Vision-based method for detecting driver drowsiness and distraction in driver monitoring system

Jaeik Jo
Sung Joo Lee
Yonsei University
School of Electrical and Electronic Engineering
134 Sinchon-dong, Seodaemun-gu
Seoul, Seoul 120-749, Republic of Korea

Ho Gi Jung
Hanyang University
School of Mechanical Engineering
222 Wangsimni-ro, Seongdong-gu
Seoul, Seoul 133-791 Republic of Korea

Kang Ryoung Park
Dongguk University
Division of Electronics and Electrical Engineering
26, Pil-dong 3-ga, Jung-gu
Seoul, Seoul 100-715, Republic of Korea

Jaihie Kim
Yonsei University
School of Electrical and Electronic Engineering
134 Sinchon-dong, Seodaemun-gu
Seoul, Seoul 120-749, Republic of Korea
E-mail: jhkim@yonsei.ac.kr

Abstract. Most driver-monitoring systems have attempted to detect either driver drowsiness or distraction, although both factors should be considered for accident prevention. Therefore, we propose a new driver-monitoring method considering both factors. We make the following contributions. First, if the driver is looking ahead, drowsiness detection is performed; otherwise, distraction detection is performed. Thus, the computational cost and eye-detection error can be reduced. Second, we propose a new eye-detection algorithm that combines adaptive boosting, adaptive template matching, and blob detection with eye validation, thereby reducing the eye-detection error and processing time significantly, which is hardly achievable using a single method. Third, to enhance eye-detection accuracy, eye validation is applied after initial eye detection, using a support vector machine based on appearance features obtained by principal component analysis (PCA) and linear discriminant analysis (LDA). Fourth, we propose a novel eye state–detection algorithm that combines appearance features obtained using PCA and LDA, with statistical features such as the sparseness and kurtosis of the histogram from the horizontal edge image of the eye. Experimental results showed that the detection accuracies of the eye region and eye states were 99 and 97%, respectively. Both driver drowsiness and distraction were detected with a success rate of 98%. [DOI: 10.1117/1.3657506]

Subject terms: driver monitoring systems; drowsy driver detection; drowsiness; distractions; inattention; blink detection; machine learning; computer vision; feature selection.

Paper 110469RR received May 2, 2011; revised manuscript received Oct. 12, 2011; accepted for publication Oct. 12, 2011; published online Dec. 2, 2011.

1 Introduction

Driver inattention is one of the major causes of highway car accidents. According to the U.S. National Highway Traffic Safety Administration (NHTSA), in the U.S. in 2007, ~6100 fatalities occurred as a result of car accidents related to driver inattention, such as distraction, fatigue, and lack of sleep.[1–3] Consequently, safe-driving assistant systems, which measure the level of driver inattention and provide a warning when a potential hazard exists, have received a great deal of attention as a measure to prevent accidents caused by driver inattention.

Generally, driver inattention relates to the degree of nonconcentration when driving; it is usually a result of drowsiness and distraction.[4] Drowsiness involves a driver closing his eyes because of fatigue, and distraction involves a driver not paying sufficient attention to the road despite the presence of obstacles or people. Previous driver-inattention monitoring systems (DIMSs) have detected driver drowsiness or distraction but not both.[5–23] Although these systems can detect drowsiness or distraction, a car accident can occur if it cannot detect both. Because driver distraction and drowsiness are the main factors in vehicle crashes, both should be considered when measuring driver-inattention level.[24] In this context, we propose a new DIMS that can detect both drowsiness and distraction. By measuring both causes of inattention, the proposed system can improve the security level of the previous DIMSs, which detected either drowsiness or distraction.

Many inattention-monitoring systems have been developed to prevent highway car accidents. These systems can be divided into two categories, as shown in Tables 1 and 2. The first is to detect driving behavior by monitoring vehicle speed, steering movements, lane keeping, acceleration, braking, and gear changing.[25] The other is to detect driver behavior, which includes two approaches, such as a visual feature–based approach[26] and a physiological feature–based approach.[27] The former is based on tracking the driver's head and eye movements and recognizing the torso and arm/leg motion. The latter measures the heart and pulse rate as well as the brain activity.

Driving-behavior information–based methods are affected by the vehicle type and the individual variation in driving behavior. Physiological feature–based approaches are intrusive because the measuring equipment must be attached to the driver. Thus, visual feature–based approaches have recently become preferred because they are nonintrusive to the driver. In this work, we also focus on the visual feature–based approach to monitor driver inattention.

A great deal of previous visual feature–based research has been studied to monitor driver inattention. These can be divided into two systems: drowsiness detection systems and distraction detection systems. The drowsiness detection system detects drowsiness using features such as eyelid movement, facial expression, yawning, nodding, etc. Many attempts to develop a drowsiness detection system have been

0091-3286/2011/$25.00 © 2011 SPIE

Table 1 Summary of previous research for detecting driver inattention.

Category		Method	Strength	Weakness
Driving behavior information[a]		Monitoring vehicle speed, steering movement, lane keeping, acceleration, braking, and gear changing	It does not require an additional camera or biosensors.	It is affected by vehicle type and the individual variation in driver driving behavior.
Driver behavior information	Physiological feature–based approaches[b]	Measuring the heart and pulse rate and brain activity	It is possible to accurately measure the inattention level	It is intrusive because measurement equipment must be attached to the driver
	Visual feature–based approaches[c]	Tracking head and eye movements, recognizing the facial expression as well as torso, arm, and leg motion	It is nonintrusive to the driver	It requires an additional camera device

[a]Reference 25.
[b]Reference 26.
[c]Reference 27.

reported in the literature.[5–17,28–31] For example, Ueno et al. described a system for drowsiness detection that recognizes whether a driver's eyes are open or closed; if open, the degree of openness is measured.[6] D'Orazio et al. introduced a system to detect driver fatigue using eyelid movement information, including new drowsiness parameters [frequency of eye closure (FEC) and eye-closure duration (ECD)].[17] The main contribution of their work was the introduction of a reliable eye-detection approach that does not impose any constraints on the driver and does not require any preprocessing to segment eye regions. They demonstrated that the performance of their system is comparable to those that utilize physiological signals. Vural et al. introduced a system to characterize a driver's state from his/her facial expression information.[28] Saradadevi and Bajaj proposed a method for monitoring driver fatigue using yawning information.[29]

Some studies detect more than one of these pieces of information and combine them to improve robustness.[30,31] However, because the aforementioned methods only detect driver drowsiness, they cannot prevent car accidents caused by distraction. This is because car accidents that result from driver distraction can occur when a driver does not look straight ahead. That is, even though the driver's eyes are not closed because of fatigue, a car accident can still occur if he or she is distracted. To develop a safer driver-monitoring system, these two risks should be monitored simultaneously.

The distraction-detection system uses head pose or gaze information to detect if a driver is paying sufficient attention to the road when obstacles or people on the road are detected.

Table 2 Summary of previous "visual feature based approaches" to monitor driver inattention.

Category	Method	Strength	Weakness
Drowsiness detection system	Eyelid movement,[a] facial expression,[b] yawning,[c] nodding,[d] more than one of the above[e]	Nonintrusive to the driver	Cannot detect the distraction of the driver
Distraction detection system	Head orientation,[f] eye orientation,[g] head and eye orientation[h]	Nonintrusive to the driver	Cannot detect the drowsiness of the driver
Hybrid method (proposed method)	This method detects both the distraction and drowsiness of the driver.	Able to detect both the drowsiness and distraction of the driver	Processing time increases slightly compared to the drowsiness detection system or the distraction detection system.

[a]References 6 and 17.
[b]Reference 28.
[c]Reference 29.
[d]Reference 64.
[e]References 30 and 31.
[f]References 18, 20, 21, and 65–68.
[g]References 22, 61, and 69.
[h]Reference 23.

Recent surveys on head orientation and gaze from image sequences can be found in Refs. 18 and 19.

Among various methods, we focused on the previous driver's gaze and head orientation methods developed for vehicular environments. For example, Hattori et al. introduced a forward warning system that employs driver behavioral information.[20] Their system determines driver distraction when it detects that the driver is not looking straight ahead. Trivedi et al. recognized driver awareness using head pose information obtained by a localized gradient orientation histogram and support vector regressors (SVRs).[21] Smith et al. analyzed global motion and color statistics to robustly track a driver's facial features.[22] Using these features, they estimated continuous gaze direction. However, this method cannot always localize facial features when the driver wears eyeglasses, makes conversation, closes his eyes, or rotates his head. It also failed to work at night. Kaminski et al. introduced a system to compute both head orientation based on a geometrical model of the human face and eye-gaze detection based on a geometrical model of the human eye.[23] They estimated continuous head orientation and gaze direction. However, the above-mentioned systems for automotive applications did not deal with the problem of driver drowsiness. That is, although the driver is looking ahead, he or she can drive while drowsy. Therefore, a method that monitors both visual distractions and drowsiness is needed for the driver-monitoring system.

Although a technical study[32] has reported an intelligent vehicle safety system that detects distraction and drowsiness in real road test conditions, such a system differs from our system in the following aspects.

First, it independently detects driver drowsiness and distraction using commercialized products, Smart Eye Pro and 3D head model, respectively. These two products are simultaneously operated in all cases. On the other hand, our system divides the driver's state into two cases, and then, if the driver is looking ahead, the system operates the drowsiness detector. However, if the driver is not looking ahead, the system operates the distraction detector. Therefore, the computational cost of the system can be decreased. The other advantage of the proposed method is that the eye-detection errors and the consequent false alarms for drowsiness are decreased in the case of large head rotation. In general, eye detection during large head rotation is difficult because the texture and shape of the eyes change markedly as a result of head rotation. Therefore, in the proposed method, eye detection is performed only in the case of drowsiness detection (when the driver is looking ahead). Both facial border lines and nose center lines are used in the case of distraction detection (when the driver is not looking ahead) without eye detection. Thus, eye-detection errors and the consequent false alarms for drowsiness can be decreased.

Second, experimental results for blink detection have not been provided in the technical study, whereas they have been included in our paper. Third, the study reports that Smart Eye Pro does not work well when the driver wears eyeglasses or sunglasses; the drivers who were tested did not wear glasses. In contrast, our system exhibits good performance regardless of the use of glasses.

Fourth, Sec. 3.3.3 of the technical study[32] states that 3-D head model estimation requires initial calibration, which causes inconvenience to the driver and requires a longer processing time. On the other hand, the proposed method does not require initial calibration. Fifth, because two 8-mm IDS uEye USB-cameras were used in that study,[32] the system was bulky and economically infeasible. In contrast, only one camera is used in the proposed system.

In order to measure both drowsiness and distraction in the proposed method, we first detect the face region and estimate the driver's head orientation to determine the gaze direction of driver. Accordingly, the proposed system determines whether the driver is looking ahead. If the estimated head orientation indicates that the driver is not looking ahead, then the system monitors the driver-distraction level and sounds an alarm when the level is dangerously high. If the estimated head orientation indicates that the driver is looking ahead, then the system detects the driver's eyes to determine the drowsiness level. In this case, the system focuses solely on driver drowsiness. The driver-drowsiness level is measured as PERCLOS, which is the percentage of eye closure time during a certain time interval.[12] Similarly, the distraction level is measured as PERLOOK, which is the percentage of time spent not looking ahead during a certain time interval.

The contributions of the proposed composite method are as follows. First, the computational cost of system can be decreased. Second, eye-detection errors and the consequent false alarms for drowsiness are decreased in the case of large head rotation. Third, we developed a hardware unit capable of eliminating specular reflection due to sunlight reflected by a driver's glasses. This unit is essential to the system because the specular reflection makes it difficult to detect the eyes. The unit comprises near-infrared (NIR) illuminators, a camera, and a narrow bandpass filter whose pass band matches the wavelength of the illuminator. The detailed configuration of the unit is described in Sec. 2.2. Fourth, we propose an eye-detection algorithm that combines adaptive boosting (adaboost), adaptive template matching, and blob detection with eye validation. This facilitates accurate eye detection, even when the driver's head is slightly rotated or the eyes are closed. Fifth, we introduce a novel eye-state–detection algorithm that combines appearance features obtained using PCA and LDA, with statistical features such as the sparseness and kurtosis of the histogram from the horizontal edge image of the eye.

The remainder of the paper is organized as follows. In Sec. 2, we describe the proposed DIMS, which comprises eye detection, eye-state detection, and inattention determination. In Sec. 3, we present experimental results with a database collected from a vehicle under various conditions. Finally, the conclusion is provided in Sec. 4.

