PERCLOS measures

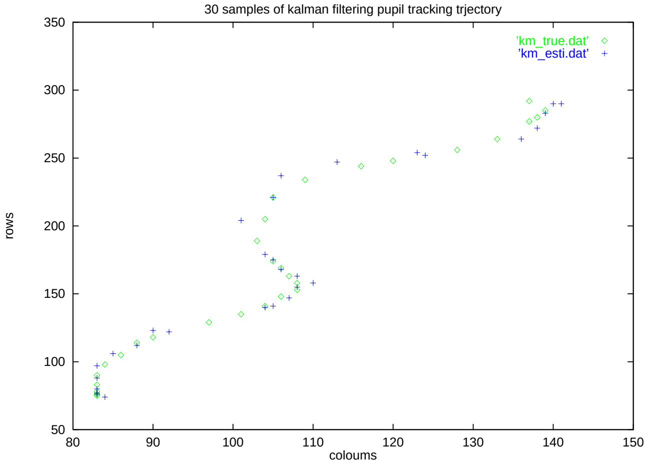
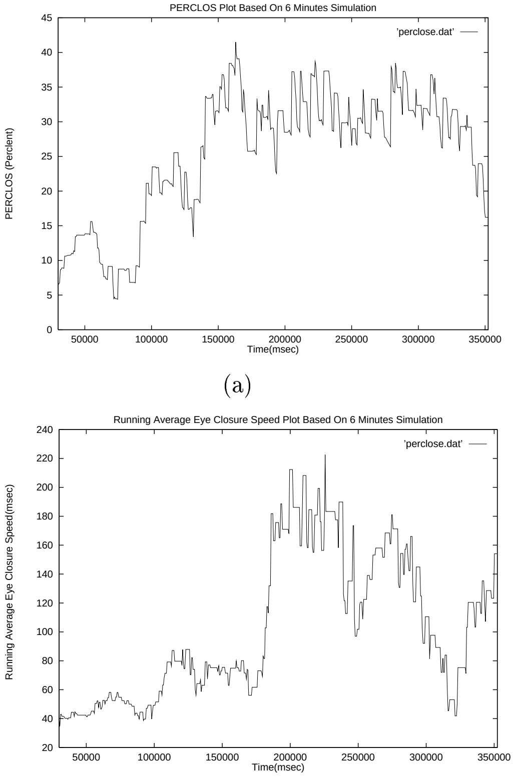


Fig. 10. Trajectory of the real and estimated pupil position in 30 consecutive frames. Crosses indicate the pupil positions estimated by Kalman filter. Small circles indicate the actual tracked pupil positions. It's apparent that the two pupil positions match very well.

percentage of eye closure over time. It has been found to be the most valid ocular parameter for characterizing driver fatigue [4]. AECS measures the average eye closure/opening speed, i.e., the amount of time needed to fully close the eyes and to fully open the eyes. To obtain these measurements, our algorithm continuously tracks the subject's pupils and determines in real time the amount of eye closure based on the the area of the pupils that have been occluded by the eyelids.

To produce realistic real data, a human subject deliberately blinks her eyes differently in front of our system to simulate different levels of fatigue. The first experiment studied the difference in eye closure speed between an alert individual and drowsy individual. Our study indicates that the eye closure speed for a drowsy person can be more than eight times slower than that of an alert individual. The second experiment lasted for 6 minutes, with the subject being alert during the first 2.5 minutes and being fatigue afterwards. The goal of this experiment is to study: 1) whether the change in fatigue level can be detected by both parameters; 2) whether the two parameters correlate. Our system recorded the eyelid movement and computed the two parameters over the period in real time. Figure 11 (a) ane (b) plot running average PERCLOSE and AECS parameters over the entire 6 minute period.

Fig. 11 (a) shows that in the early session of the simulation (before 150000 ms or 2.5 minutes), PERCLOS is measured 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, Fig. 11 (b) follows the similar trend, i.e., it takes much shorter time to close the eyes when the subject is alert (before 150,000 ms) and much longer time to close when the



(a)

(b)

Fig. 11. Running average PERCLOS (a) and AECS (b) measurements over 6 minutes

subject is drowsy (after 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 the two parameters correlate to certain degree.

In summary, we have successfully developed a prototype hardware and software system that tracks pupils and computes PERCLOS and AECS parameters in real time. 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, a measure recommended by the US Department of Transportation for monitoring driver fatigue; 4) it is non-invasive, it can be executed without the knowledge of the user due to its use of infrared illumination. For a real time demo, please refer to http://www.cs.unr.edu/~qiangji/fatigue.html.

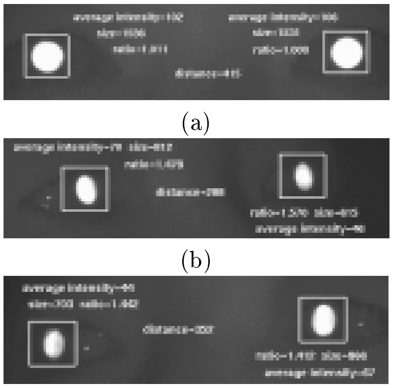
5 Face (head) Pose Estimation

Face pose determination is concerned with computation of 3D face orientation and position. Face pose contains information about one's attention, gaze, and level of fatigue. 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, this is either due to fatigue or inattention. Face pose estimation, therefore, can detect both fatigue and inattentive drivers.

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, pupils shape, and average pupil intensities. Figure 12 shows pupil measurements under different head orientations. It is apparent from these images that

- 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.
- The shapes of two pupils become more elliptical as the face rotates away.
- The sizes of the pupils also decrease as the face rotates away.

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



(a)

(b)

(c)

Fig. 12. Pupil images for different head orientations: (a) frontal; (b) left; and (c) right

5.1 Algorithm for Head Orientation Monitoring

Based on the above observations, we can develop a face pose estimation algorithm by exploiting the relationships between face orientation and these 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 according values of the front view. Figure 13 shows sample data projections in PFS, from which we can see clearly that there are distinctive clusters of different poses in the PFS. Note that although we can only plot 3-dimensional space here, PFS is constructed by seven features, in which the clusters will be more distinctive. So a pose can be determined by the projection of pupil properties in PFS.

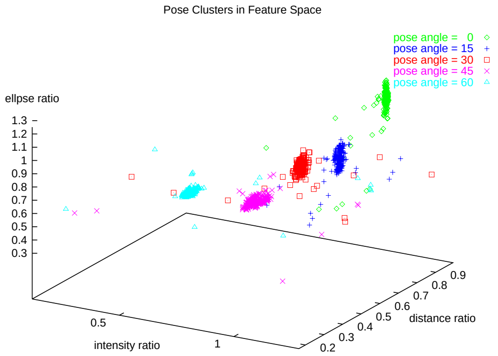


Fig. 13. Face pose clusters in pupil feature space

The five clusters correspond to five face orientations respectively. It is clear that there exist five distinctive clusters, representing each of five face orientations. Given sufficient training data, we are able to identify five distinctive clusters.

When determining pose by the projection in PFS, we need to find a representation of the PFS by which different pose classes are the most apart from each other. A well known method to achieve this goal is principal component algorithm (PCA), or eigen space 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.

Before pose estimation, we need training data to build the eigen PFS, and store several models representing typical poses, which are, in our experiments, varies every 15° from −45° to 45°. Figure 14 shows the distribution of the models in eigen PFS, where again, a 3-dimensional projection is used while the actual dimensions of the eigen PFS is 7. The face orientation of an input face can then be mapped to one of the seven clusters based on its Euclidean distance to the center of each cluster.

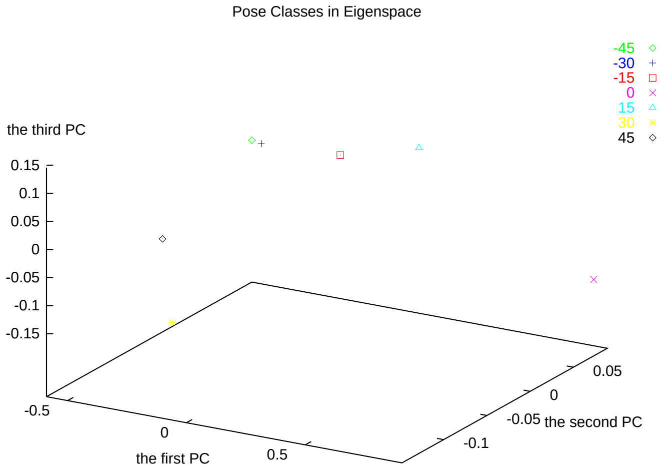


Fig. 14. Projection of pose classes in eigen PFS

We developed a software based on the above idea to determine where the heads turn to and will warn the driver by sounding alarm if certain threshold is exceeded. Figure 15 shows the running average face pose estimation for a period of 6 minutes. 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.