2 Proposed Driver-Monitoring Method

2.1 *Overview of Proposed Method*

The proposed method consists of face-detection, head-orientation–estimation, eye-detection, eye-state–detection, drowsiness-detection, and distraction-detection steps, as shown in Fig. 1. In this work, we used the methods for face detection and head-pose estimation proposed in Ref. 33. In the face-detection step, the face region is found within the driver's entire facial image to remove unnecessary background and to set the regions of interest (ROIs) used in

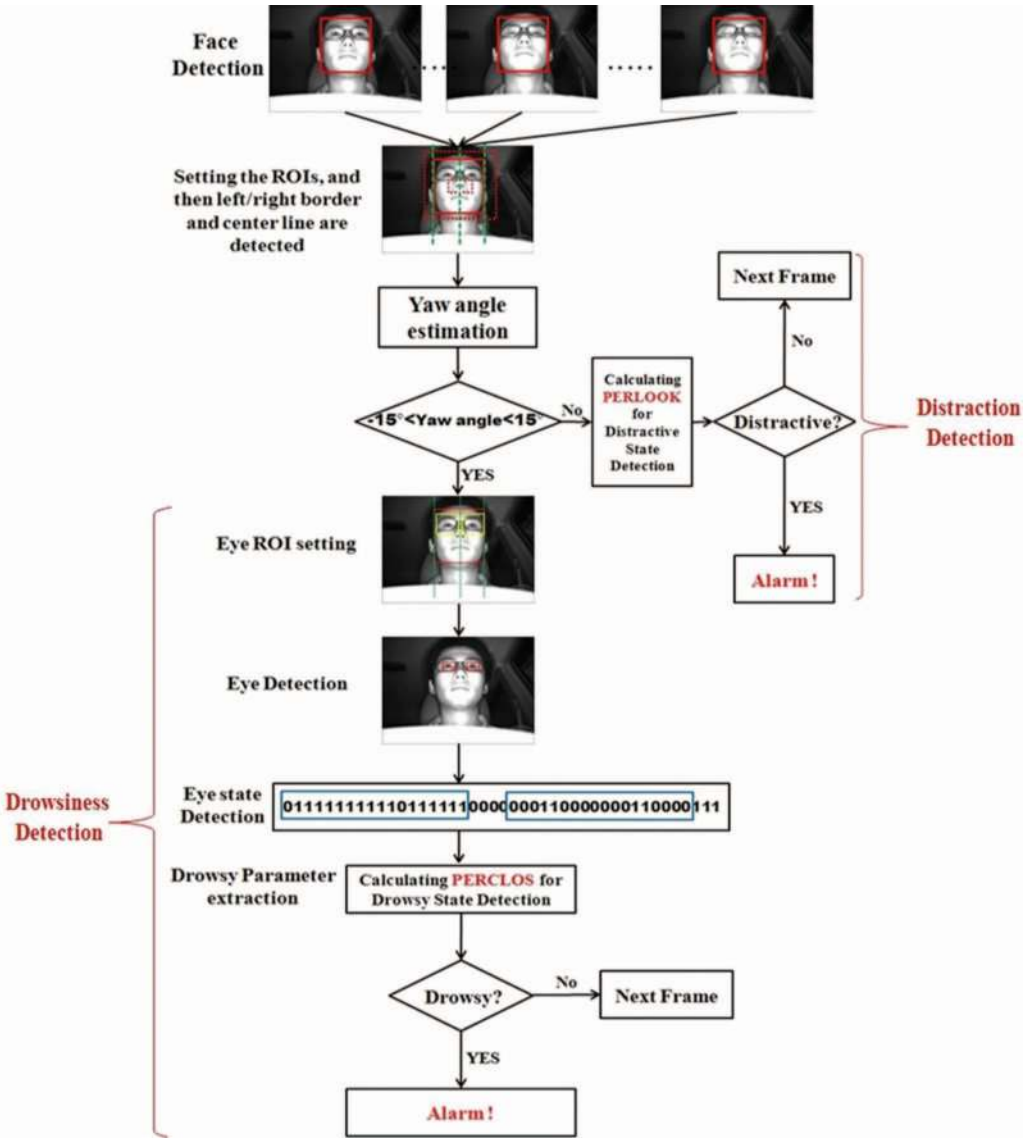


Fig. 1 Flowchart of the proposed driver-monitoring system.

the yaw angle–estimation step. In the yaw angle–estimation step, the left and right borders and center of the driver's face are extracted to estimate the driver's yaw. In addition, the normalized mean and standard deviation of the horizontal edge-projection histogram are extracted to estimate the driver's pitch. In the estimation step of the driver's head orientation, using the extracted features, the driver's yaw and pitch angles are determined by the ellipsoidal face model and SVR, respectively.[33]

The distraction and drowsiness are determined from the head pose of a driver. First, the proposed system monitors whether the driver is paying adequate attention to the road ahead based on the estimated head pose. If the frequency of time in which the driver does not see the road ahead over a certain period of time is greater than a predetermined threshold, then a warning signal is produced by the distraction detection. Even if the driver is looking straight ahead, if the percentage of time that the driver closes his/her eyes during a certain period of time exceeds a predetermined threshold, a warning signal is also generated by the drowsiness-detection system. A detailed description of each stage can be found in Sec. 2.3.

2.2 *Image Acquisition Method*

The proposed system consists of two 850-nm illuminators (LED 850-66-60),[34] a camera (EC650) (Ref. 35) having a lens (ML 0614) of 6-mm focal length,[36] a laboratory-made device for controlling the illuminators, and a narrow band-pass filter (NT43-148) (Ref. 37) placed in front of the camera. Except for the device that controls the illuminators, all the other components are commercially available. In our experiments, we also used two types of vehicles, a sedan (Grandeur, Hyundai Motors) (Ref. 38) and a sport-utility vehicle [(SUV); Sportage, KIA motors].[39]

To test the applicability of the system, images were acquired for the two types of vehicles; these images were nearly similar. When the driver wears eyeglasses, glint may be generated by NIR light, which may hide the entire eye region. To resolve this issue, in the proposed system the camera tilting angle is constrained to ~45 deg below the driver's face. In

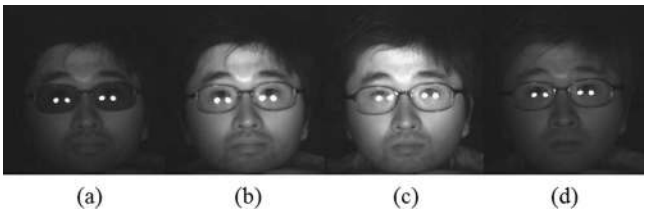


Fig. 2 Images obtained in various wavelengths using sunglasses: (a) 700 nm, (b) 750 nm, (c) 850 nm, and (d) 950 nm.

addition, the installation height of the proposed device is set to ∼70 cm from the driver's feet.

A system that can operate in various vehicle environments should satisfy the following three conditions:

1. The system should work during both daytime and nighttime. Because there is no sunlight at night, we provide additional NIR illumination to capture the driver's facial image. Because visible lights can dazzle drivers when driving, NIR illuminators were used to capture images. We also adjusted the shutter speed of the camera automatically to prevent image saturation.
2. The system should be operative for drivers who wear sunglasses. We performed experiments to identify the proper wavelength of NIR illumination. To observe the transmittance of various colors of sunglasses according to various wavelengths of NIR illuminators, we conducted tests with black, brown, blue, yellow, and pink semitransparent sunglasses. As an experiment, we used an electronically tunable liquid-crystal filter that has 10-nm bandwidth and a pass-band range from 650 to 1100 nm to capture the images of users wearing sunglasses in the range of 700–950 nm. Figure 2 depicts the captured face images of a user wearing black sunglasses at 700-, 850-, and 950-nm wavelengths. From this, it is evident that the black sunglasses can be transmitted by the NIR illumination, which has a longer wavelength than 750 nm. In the experiments with various wavelengths, we found that the transmission increased from 700 to ∼850 nm and then decreased after ∼900 nm. We can confirm in Fig. 2 that the light in 700 nm is not transmitted but that the light in 850 nm is well transmitted into the sunglasses. In Fig. 2(d), we can confirm that the brightness in 950 nm is decreased because the sensor response of the camera is decreased in accordance with the increase in wavelength. Thus, we selected 850 nm as the wavelength of the NIR illuminator, considering the good transmission, camera-sensor response, and lack of dazzling effect.
3. The system should be operative when drivers wear eyeglasses on which reflected sunlight is generated. This problem is often encountered in a real automotive environment, as shown in Fig. 3. A method was report that utilized a polarizing filter to remove the reflected sunlight.[40] However, because sunlight includes various directional lights, it is unknown which direction of these lights caused the reflection. Thus, a polarizing filter, which eliminates the reflection by only penetrating specific directional light, has limits. In the experiment with a polarizing filter, the reflection could be removed only when the directions of the polarizing filter and the light causing the reflection matched. Because the directions of light that causes the reflections can be changed, but the direction of the polarizing filter cannot be changed in real time when driving, the polarizing filter is unsuitable for vehicle application. Moreover, because only specific light passes through the polarizing filter, the image brightness was decreased.

Fig. 3 Images of reflected sunlight on eyeglasses.

To overcome these problems and remove the reflected sunlight, we use a NIR illuminator and a narrow bandpass filter whose pass band matches the illuminator. First, the narrow bandpass filter is installed in front of the camera and restricts the incoming wavelength of light to 850 nm, with a central wavelength tolerance of 2 nm.[37] In the various wavelengths of sunlight, only sunlight with a wavelength of 850 nm passes through the narrow bandpass filter and lights of most wavelengths that caused the reflection are diminished. Two high-power light-emitting diode (LED) illuminators[34] cast a light with 850 nm on the driver's face. Because the driver sits inside a car in which the amount of sunlight is not greater than that on the outside, we can make the effect of the high-power LED illuminators greater than the sunlight. Consequently, the reflections on glasses can be eliminated. A configuration of our system is shown in Fig. 4, and the experimental results for removing reflected sunlight using a NIR illuminator and a narrow bandpass filter are shown in Fig. 5.

The VTI report[32] for driver monitoring briefly mentions the use of NIR illuminators and a bandpass filter; commercial products are used, and details, such as the wavelengths of NIR illuminators and the filter, are not provided. Although NIR illuminators and a bandpass filter have been used previously for eye detection, in our study, we experimentally determined the optimal wavelengths of these components. To the best of our knowledge, the details of the NIR illuminators and bandpass filter have been reported for the first time in our study.

Fig. 4 Developed driver-monitoring device (Ref. 33).

2.3 *Eye-Detection Method Considering Driver's Head Pose*

2.3.1 *Determination of the range of eye detection considering head pose*

The face-detection and head-pose estimation process is performed before detecting the driver's eyes.[33] Through the estimated head pose, if it is determined that the driver is looking ahead, the system detects the eyes to determine drowsiness. Otherwise, the system does not detect the eyes but does monitor the driver's distraction level, which is the percentage of areas viewed other than the front for a certain time interval. The reasons why the proposed system does not detect the eyes when the driver is not looking ahead are as follows. For driver monitoring, the information regarding eye blinking is required to detect driver drowsiness. However, an early symptom of drowsy driving is that drivers begin to fall asleep when looking ahead, but not to the side. In light of this phenomenon, the proposed system does not detect the eyes when the driver is not looking ahead. This can also solve the problem of high computational cost and low accuracy of driver-monitoring systems for detecting rotated eyes. Moreover, if drivers look at a nonfrontal area when driving, this is as hazardous as drowsy driving. Thus, it should be detected by the distraction-detection method and an alarm signal should be set off. Therefore, we propose a driver-monitoring system that detects the eyes only when the driver looks ahead and monitors driver distraction when the driver does not look ahead.

Before the system detects the eyes, it is necessary to determine whether the driver is facing the front of the vehicle. In normal straight-road driving, when drivers are paying attention to the road ahead, their facial direction is within approximately ± 15 deg from the straight normal. From this fact, the range of facial directions spanning ± 15 deg is designated "front," and facial directions greater than ± 15 deg are designated "nonfront." The validity of this figure has been proven in our real-driving test and also in the experiments conducted by Toyota.[41] In the experiments by Toyota, they suggested ± 15 deg as the yaw-angle condition for the front angle of a driver. The same result is obtained in our real-driving experiments. In order to measure the ground-truth data for a head orientation, we used an electromagnetic sensor called Patriot, as shown in Fig. 6.[42] The experimental result is shown in Fig. 7.

From the above real-driving test, we can confirm that the yaw angle of a driver facing the front of a vehicle is within ± 15 deg. In the graph, the angles are greater or less than this yaw-angle range when the driver views the side mirror or side windows. Except for these cases, the head-pose range of driver is included in the above-mentioned range of front viewing, within ± 15 deg.