6 Eye-gaze Determination and Tracking

Gaze has the potential to indicate a person's level of vigilance. A fatigue 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 [1] [15], [12], the one proposed by Ebisawa [5] 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;

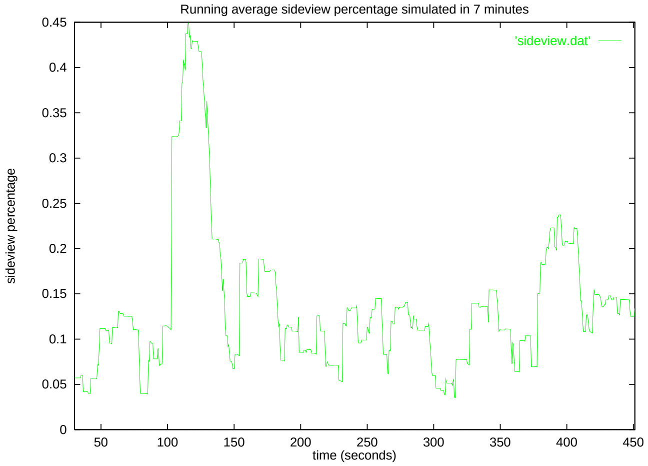


Fig. 15. Face Orientation Monitoring Over Time

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 screen coordinates (mm).

6.1 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 16. Gaze estimation starts with pupil-glint detection and tracking. For gaze estimation, we continue using the IR illuminator as shown in Figure 2. To produce the desired pupil effects, the two rings are turned on and off alternately via a micro-controller to produce the so called bright and dark pupil effect as shown in Figure 3 (a) and (b). The pupil looks bright when the inner ring is turned on as shown in Fig. 3 (a) and the pupil looks dark when the outer ring is turned on as shown in Fig. 3 (b). Note glint appears on both images. Algorithm-wise, 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.

Given a bright pupil image, the pupil detection and tracking technique described in section 3 can be directly applied for pupil detection and tracking. 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 17 gives an example of bright pupils (a); dark pupils with glint (b); and the detected pupil and glint (c).

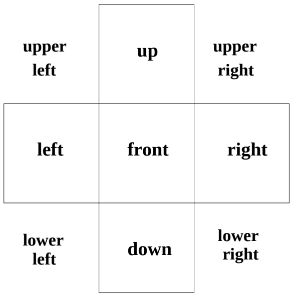
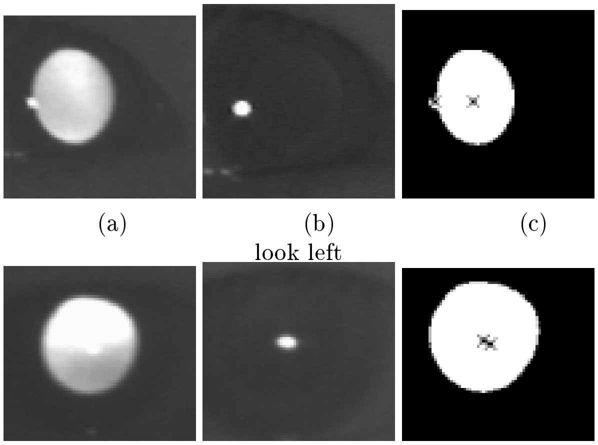


Fig. 16. The quantized eye gaze regions


(a) (b) (c)
look left

(a) (b) (c)
look frontal

Fig. 17. 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.

6.2 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 [6] [8] 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 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

$$\mathbf{g} = \left[\, \Delta x \; \Delta y \; g_x \; g_y \; 1 \,\right]$$

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

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

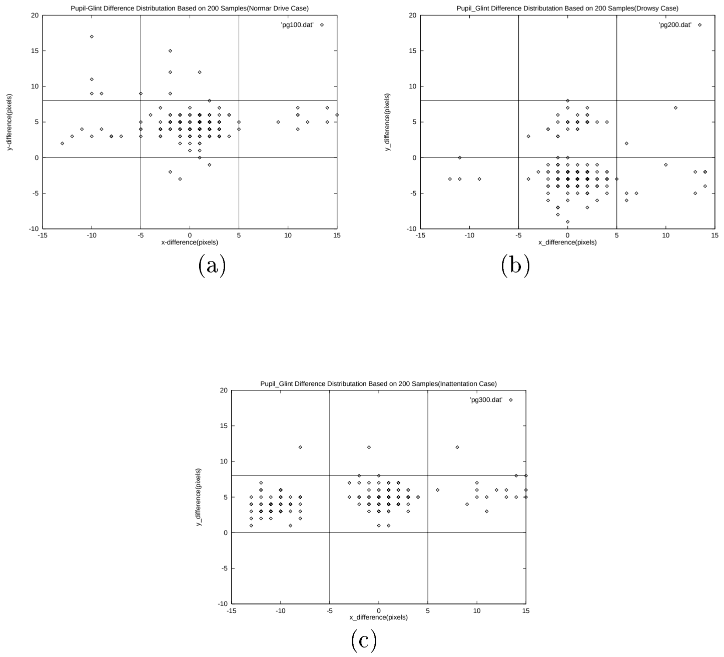
Assuming the gaze point is located at one of the nine locations on the screen as illustrated in Figure 16. The pupil-glint vector measured during runtime can be mapped to the image screen locations through following equations:

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

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.

6.3 Experimental Results and Analysis

The gaze tracker is currently running on Sun Ultra10 (300 MHz) in real time. In operation, the subject faces directly to camera and changes his or her gaze direction after finishing calibration. Our study shows that the gaze estimation procedure is accurate. Of the 100 samples representing the subject fixing on different gaze regions, the system has over 95% accuracy. Another experiment was conducted to simulate different driving state: normal, drowsy and inattention and to study gaze distribution. 200 data samples were taken for each case. The gaze distributions plots are in Figure 18. Fig. 18 (a) shows most gaze points are located in region 5 (frontal), which reflects normal driving case. Fig. 18 (b) shows most gaze points are located in region 8 (lower center), which reflects drowsy driving case. Fig. 18 (c) shows most gaze points are distributed in region 4 (middle left), 5 (frontal), and 6 (middle right), which reflects inattentive driving case. For a real-time demonstration of the gaze estimation, please refer to http://www.cs.unr.edu/~qiangji/fatigue.html.



(a)

(b)

(c)

Fig. 18. Gaze distribution (a) under normal driving. Gaze most times are frontal; (b) fatigue driving, gaze looks down most time; and (3) inattentive driving, gaze looks away sometimes

7 Conclusions

Through research presented in this paper, we developed an non-intrusive prototype computer vision system for 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 extensively 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 in a vehicle to evaluate its performance under real world driving conditions.

References

1. S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. *Technical Report CMU-CS-94-102, Carnegie Mellon University*, 1994.
2. A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. Journal of Computer Vision*, 11(2):127–145, 1993.
3. S. Boverie, J. M. Leqellec, and A. Hirl. Intelligent systems for video monitoring of vehicle cockpit. *1998 International congress and exposition ITS: Advanced controls and vehicle navigation systems*, pages 1–5, 1998.
4. D. F. Dinges, M. Mallis, G. Maislin, and J. W. Powell. 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.
5. Y. Ebisawa. Unconstrained pupil detection technique using two light sources and the image difference method. *Visualization and Intelligent Design in Engineering*, pages 79–89, 1989.
6. T. E. Hutchinson. Eye movement detection with improved calibration and speed. *United States Patent [19]*, (4,950,069), 1988.
7. T. E. Hutchinson. Eye movement detection with improved calibration and speed. *U.S. patent 4950069*, April, 1990.
8. T. E. Hutchinson, K. White, J. R. Worthy, N. Martin, C. Kelly, R. Lisa, , and A. Frey. Human-computer interaction using eye-gaze input. *IEEE Transaction on systems,man,and cybernetics*, 19(6):1527–1533, 1989.
9. T. Ishii, M. Hirose, and H. Iwata. Automatic recognition of driver's facial expression by image analysis. *Journal of JSAE*, 41(12):1398–1403, 1987.
10. J. Kittler, J. Illingworth, and J. Foglein. Threshold selection based on simple image statistic. *Computer vision, graphics, and image processing*, 30:125–147, 1985.
11. P. S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, Inc, 1979.
12. R. Rae and H. Ritter. Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks*, 9(2):257–265, 1998.
13. H. Saito, T. Ishiwaka, M. Sakata, and S. Okabayashi. 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, Aug. 1994*, pages 21–26, 1994.
14. H. Ueno, M. Kaneda, and M. Tsukino. Development of drowsiness detection system. *Proceedings of 1994 Vehicle navigation and information systems conference, Yokohama, Japan, Aug. 1994*, pages 15–20, 1994.

15. G. Yang and A. Waibel. A real-time face tracker. *Workshop on Applications of Computer Vision*, pages 142–147, 1996.

This article was processed using the LaTeX macro package with LLNCS style