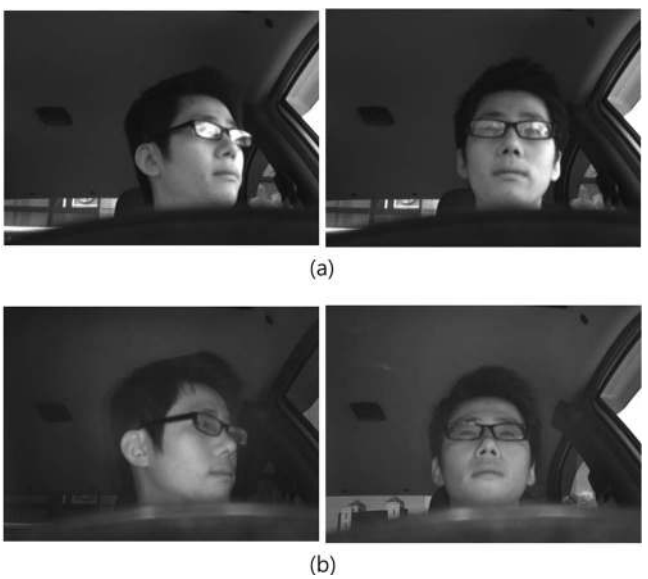


Fig. 5 Experimental results for removing a reflected sunlight with/without NIR illuminators and a narrow bandpass filter: (a) without a filter and illuminators (b) with a filter and illuminators.

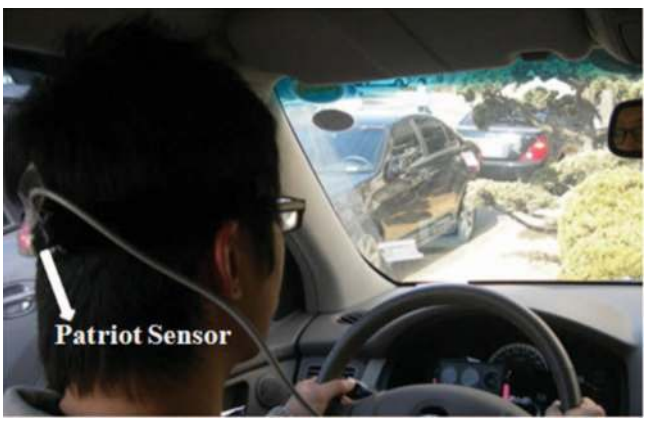


Fig. 6 Patriot sensor attached behind the driver's head (Ref. 33).

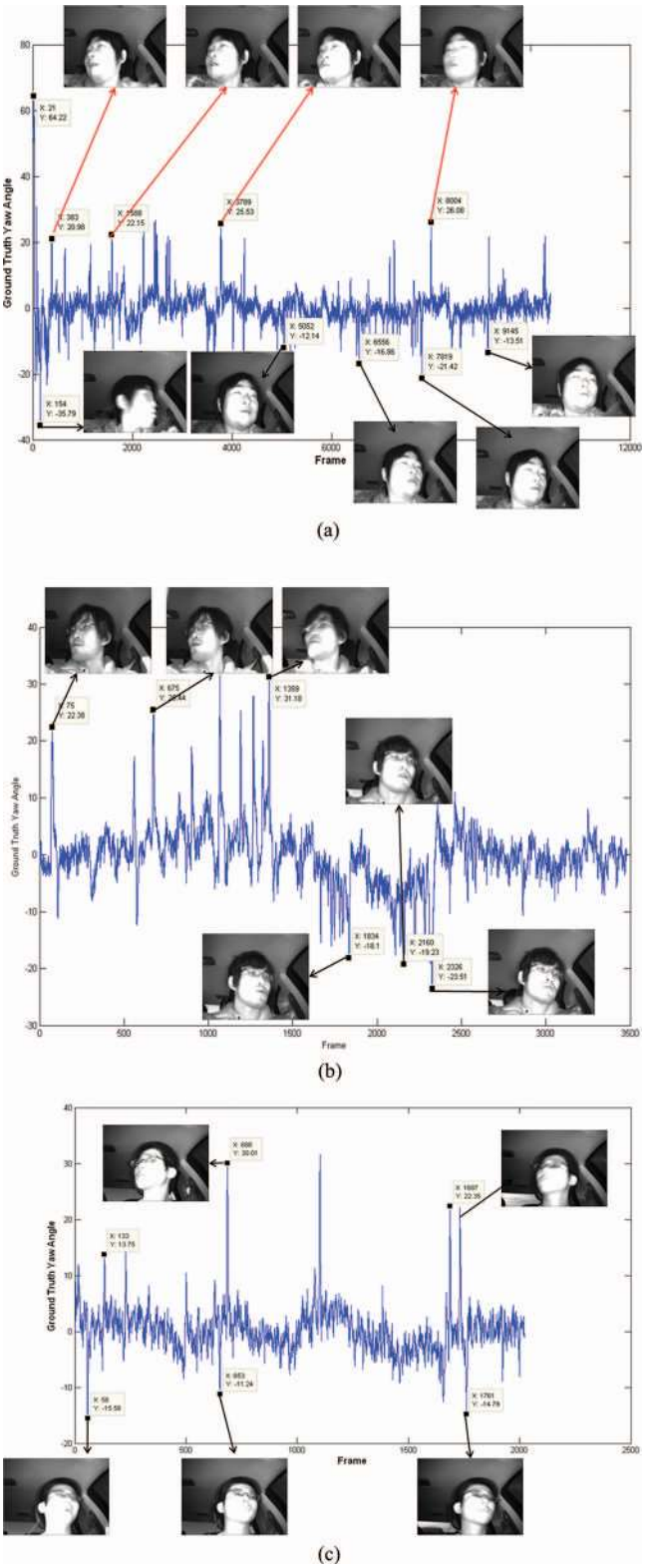


Fig. 7 Facial yaw angle distribution in a real driving test: (a) driver 1, (b) driver 2, and (c) driver 3.

The proposed system only detects eyes within this yaw angle range. If the estimated facial yaw angle is within ±15 deg, then the eye ROI is set by using the left and right border line, centerline, and statistical position of the eye in the face. The system then detects the eyes only in this region. The proposed eye-detection algorithm is presented in Sec. 2.3.2.

2.3.2 *Eye-detection method*

In a previous study, Scheirer et al.[43] proposed two different approaches—a machine learning approach and a correlation filter approach—for eye detection under unfavorable acquisition circumstances, such as low illumination, distance variation, pose variation, and blur. This eye-detection method is inconvenient because visible light is required at nighttime, which distracts the driver or causes pupil contraction. In another study, Whitelam et al.[44] analyzed eye detection across three different bands (i.e., the visible, multispectral, and short-wave–infrared (SWIR) bands), in order to illustrate the advantages and limitations of multiband eye localization. However, they did not use a narrow bandpass filter for the NIR camera; hence, the captured eye image may contain a large amount of sunlight, which can make it difficult to locate the eye region. Moreover, eye validation is not conducted, which may result in false eye detection. Our system adopts a narrow bandpass filter and eye validation in order to improve eye-detection performance.

Our eye-detection algorithm is developed by combining eye adaboost,[45] adaptive template matching,[46] blob detection,[47] and eye validation. The adaboost method has been widely adopted for detecting various facial components, such as the face, nose, mouth, and eyes, where weak classifiers are combined in a cascade; this yields a stronger classifier.[45] In our system, we used the standard OpenCV Haar cascade to detect the face and eyes.

The adaptive template-matching method is a digital image-processing technique for identifying the corresponding parts that match a template in an input image. The template and eye candidate images are compared to determine whether the two images have a matching region. One method for template is given by

$$R(x, y) = \frac{\sum_{x',y'} [T(x', y') - I(x + x', y + y')]^2}{\sqrt{\sum_{x',y'} T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}}, \tag{1}$$

where I denotes the input image, T denotes the template, and $R(x, y)$ denotes the matching result at position (x, y). The ranges of x' and y' are $[0, w - 1]$ and $[0, h - 1]$, respectively. w and h are the width and height of the template image, respectively. If the matching value $R(x, y)$ is small, then it indicates higher similarity; if it is large, then it indicates lower similarity. Among all the calculated $R(x, y)$, the position (x, y) where $R(x, y)$ is minimized is selected as the final matching position. The shape of the eye slightly changes in every frame; hence, it is impossible to detect all possible eye shapes with one fixed template. Thus, the template is updated in every frame by the eye image that is successfully detected in the previous frame.

Blob detection is explained as follows. From the original eye image, a binary image is obtained on the basis of the adaptive threshold that is set with the mean value of the block; then, it is subjected to the morphology step. In the morphology step, there are two phases, erosion and dilation.[48] First, the white pixels are expanded and the black pixels are dimin-

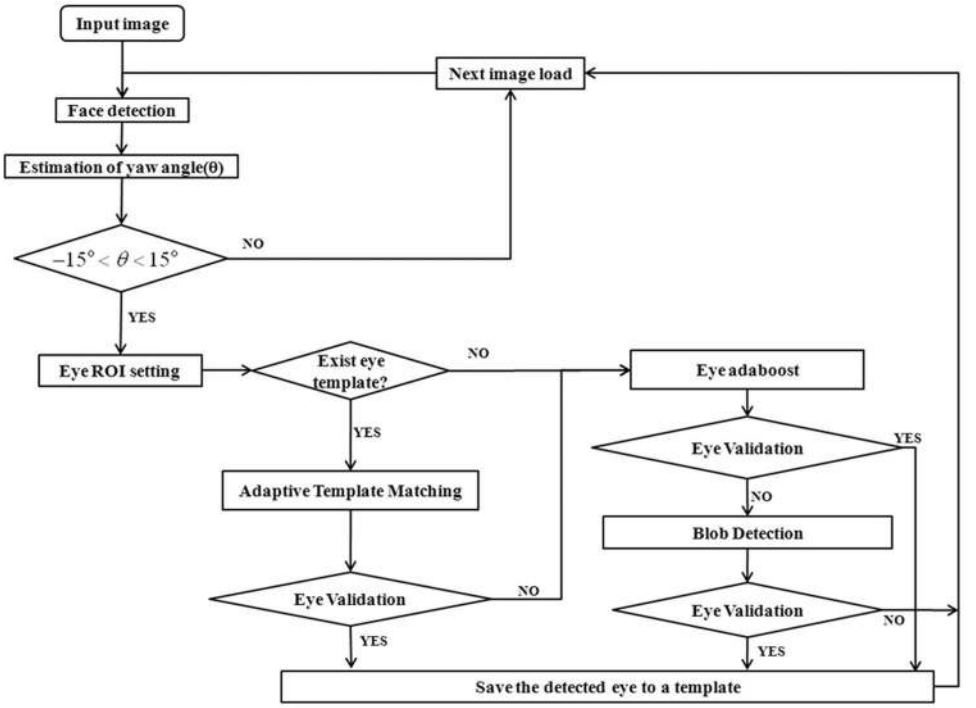


Fig. 8 Flowchart of proposed eye-detection algorithm.

ished by an erosion operator. Because the remaining blobs of the eyeglass frame may be wrongly detected as eyes, the eyeglass frame should be removed by an erosion operator. Next, the black pixels of the eye region diminished by the erosion operator are expanded by a dilation operator. This process of sequentially erosion and dilation is defined as the opening process, and it has the effect of noise elimination.[48] After the opening process is applied twice, a large blob is generated for an eye and small blobs are generated for noise. The labeling process is performed with all the remaining blobs. Thus, the remaining blobs are grouped with neighboring blobs and labeled according to their connectivity.[48] Therefore, all the blobs are represented as being isolated. After that, all the pixel "positions" of each isolated blob can be obtained. Accordingly, the "size" of the blob is calculated as the number of the pixels of the blob. In addition, the width and height of the outermost rectangular box including the blob can be obtained. Then, the ratio of the height to the width can be calculated as the "shape" of the blob. The position, size, and shape are finally used to determine the location of the eye. Open source is used for blob detection.[47]

These methods are combined to detect the eyes quickly and accurately. If adaboost alone is used for eye detection, then it has a great deal of computational cost and cannot detect closed eyes. To resolve these problems, in the first frame, eye adaboost is used in the eye-detection step and, after the eye verifier checks that the eyes are correctly detected, the detected eye image is saved as a template. In the next frame, the eyes are detected by adaptive template matching using the previously saved template and the template is then updated by the newly detected eye image. This is better than only using eye adaboost in every frame in terms of computational cost and performance of closed-eye detection. However, eye shapes are changeable from an open shape to a closed shape. If the shape of an opened eye is saved as the template and the closed eye appears at the next frame, then the eyes cannot be detected by the adaptive template-matching method. Thus, when the adaptive template-matching method fails to detect eyes, adaboost is used again to detect the eyes. If adaboost also fails to detect the eyes, then the blob-detection method is applied to do so. When the eyes cannot be detected by these three eye detectors, the previous position of the eye is kept. Figure 8 depicts a flowchart of the eye-detection algorithm (a detailed explanation for eye detection will be provided).

The system attempts to realize eye detection using the adaboost eye detector.[45] The eye adaboost has good performance for detecting eyes when the driver's head pose is frontward. However, as shown in Fig. 9, it has the disadvantage of missing eyes when the driver's head pose is rotated or when the driver's eyes are closed. It also takes a long time to detect the eyes if the eye adaboost is applied in every frame.

To resolve these problems with adaboost, the adaptive template-matching method was combined with adaboost. In the first frame, the system detects eyes by using adaboost and then saves the detected eye images to a template. In the next frame, adaptive template matching is applied to locate the eyes. As such, even the rotated or closed eyes that were not detected by adaboost can be detected. The advantage of adaptive template matching is that it requires less computational cost than adaboost. The adaboost needs 30 times more computational cost than the adaptive template method. As such, the eye-detection method combined with adaboost and adaptive template matching is more efficient than using adaboost alone. This combination is robust to head-pose variation and can detect a closed eye, as shown in Fig. 10.

However, because the eye detector using these combinations may still fail to find eyes, the blob-detection method is additionally supplemented to the combination to improve the performance of eye detection. In particular, blob detection

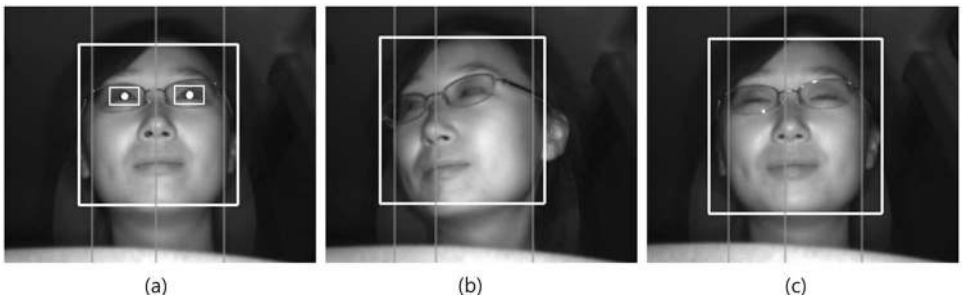


Fig. 9 Eye-detection results obtained using adaboost: (a) Frontal face (success), (b) rotated face (fail), and (c) closed eye (fail).

improves the detection rate of closed eyes that are not detected by adaboost and template matching and saves processing time. Figure 11 shows a flowchart of the blob-detection method.

From the original eye image, a binary image is obtained by the adaptive threshold that is set with the mean value of the block and then passes through the morphology step. In the morphology step, there are two phases: erosion and dilation. First, the white pixels are expanded and the black pixels are diminished by an erosion operator. Because the remaining blobs of the eyeglass frame can be misrecognized as eyes, the eyeglass frame should be removed by an erosion operator to prevent this from happening. Next, the diminished black pixels of the eye region by the erosion operator are expanded by a dilation operator. This process of sequentially passing through erosion and dilation is defined as the opening process and has an effect of noise elimination.[48] After the opening process is applied twice, a large blob for an eye and small blobs for noise are finally generated. All remaining blobs are grouped with neighbor blobs into individual groups, and a labeling process ensues. To apply the labeling process, we used an open source of blob detection (this source code is available in Ref. 47). The location of the eye is finally determined by considering the size, shape, and position of blobs.

In the eye-detection step, the eye-validation process plays an important role in checking whether the detected eye region actually contains eyes. As shown in Fig. 12, this is an eye-validation process that validates the eye detected by adaboost, adaptive template matching, and blob detection. If a noneye region is falsely found by the eye detector and saved in the template without any further validation process, the region containing no eye will be continuously detected by the adaptive template matching in the next frame. Thus, an eye-validation process is needed to prevent error propagation.

In the eye-validation process, we used two classical feature-extraction methods, namely, principal component (PC) analysis[49–52] (PCA) and linear discriminant analysis (LDA),[49–51] which were used in a different application in (Refs. 51 and 53). In this paper, we use the PCA + LDA methods to extract features for eye validation.

PCA is a well-known unsupervised algorithm for linear feature extraction; it is a linear mapping that uses the eigenvectors with the largest eigenvalues. In the PCA method, the pixels of an eye image are ranked as a column vector $x_i = (x_{1i}, x_{2i}, \ldots, x_{ni})^T \in R^n$ and then by the l matrix $X = \{x_1, x_2, \ldots, x_l\}$ denotes the training sample set that consists of l eye images. After the mean vector of $\boldsymbol{X}$ is calculated, centered data are obtained by subtracting the mean from all samples. The PCA can be used to find a linear transformation orthonormal matrix $W_{\mathrm{PCA}}[m \times n(m \ll n)]$, mapping the original n-dimensional feature space into m-dimensional feature subspaces. The reduced feature vector $\mathbf{y}_i$ is defined by

$$\mathbf{y}_i = W_{\mathrm{PCA}}^T x_i \qquad (i = 1, 2, \ldots, l). \tag{2}$$

The columns of W_{PCA} are the m eigenvectors associated with the m largest eigenvalues of the scatter matrix S, which is defined as

$$S = \sum_{i=1}^{l} (x_i - \mu)(x_i - \mu)^T, \tag{3}$$

where μ is the mean of all images in the training set.

LDA is a supervised learning method that utilizes the category information associated with each sample. The goal of LDA is to maximize the between-class scatter while minimizing the within-class scatter. The within-class scatter matrix S_{W} and between-class scatter matrix S_{B} are defined as

$$S_{\mathrm{B}} = \sum_{j=1}^{c} (\mu_j - \mu)(\mu_j - \mu)^T, \tag{4}$$

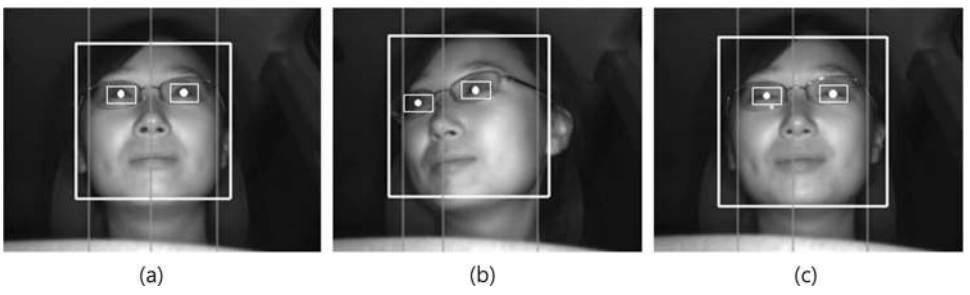


Fig. 10 Eye-detection results obtained by combining with adaptive template matching and adaboost: (a) Frontal face (success), (b) rotated face (success), and (c) closed eye (success).

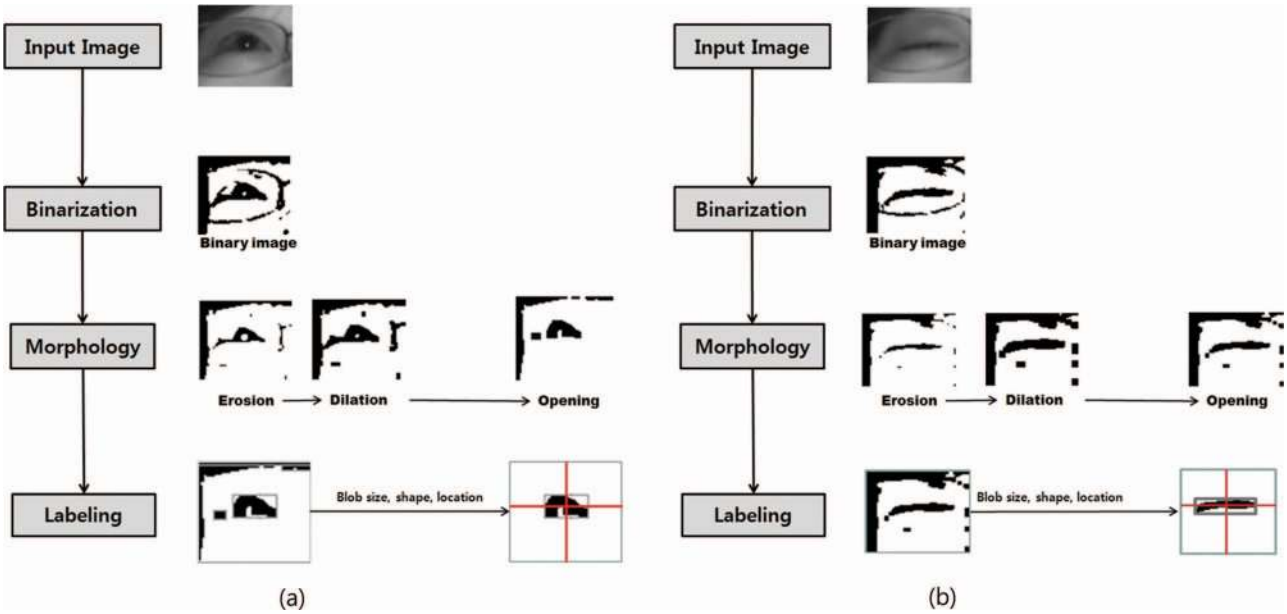


Fig. 11 Flowchart of the blob-detection method: (a) open eye and (b) closed eye.

$$S_W = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_i^j - \mu_j)(x_i^j - \mu_j)^T, \tag{5}$$

where x_i^j is the i'th sample of class j, μ_j is the mean of class j, μ is the mean image of all classes, N_j is the number of samples of class j, and c is the number of classes.

In order to guarantee that S_W does not become singular, we require at least $n + c$ samples. In practice, it is difficult to obtain so many samples when the dimension of feature n is very high. PCA + LDA is proposed to solve this problem.[51,52] Let the output of PCA be the input of LDA, and finally, the feature vectors of eye images are given by

$$z_i = W_{\mathrm{LDA}}^T W_{\mathrm{PCA}}^T x_i \qquad (i = 1, 2, \ldots, l). \tag{6}$$

Now, let us return to our eye-validation process. Eye validation is composed of a training and testing process, as shown in Fig. 12. In the training step, the eye-template image of 40×26 pixels in the training database is defined as the column vector of 1040×1 pixels. The column vector is then reduced to an eigenvector with 13 dimensions by using PCA, which seeks a projection that best represents the original data in a least-squares sense. The eigenvector is reduced to 11 dimensional spaces (features) by using LDA, which seeks a projection that best separates the data in a least-squares sense. After the features that best separate eye and noneye are normalized, they are entered into a support vector machine (SVM) and trained to classify eye and noneye well. An SVM is a classifier that finds the maximum

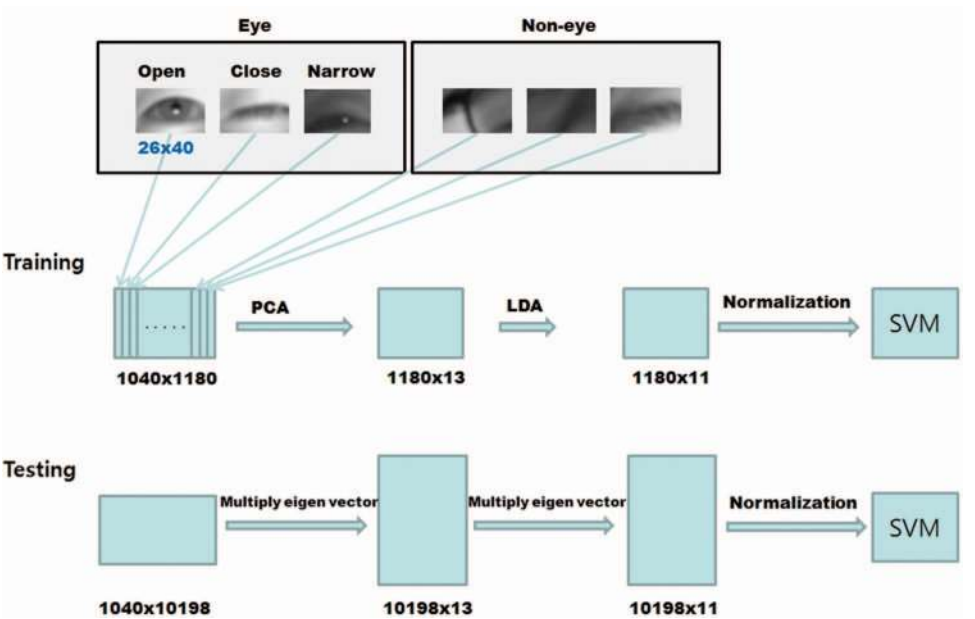


Fig. 12 Eye-validation process.

margin hyperplane that optimally separates the data into two categories for the given training set. In general, an SVM can be represented by[54]

$$f(x) = \text{sgn}\left(\sum_{i=1}^{k} \alpha_i y_i K(x, x_i) + b\right), \tag{7}$$

where k is the number of data points and $y_i \in \{-1, 1\}$ is the class label of training point x_i. The coefficients α_i are found by solving a quadratic programming problem with linear constraints, and b is a bias. To obtain a good classification performance, a SVM needs to choose a "good" kernel function, $K(x, x_i)$. In this paper, we used a radial basis function (RBF) kernel, as shown in Eq. (8), because it generally has fewer numerical difficulties[55]

$$K_{\text{RBF}}(x, x') = \exp(-\gamma ||x - x'||^2). \tag{8}$$

In the testing step, after features are extracted from the eye image in the testing set by PCA + LDA in the same way as the training step, the extracted features are entered into the SVM to learn the classification of eye and noneye. Figure 12 depicts this eye-validation process for training and testing. When the eye validation was added to the eye-detection algorithm, the accuracy of eye detection was improved. This result will be shown in the Sec. 3 experiment.

2.4 *Eye-State Detection (Determination of Open or Closed Eye)*

Drivers that are sleepy exhibit several visual behaviors that are easily observable from the changes in their facial features, such as the eyes, head, and face. There are many parameters of drowsiness that reflect the vigilance level of a driver, such as eye-closure duration (ECD), frequency of eye closure (FEC), average eye-closure speed (AECS), and percentage of eye closure over time (PERCLOS).[56,57] For all of them, the determination of eye state, open or closed eye, is a necessary step. In this section, we introduce the novel features for eye-state analysis. The performance of determining eye status (opening and closing) is enhanced by combining the appearance features by PCA and LDA, and the statistical features, such as the sparseness and kurtosis of the histogram, from the eye-edge image.

2.4.1 *Feature extraction*

Principal component and linear discriminant analyses. The features used to determine open or closed eye are 12 dimensional features obtained by PCA + LDA. This method is the same as the eye-validation process explained in Fig. 12. PCA + LDA exhibits better performance than PCA alone when there are few samples in each class.[51]

Sparseness and kurtosis. Other features for determining open or closed eyes are the sparseness and kurtosis of the histogram from the eye-edge image. These features are extracted to classify open and closed eyes by using horizontal edge image processing. Here, we briefly explain how these features are extracted. The Sobel operator performs a 2-D spatial gradient measurement on the detected eye image. Typically, this is calculated by the absolute gradient magnitude at each point in an input gray-scale image. The Sobel edge detector uses a pair of 3×3 convolution masks, one estimating the gradient in the horizontal direction (columns) and the other estimating the gradient in the vertical direction (rows). A horizontal edge image is obtained using a horizontal Sobel convolution mask. By applying a p-tile threshold method to the horizontal edge image, it is transformed into a binary image.[58] From the training data, we experimentally obtained 7% by the p-tile method. The p-tile threshold method uses the size and area of eye to obtain a binary image. For example, if eye occupy p% of total area, gray-value histogram of input image is divided by the percentage, and then a threshold is selected considering the p% pixel for eye. The binarized image is projected onto the horizontal axis (x-axis), and a vertical projection histogram is then acquired, as shown in Fig. 13. The histogram of the open eye exhibits a peak that is concentrated into vertical lines. This peak is caused by the brightness difference of the pupil, specular reflection, and iris. In contrast, the histogram of the closed eye exhibits a flat distribution on the x-axis. Sparseness is a parameter that measures how much energy of a vector is concentrated on a few components.[59] This implies that most values are close to zero and that only a few take significantly nonzero values. Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution.[60] That is, data sets with high kurtosis tend to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. A uniform distribution would be an extreme case. Therefore, there is similarity between the two parameters. The equations of sparseness and kurtosis are as follows.

$$\text{Sparseness}(x) = \frac{\sqrt{d} - \left(\sum |x_j|\right) \Big/ \sqrt{\sum |x_j|^2}}{\sqrt{d} - 1}, \tag{9}$$

where d represents the dimensionality of the vector $\boldsymbol{x}$ whose j'th component is x_j.

$$\text{Kurtosis} = \frac{(1/N)\sum_{n=0}^{N-1} (x_n - \mu)^4}{\sigma^4}, \tag{10}$$

where μ is the mean of x and σ is the standard deviation of x.

Figure 13 shows that open/closed eye can be classified by the sparseness and kurtosis of the histogram. As shown in Fig. 13, it can easily be predicted that an open eye has larger values of sparseness and kurtosis than a closed eye because the open eye has greater bulginess than a closed eye. We can confirm that the distribution of the open-eye histogram is concentrated in the center of the x-axis when the histogram of the closed eye exhibits a relatively flat distribution. Therefore, open and closed eyes are classified in terms of whether the sparseness and kurtosis of the projected histogram exceed a certain threshold value.

2.4.2 *Combination of features*

The features that are used to classify open or closed eyes were first extracted from the appearance features of 12 dimensions by PCA + LDA and were then extracted from the statistical features of two dimensions by sparseness and kurtosis. These extracted features were applied to the SVM classifier. We used a SVM with RBF kernels because the classification ability of the SVM with radial basis function kernels was superior to that of the SVM with polynomial and linear kernels. Figure 14 shows these processes that combine features and classify open or closed eyes.

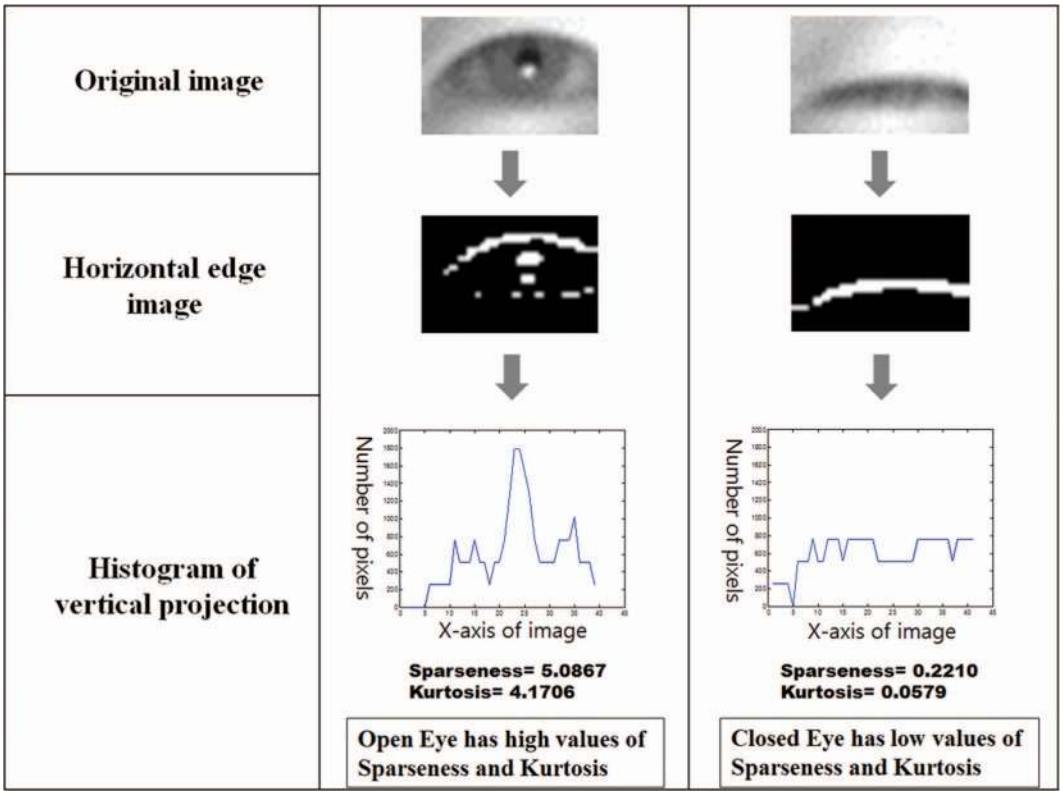


Fig. 13 Horizontal edge-image process for classification of open and closed eye.

2.5 *Detection of Drowsiness and Distraction*

After classifying open or closed eyes and estimating the head pose of a driver, the inattention of the driver is determined by PECLOS, which represents the drowsiness level, and PERLOOK, which represents the distraction level. PERCLOS is a parameter of drowsiness that is widely used to monitor the drowsiness of a driver. It is defined as the portion of time that the driver's eyes are closed over a certain period.[57,61–63] This parameter can be obtained from

$$\mathrm{PERCLOS}[k] = \frac{\sum_{i=k-n+1}^{k} \mathrm{Blink}[i]}{n} \times 100\,(\%), \qquad (11)$$

where PERCLOS[k] means PERCLOS figure in the k'th frame and n is a period measuring PERCLOS. Blink[i] represents that the eye is opened or closed in the i'th frame. It has "0" if the eye is opened and "1" if the eye is closed. The higher the PERCLOS is, the higher the level of driver fatigue. When PERCLOS exceeds the predetermined threshold, the proposed system generates a warning of inattention. The predetermined threshold was experimentally determined as 0.2 by our experiments with training data.

When a driver cannot look at the road ahead because the head is rotated, it causes the same accident hazard as when a driver cannot look ahead because their eyes are closed. As such, we propose a parameter, PERLOOK, which measures the driver-distraction level in the same way as PERCLOS. PERLOOK is defined as the proportion of time that a driver's head is rotated and the driver does not look at the road ahead. This parameter can be obtained from

$$\mathrm{PERLOOK}[k] = \frac{\sum_{i=k-n+1}^{k} \mathrm{Nonfront}[i]}{n} \times 100\,(\%), \qquad (12)$$

where PERLOOK[k] indicates the PERLOOK figure in the k'th frame and n is a period measuring PERLOOK. Nonfront[i] indicates that the driver is looking at the frontal or nonfrontal area in the i'th frame. It is 1 if the driver is look-

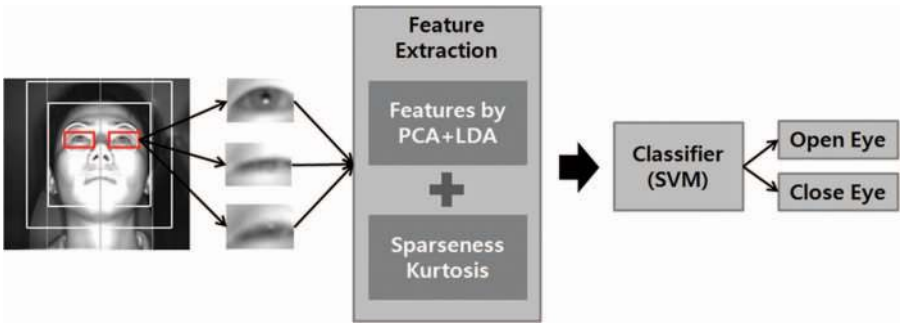


Fig. 14 Process of combining features and classifying open or closed eyes using an SVM.

Table 3 Specification of Database 1 for training.

Daytime	No. subjects	12 persons (Male: 9, Female: 3, without glasses: 2, with glasses: 10)
	No. images	44,197 images from 216 image sequences (12 persons×18 gaze zones)
Nighttime	No. subjects	11 persons (Male: 8, Female: 3, without glasses: 3, with glass: 8)
	No. images	41,155 images from 198 image sequences (11 persons×18 gaze zones)

ing at the non-frontal area and 0 if the driver is looking at the frontal area. The higher the PERLOOK is, the higher the driver-distraction level is. When PERLOOK is greater than a predetermined threshold, the proposed system generates a warning of inattention. The predetermined threshold was experimentally determined as 0.35 by our experiments with training data. From Sec. 2.3.1, if the yaw angle of a driver is greater than +15 deg or less than –15 deg (based on the frontal direction), and if the corresponding accumulative time is longer than 3.5 s during the unit interval of 10 s, our system determines it as a "distraction." For example, if the yaw angle and accumulative time when a driver looks at a GPS device or changes a CD is satisfied by this condition, it is determined as a distraction.

3 Experiments

3.1 *Database*

In order to evaluate the proposed method, we attached the developed system in front of a dashboard, as shown in Fig. 4, and we collected 162,772 frames from 22 subjects. Several experiments were carried out using images of subjects of both sexes, some of whom wore sunglasses and eyeglasses, during both daytime and nighttime. The databases were acquired in a car using an NIR camera with a resolution of 640 × 480 pixels and a frame rate of 25 fps.

In our databases, there are two data sets for training and testing. 12 subjects were categorized into a training group, and the remaining 10 subjects were categorized into a testing group. The images from the training group were stored in database 1, whereas those from the testing group were recorded in database 2. Database 1, which is shown in Table 3, has also been used in a previous study.[33] It contains 85,352 images (6700–7500 images per subject). 1090 images from database 1 were used to train the eye validation process, as shown in Fig. 12 and Table 4; 703 images were used for the classification process for open and closed eyes, as shown in Fig. 14 and Table 5. The 85,352 training images include numerous overlapped data, which can cause overfitting of the classification hyperplane of the SVM. In order to solve this problem, 1090 samples from 12 subjects were randomly selected and used in SVM training for eye validation. The training set for the SVM classifier in the eye-validation process contains 860 positive samples (eye images) and 230 negative samples (noneye images), as shown in Figs. 15(a) and 15(b). In addition, 703 samples from 12 subjects were randomly selected and used in SVM training for eye-state detection. The training set for the SVM classifier in the eye-state–detection process contains 530 positive samples (open-eye images) and 173 negative samples (closed-eye images), as shown in Figs. 15(c) and 15(d).

The remaining images in database 1 were heuristically used for parameter optimization in eye detection (Sec. 2.3.2). In conclusion, the training data were only used for parameter optimization and SVM training.

Database 2 contains 77,420 images (7500–8000 images per subject) from the testing group. In database 2, longer image sequences were captured for each subject in order to test the proposed inattention-detection method. The subjects were requested to look frontally for 5 s and to then drive normally, drowsily, or distractedly for the remaining period. Thus, for each subject, we obtained three types of sequences: a normal driving video, a drowsy driving video, and a distracted driving video. In addition, the subjects who did not wear glasses were asked to wear sunglasses in order to capture a new sequence. Similarly, the subjects who wore glasses were asked to remove their glasses or wear sunglasses in order to capture a new sequence.

As compared to the number of drowsy driving videos, the number of distracted driving videos is relatively large. The reason is as follows. A driver can gaze at the 12 different positions (1–5, 8, 11, 14–18) described in Fig. 16; hence, we need to test the performance of distraction detection for the various gaze directions. Thus, for each subject, we obtained multiple distracted videos in which the subject gazed at the 12 different positions.

Table 4 Train database for eye validation.

		Number of eye images used as training data in eye validation	
	Glass type	Left eyes	Right eyes
Eye (positive samples)	Not wearing glasses	627	473
	Eyeglasses	128	208
	Sunglasses	105	189
	Subtotal	860	870
Noneye (negative samples)	Not wearing glasses	78	123
	Eyeglasses	103	108
	Sunglasses	49	79
	Subtotal	230	310
Total		1090	1180

Table 5 Train database for eye state detection.

		Number of eye images used as training data for blink detection	
	Glass type	Left eyes	Right eyes
Open eye (positive samples)	Not wearing glasses	420	440
	Eyeglasses	50	70
	Sunglasses	60	50
	Subtotal	530	560
Closed eye (negative samples)	Not wearing glasses	102	105
	Eyeglasses	33	40
	Sunglasses	38	40
	Subtotal	173	185
Total		703	745

In the case of distraction, a driver is not looking ahead. Thus, the videos in which a driver gazed at 12 different positions among the 18 positions shown in Fig. 16 were used for the performance evaluation of distraction detection. Unlike distraction detection, only two states [i.e., drowsy and normal (awake)] can be considered in the case of drowsiness detection. Hence, the number of drowsy videos obtained from each

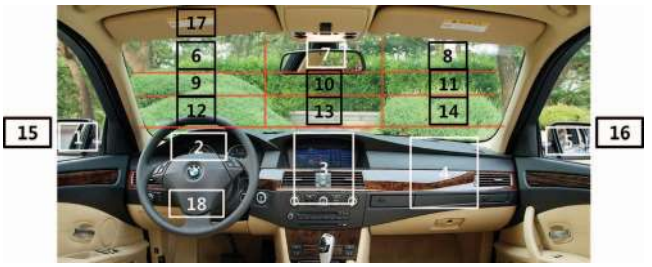


Fig. 16 Discrete gaze zones in a vehicle.

subject is less than that of distracted videos. The duration of each distracted video is ∼1 min, whereas that of each drowsy video is around 5–10 min. Table 6 shows database 2 in detail.

3.2 *Experimental Results*

The experimental phase consists of three parts. First, the eye-detection algorithm was tested using image sequences of different people obtained in a vehicle under different times and light conditions. Then, the algorithm for eye-state detection was tested. The features used to classify eye states were extracted from a long image sequence. Finally, the algorithm measuring the inattention level of a driver was tested. The inattention level is composed of the drowsiness and distraction levels. Drowsiness level was measured by PERCLOS and distraction level was calculated by PERLOOK.

3.2.1 *Eye-detection results*

In order to evaluate the performance of our eye-detection algorithm, three experiments were carried out and compared to each other. The first was performed using only the eye adaboost with our database 2. When it is applied to every

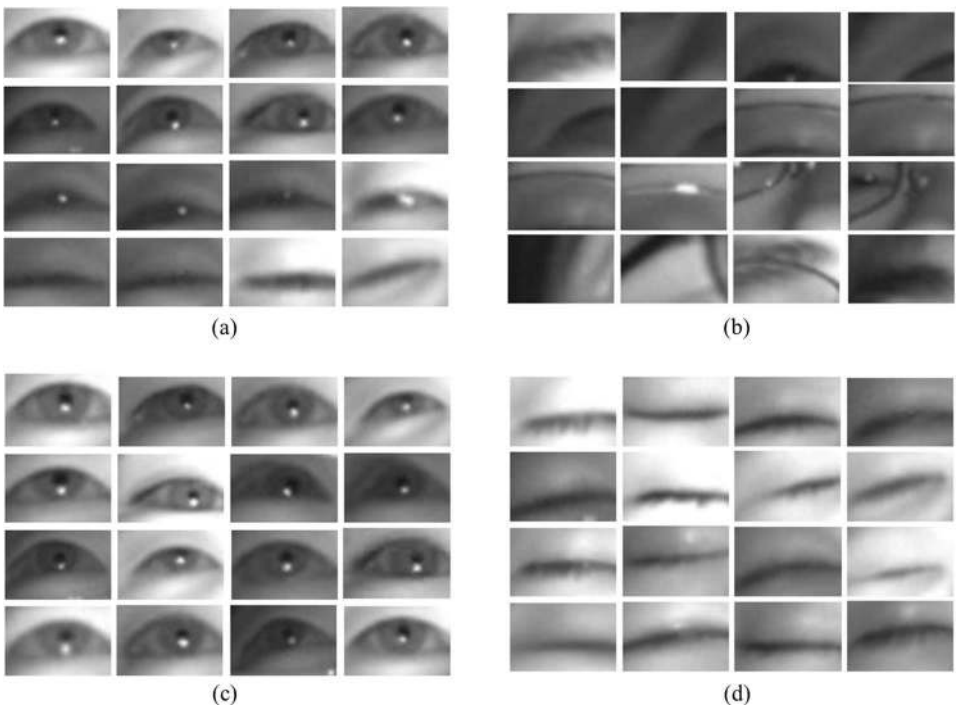


Fig. 15 Positive and negative samples provided to the SVM classifier in the training phase: (a) Eye images, (b) noneye images, (c) open-eye images, and (d) closed-eye images.

Table 6 Specification of Database 2 for testing.

	No. subjects	9 persons (Male: 6, Female: 3, without glasses: 9, with glasses: 4, with sunglasses: 5)
	No. images for eye detection and eye state detection	40,256 images from 36 image sequences
	No. images used for eye detection evaluation	32,365 images from 36 image sequences (7891 images were excluded because the subject was looking away from the road ahead.)
Daytime	No. images used for eye state detection evaluation	30,483 images from 36 image sequences (9773 images were excluded because the subject was looking away from the road ahead (7891) or because the eyes could not be classified as open or closed (e.g., narrow eyes) (1882).
	No. sequences used to measure inattention level	189 video sequences driving normally
		36 video sequences driving drowsily
		189 video sequences driving distractedly (Total number of images: about 239,200 images from 414 image sequences.)
	No. subjects	10 persons (Male: 7, Female: 3, without glasses: 5, with glasses: 5)
	No. images for eye detection and eye state detection	37,164 images from 20 image sequences
	No. images used for eye detection evaluation	28,608 images from 20 image sequences (8556 images were excluded because the subject was looking away from the road ahead.)
Nighttime	No. images used for eye state detection evaluation	26,672 images from 20 image sequences (10,492 images were excluded because the subject was looking away from the road ahead (8556) and because the eyes could not be classified as being open or closed (e.g., narrow eyes) (1936)).
	No. sequences used to measure inattention level	120 video sequences driving normally
		20 video sequences driving drowsily
		120 video sequences driving distractedly (Total number of images: about 105,800 images from 260 image sequences.)

sequence frame, it requires a long processing time and fails to detect eyes when the head is rotated or the eyes are closed. Thus, it resulted in unsatisfactory eye-detection performance and a slow processing time, as shown in Table 7. In the second experiment, we combined eye-detecting methods, including adaboost, adaptive template matching, and blob detection, and this improved both the detection accuracy of the closed-eye and the computational performance. The third experiment was performed by adding the eye validation to those used in the second experiment. When the eye validation was added to the eye-detection algorithm, the accuracy of eye detection was improved. Without any eye verifier to check whether the detected eye region truly contains an eye, the falsely detected noneye image may be stored in the template and the falsely detected eye could be repeatedly detected by the adaptive template matching from the next frame. Consequently, in the second experiment, propagation error was generated and the error rate was 7.56% in eye detection. However, after applying the eye-validation process, the error rate was decreased from 7.56 to 1.04%.

The eye-detection rate is calculated as follows. First, images looking at the frontal area are extracted from total images in database 2 (Table 6) and then manually divided into eye images [as in Fig. 15(a)] and noneye images [as in Fig. 15(b)]. Then, the eye-detection rate is calculated by the number of images that successfully detect eyes over the total number of images. Table 7 presents the eye-detection rate results and also shows that the eyes can be ∼98.9% success-

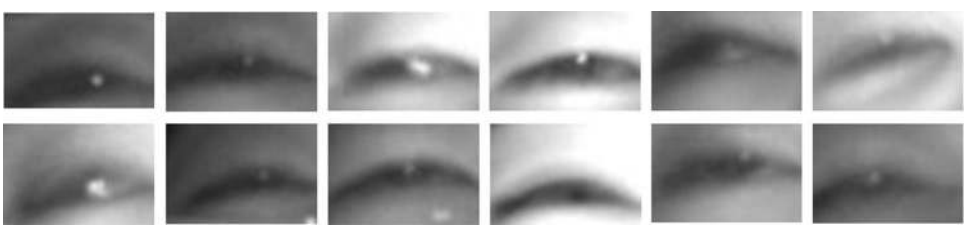


Fig. 17 Excluded narrow eye samples.

Table 7 Results obtained after eye detection algorithms on images in Database 2.

Conditions				Eye-detection rate		
Time	Glass type	Sequences	Frames	Adaboost	Proposed method (without the eye validation process)	Proposed method (with the eye validation process)
	Not wearing glasses	18	19,994	76.52%	92.54%	98.55%
Day	Eyeglasses	8	8,012	67.35%	92.35%	98.88%
	Sunglasses	10	4,359	63.53%	92.02%	98.91%
	Subtotal	36	32,365	72.50%	92.42%	98.68%
	Not wearing glasses	10	16,860	73.98%	92.58%	98.85%
Night	Eyeglasses	10	11,748	70.12%	92.05%	97.89%
	subtotal	20	28,608	72.39%	92.36%	98.46%
Total		68	60,973	72.45%	92.39%	98.58%
Processing time				23.1 ms	11.5 ms	14.5 ms

fully detected by the proposed eye-detection method regardless of time, gender, and glass type. Detection error is caused by misrecognition of eyeglass frames, eyebrows, and eyes that are partially visible in Fig. 15(b) as true eyes. However, this detection error is not continuous in the video sequences because the eye template is reset by the eye-validation process in the next frame when the eye detector detected a noneye.

3.2.2 *Eye-states–detection results*

In order to evaluate the performance of determining eye status, eye images from database 2 in Table 6 were manually divided into two classes, open and closed eyes, and they were used as ground-truth data. Two experiments were then performed. The first experiment is designed to test the effect of PCA + LDA features compared to PCA features. The second experiment is designed to compare and analyze the classification performance of multiple features combining PCA + LDA, sparseness, and kurtosis.

It is of note that narrow eyes, such as those in Fig. 17, were excluded from the performance measurement in the testing data. Because human eyes change smoothly from an open eye to a narrow eye and then a closed eye, it is difficult to classify crisply whether a narrow eye is an open eye or a closed eye in the middle of the change. However, although a narrow eye is judged as either an open or a closed eye, it does not affect the total performance of determining drowsy behavior because it is a temporary and transitional state between an open and closed eye. Therefore, narrow eyes were excluded from the testing data.

The performance in eye-states detection is measured by considering two errors. For convenience, when an open eye is falsely accepted as a closed eye, we define it as a type I error. In addition, when a closed eye is falsely rejected as an open eye, we define it as a type II error. Thus, the performance of eye-states detection was measured by type I error and type II error.

In the first experiment, in order to detect eye states, features are extracted by PCA or PCA + LDA and the features are then entered into SVM and finally classified into open and closed eyes. This process is the same as the eye-validation process explained in Fig. 12. As shown in Table 8, 530 frames of open eyes and 173 frames of closed eyes were used in the training and some of the eye images in database 2 were tested. Then, 12 dimensional features for the best performance were selected. The method using the PCA + LDA exhibits better performance in classifying eye states than the method using only PCA in our experiment. In particular, the PCA + LDA feature is more effective than the PCA or LDA feature when there are fewer training samples in each class. The combined features of PCA + LDA are less sensitive to different illuminations, whereas the PCA feature is sensitive to changing illumination.[51,53]

Table 9 shows the eye-states–detection results by PCA + LDA for each dimension. First, in order to find the proper feature dimension yielding the best recognition rate when features were extracted solely by PCA, we tested for each dimension and obtained the 15 dimensions as the best number

Table 8 Training and testing data sets for eye states detection.

	Eye states included	
Data set	Open eye	Closed eye
Train	530	173
Test	7250	800

Table 9 Recognition error rates by eliminating the first one to six principal components.

	No. eliminated principal components											
	1 (14 dim.)		2 (13 dim.)		3 (12 dim.)		4 (11 dim.)		5 (10 dim.)		6 (9 dim.)	
Input feature	Type I error	Type II error	Type I error	Type II error	Type I error	Type II error	Type I error	Type II error	Type I error	Type II error	Type I error	Type II error
PCA + LDA	3.52% $\left(\frac{255}{7250}\right)$	3.50% $\left(\frac{28}{800}\right)$	3.43% $\left(\frac{249}{7250}\right)$	3.38% $\left(\frac{27}{800}\right)$	3.26% $\left(\frac{236}{7250}\right)$	3.25% $\left(\frac{26}{800}\right)$	3.48% $\left(\frac{252}{7250}\right)$	3.50% $\left(\frac{28}{800}\right)$	3.39% $\left(\frac{246}{7250}\right)$	3.38% $\left(\frac{27}{800}\right)$	4.15% $\left(\frac{301}{7250}\right)$	4.13% $\left(\frac{33}{800}\right)$

of feature dimensions. The recognition rates of PCA features are 96.12 and 96.15% for open and closed states, respectively.

Previous works in face recognition[50–52] reported that eliminating the first one to three PCs improved performance. Because these PCs are sensitive to various illuminations, they should be removed. In this study, we eliminate the first one to six PCs to analyze the recognition effect for eye-states detection. The experiment results are presented in Table 9.

Table 9 demonstrates that eliminating the first one to three PCs will achieve the best performance rates in our data. The best performances for the PCA + LDA feature, 3.26 and 3.25%, were obtained by eliminating the first three PCs. When removing more than four PCs, the results grew worse.

In the second experiment, in order to evaluate the classification performances of several features, features are extracted by feature extraction methods, their combined features are entered into an SVM, and their classification performances are then measured. There are two kinds of features used in the second experiment. First, appearance features are extracted by PCA + LDA and statistical features are also extracted by sparseness and kurtosis. The classification performances resulted from combining these features are shown in Table 10. We can confirm that the performance was improved by combining the appearance features by PCA + LDA and the statistical features such as the sparseness and kurtosis of the histogram from the eye-edge image.

Table 11 shows the total performance of the proposed eye-states–detection method with database 2, the total testing data. As shown in Table 11, our proposed eye-states–detection method works well in both daytime and nighttime, and also for drivers wearing eyeglasses or sunglasses. Finally, we obtained results that are 1.45 and 2.91% for type I and type II errors, respectively.

Some qualitative results of the proposed eye-detection and eye-states–detection method are shown in Figs. 18 and 19. The proposed method was robust to image variations caused by sunlight, eyeglasses, sunglasses, blinking, mouth movement, and specular reflection on glasses. Here, the open eye was marked as a circle and the closed eye was marked as a rectangle. The eye region is set in proportion to the detected face region and the average size of the eye region is 40×26 pixels.

3.2.3 *Results of measuring the inattention level*

In the driver-monitoring system, it is very important to ensure that a driver is paying adequate attention to the road ahead, a must-see position when driving. We can consider two situa-

Table 10 Accuracy of the eye states detection method when using each feature or combining multiple features.

Input features	Type I error (Error rate for when an open eye is falsely accepted as a closed eye)	Type II error (Error rate for when a closed eye is falsely rejected as an open eye)
Kurtosis	8.74 (634/7250)	8.75% (70/800)
Sparseness	7.86 (570/7250)	7.88% (63/800)
Sparseness, kurtosis	6.97 (505/7250)	7.00% (56/800)
PCA	3.93% (285/7250)	4.00% (32/800)
PCA + LDA	3.26% (236/7250)	3.25% (26/800)
PCA + LDA, sparseness	2.80% (203/7250)	2.75% (22/800)
PCA + LDA, kurtosis	2.97% (215/7250)	3.00% (24/800)
PCA + LDA, sparseness, kurtosis	2.23% (162/7250)	2.25% (18/800)

Table 11 Performance of our proposed method for eye states detection in Database 2.

Time	Glass type	Type I error (Error rate for when an open eye is falsely accepted as a closed eye)	Type II error (Error rate for when a closed eye is falsely rejected as an open eye)
	Not wearing glasses	1.20% (198/16451)	3.54% (72/2032)
	Eyeglasses	3.05% (204/6699)	1.56% (16/1024)
Day	Sunglasses	0.73% (28/3830)	1.12% (5/447)
	Subtotal	1.59% (430/26980)	2.65% (93/3503)
	Not wearing glasses	0.49% (64/13185)	3.15% (62/1967)
Night	Eyeglasses	2.30% (241/10465)	3.32% (35/1055)
	Subtotal	1.29% (305/23650)	3.21% (97/3022)
Total		1.45% (735/50630)	2.91% (190/6525)

tions when a driver does not see the road ahead when driving. First, it is a situation when the driver closes his or her eyes during driving while drowsy. The second situation is when the driver's head is rotated when driving distractedly. Both of these situations are dangerous because the driver does not pay adequate attention to the road ahead. In this paper, we proposed a system to monitor these hazards simultaneously. Two measures, PERCLOS and PERLOOK, are used to detect the drowsiness and distraction level, respectively.

Figure 20 shows the result of PERCLOS measurement for drowsy sequences: the state of a driver is alert for the first 100 s, and during the remaining time, the driver exhibits drowsy driving behavior. In Fig. 20, we were able to confirm that the driver drowsiness was well detected by PERCLOS. However, if the driver exhibits distracted driving behavior, the distracted behavior cannot be detected using the PERCLOS measurement. In Fig. 21, the driver was requested to drive normally for the first 160 s and then distractedly for the remaining time. Figure 21(a) shows the result of PERCLOS measurement for distracted sequences. In Figs. 21(a) and 21(b), it is evident that distracted behaviors cannot be detected using the PERCLOS measure alone. Figure 21(a) measured by PERCLOS shows the false determination of a distracted behavior as a normal behavior. Figure 21(b) measured by PERLOOK shows that the driver distraction was well detected at ∼160 s.

The experiments to measure the drowsiness and distraction levels were executed on the image sequences in database 2. For each of the subjects, 36 drowsy video sequences and 189 distracted video sequences were recorded during the daytime and 20 drowsy video sequences and 120 distracted video sequences were recorded during the nighttime. For each of the drowsy video sequences, about 5–10 min of observation was recorded. The subjects were requested to look frontally

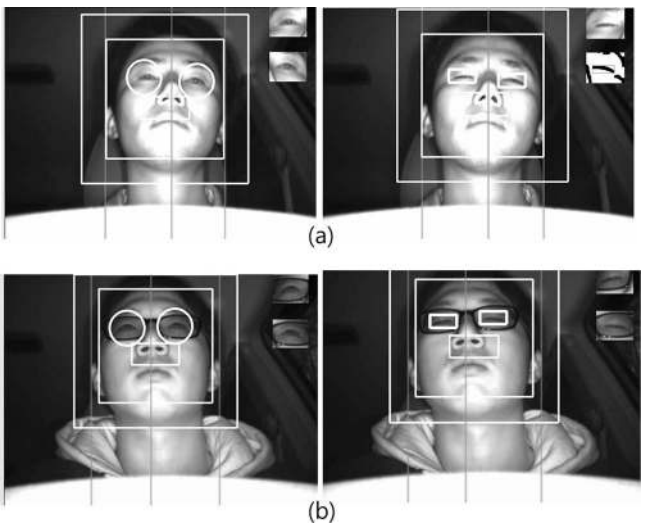


Fig. 18 Results of the proposed eye-detection and eye-states–detection method on images captured during the daytime for a driver who: (a) did not wear any glasses and (b) wore sunglasses. (Three vertical lines refer to the left, right, and center of the face.)

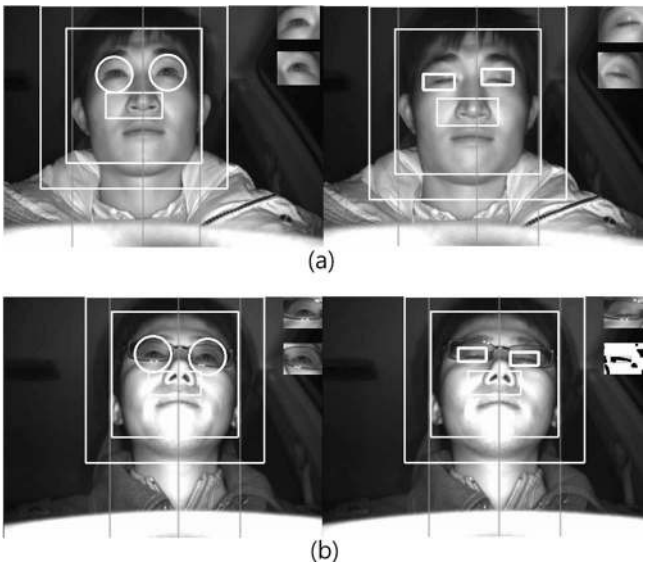


Fig. 19 Results of the proposed eye-detection and eye-states–detection method on images captured during the nighttime for a driver who: (a) did not wear any glasses and (b) wore eyeglasses.

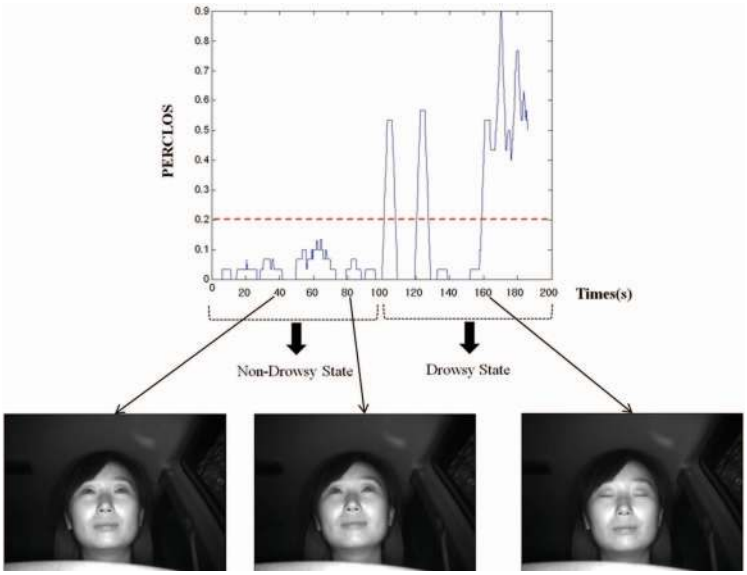


Fig. 20 PERCLOS measurement for drowsy sequences.

for 5 s and then drive normally or to drowsily drive for the remaining period. For each of the distracted video sequences, ∼1 min of observation was recorded. The subjects were requested to look frontally for 5 s and then drive normally or to distractively drive for the remaining period. Thus, for each subject, we obtained three kinds of sequences, normal, drowsy, and distracted driving video sequences. The experimental results using these video sequences are presented in Table 13. In this experiment, we achieved satisfying results. As shown in Table 13, the error rate of falsely recognizing a drowsy state or a distracted state as a normal state is 0%, while the error rate (false alarm) of falsely recognizing a normal state as a drowsy state or a distracted state is ∼2%. That is, there was no error in detecting drowsiness or distraction when the driver was actually drowsy or distracted. On the other hand, six cases of a false alarm occurred in nor-

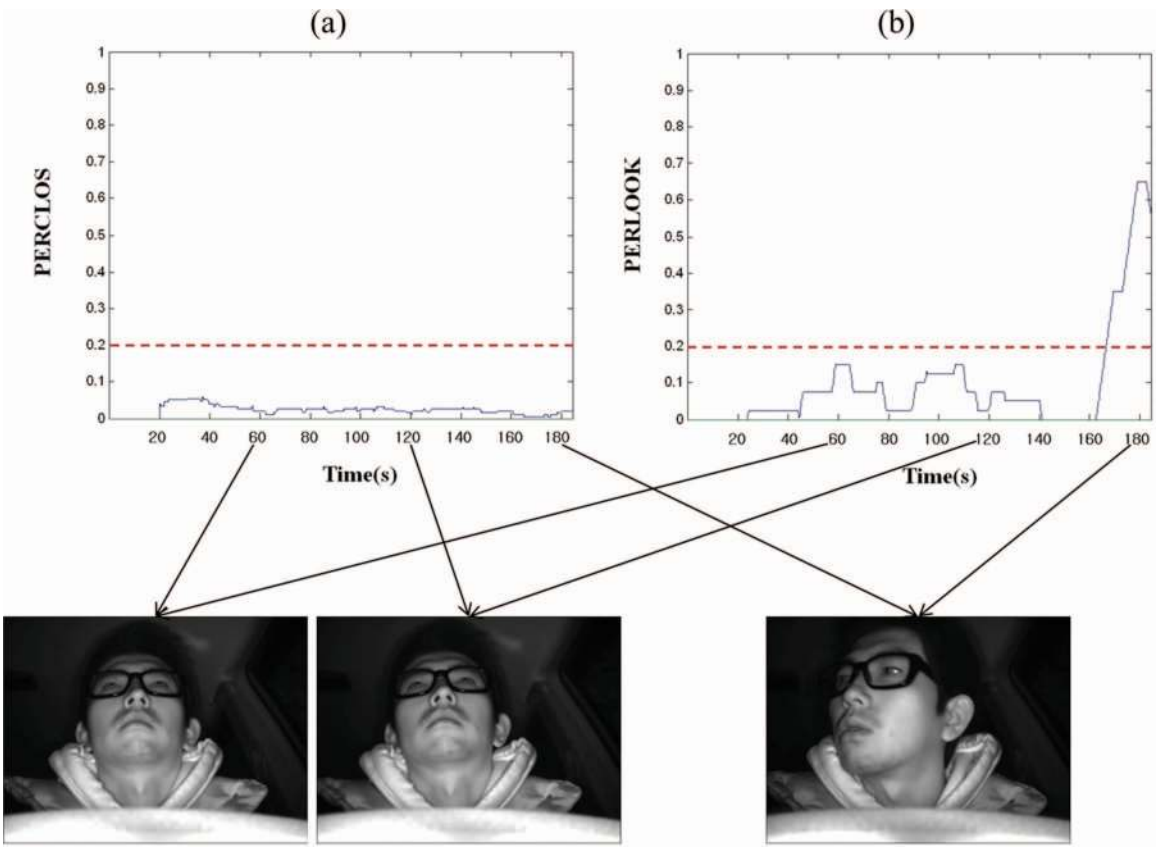


Fig. 21 PERCLOS and PERLOOK measurements for a distracted video sequence: (a) PERCLOS and (b) PERLOOK.

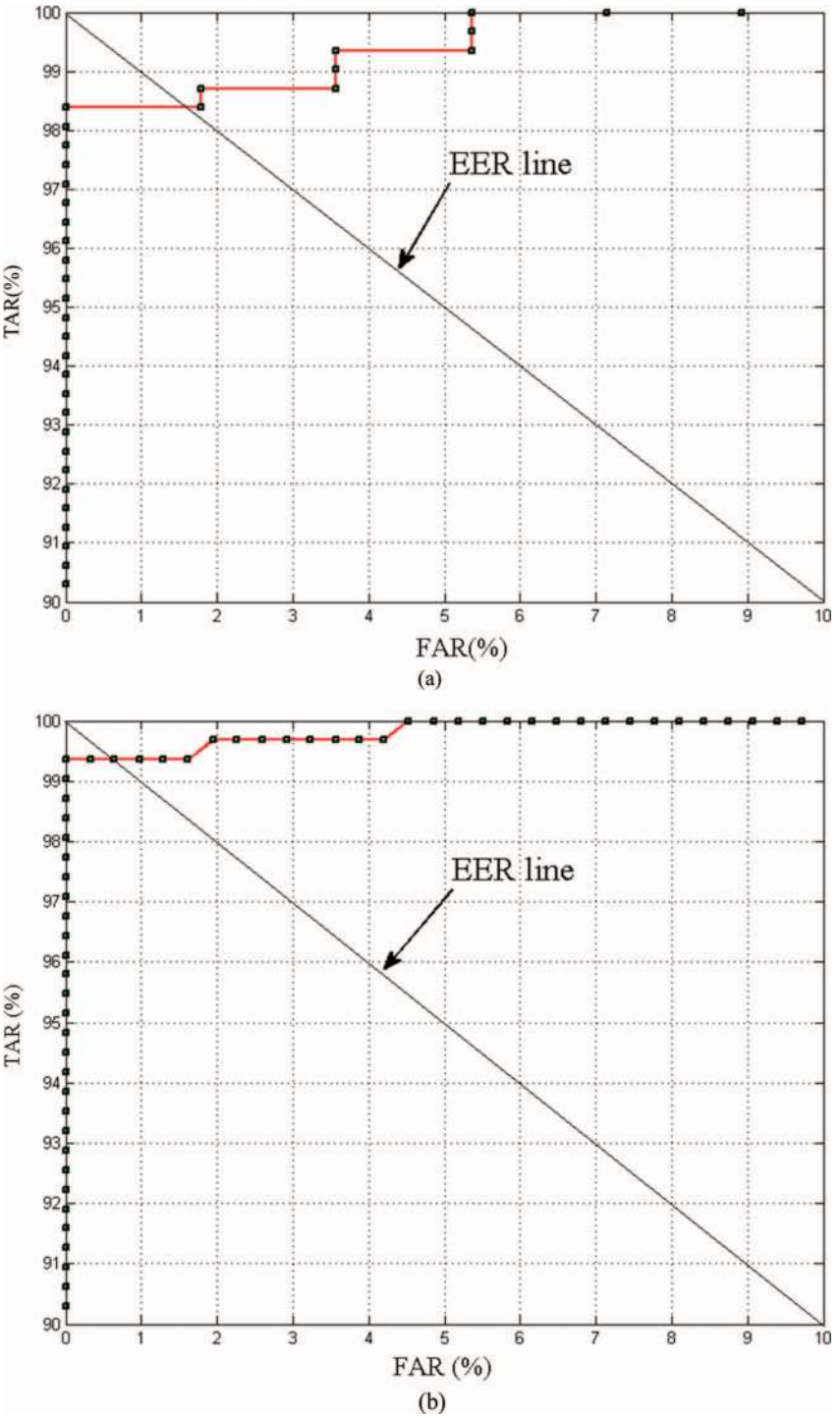


Fig. 22 ROC curves for drowsiness and distraction detection accuracy: (a) drowsiness detection and (b) distraction detection.

mal driving videos of ∼4 h, which corresponds to 1–2 false alarms per hour; these can be regarded as small errors. Failure analysis revealed that a normal state was falsely recognized as a drowsy state because the eyes were recognized as closed eyes when the driver laughed for a certain amount of time. Errors related to the driver's habit of blinking his/her eyes occur frequently. Thus, the measured PERCLOS was larger than that for normal drivers. Because a false alarm can annoy the driver, we focused on the detection of dangerous inattention. By changing the decision threshold value, the number of false alarms can be reduced at the expense of increasing the error of missing true detections.

Also, the false alarm can be reduced using the other methods, such as facial expression recognition, user-specific driver monitoring, or driving behavior information, such as monitoring vehicle speed, steering movement, lane keeping, acceleration, braking, and gear changing. These works would be considered in the future.

Figure 22 shows the ROC curve for drowsiness and distraction detection accuracy by thresholds, PERCLOS and

Fig. 23 One screen shot of actual drowsy video.

Table 12 Testing data to detect the drowsiness and distraction of a driver in a vehicle.

Time	Driver state	No. test videos
	Normal state	189
Day	Drowsy state	36
	Distracted state	189
	Normal state	120
Night	Drowsy state	20
	Distracted state	120
Total		674

PERLOOK. ROC curve is a graphical plot of the sensitivity, or true acceptance rate, versus false acceptance rate, for a binary classifier system because its discrimination threshold is varied.[70] For convenience, when the inattentive state is falsely accepted as the normal state, we define it as false-acceptance rate (FAR) in this paper. In addition, when the normal state is falsely accepted as the inattentive state, we define it as false-rejection rate (FRR). When the FAR and FRR are equal, the common value is referred to as the equal-error rate. The true-acceptance rate indicates a value subtracting FRR from 100%.

We describe a comparative experiment for measuring drowsy parameters in each actual and virtual drowsiness video. In our test, the drowsy driver–detection performance was measured using the virtual drowsiness video that was obtained when 10 people simulated drowsy driving behavior. We used a virtual drowsiness video because it is dangerous and difficult to obtain an actual drowsiness video; moreover, an open database of actual drowsiness videos is not available. Thus, it was necessary to compare the values of PERCLOS in the virtual drowsiness video with those in the actual drowsiness video. The actual drowsiness video shown in Fig. 23 was obtained when a subject was actually drowsy at dawn (at around 4:00 or 5:00 A.M.). Although this video was obtained in the laboratory, the acquisition environment (the positions of a user and the proposed device, the viewing angle of the camera, etc.) was similar to that in a car. Figure 24 shows graphs for PERCLOS in the actual and virtual drowsiness videos. It is easy to identify that these features of PERCLOS are similar. In the actual drowsiness video, the driver begins to falls asleep after 300 frames, and the average value of PERCLOS is ∼0.5254. In the virtual drowsiness video, the driver begins to falls asleep after 400 frames and the average value of PERCLOS is ∼0.5024. Therefore, we can predict that the drowsy driver–detection performance in the virtual drowsiness video will be similar to that in the actual drowsiness video.

Because it is difficult and dangerous to obtain drowsy or distracted videos in real conditions, they were not obtained. However, the 189 videos [in (daytime) normal state of Table 12] include five videos that were obtained when users were actually driving on an expressway. In the future, we believe that it would be possible to obtain a greater number

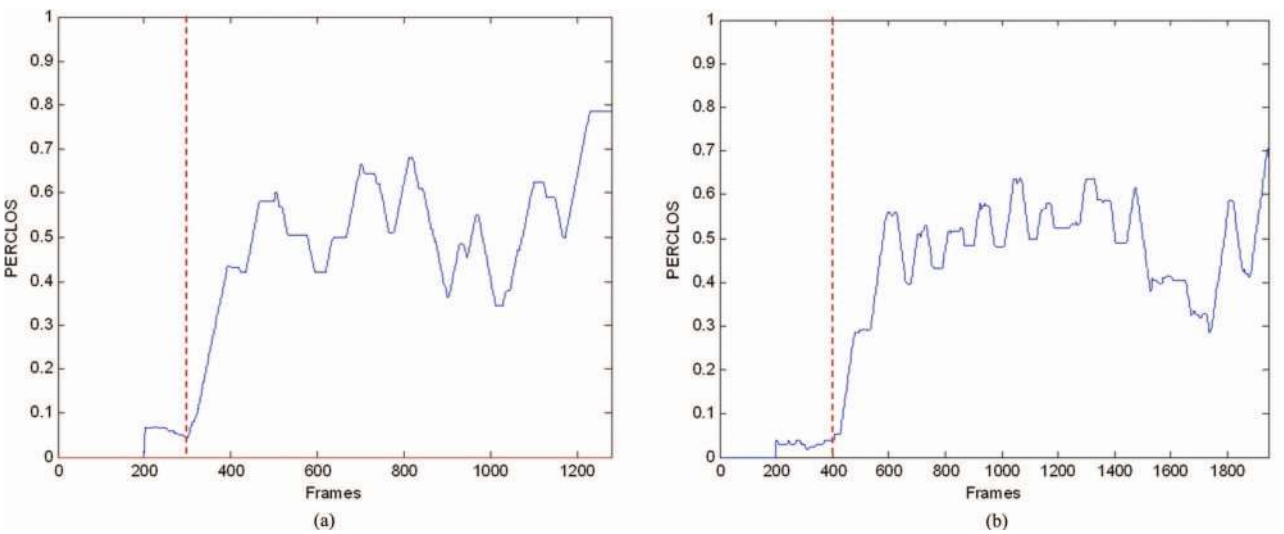


Fig. 24 Comparison between the PERCLOS values in the actual and virtual drowsiness video: (a) actual drowsiness and (b) virtual drowsiness.

Table 13 Drowsiness and distraction detection results measured from video sequences in a vehicle.

Error rates (%)					
Misrecognizing a drowsy state as a normal state	Misrecognizing a normal state as a drowsy state	Misrecognizing a distracted state as a normal state	Misrecognizing a normal state as a distracted state	Misrecognizing a drowsy state as a distracted state	Misrecognizing a distracted state as a drowsy state
0% (0/56)	1.62% (5/309)	0% (0/309)	0.64% (2/309)	1.79% (1/56)	0% (0/309)

of drowsy or distracted videos in real conditions with the support of car manufacturers.

3.2.4 *Processing time*

The proposed driver-monitoring system was efficient and worked in real time. The software was implemented using Visual C++ on an Intel Pentium M processor, 1.60 GHz, 500 MB RAM. Although not all the code optimizations were completed, the processing times are encouraging for a real-time implementation. The total processing time was 31 ms/frame, and the processing times of each step are shown in Table 14.

4 Conclusion

In this study, we proposed a driver-monitoring system that can be used for warning against driver inattention, including drowsiness and distraction. We proposed an algorithm that automatically localizes the eyes, determines whether they are opened or closed, and finally judges whether the driver is driving drowsily or distractedly. This constitutes a nonintrusive approach to detecting driver inattention without annoyance or interference in both daytime and nighttime.

The framework consists of three steps. The first step involves eye detection and tracking. In driving sequence images, adaboost, adaptive template-matching, and blob-detection methods are combined to detect the eyes automatically. Eye detection is verified using an eye-validation technique based on PCA + LDA and SVM. The proposed eye detector works well when the driver closes his or her eyes or wears eyeglasses and sunglasses. High detection rates were achieved in tests carried out on 12 people in the database during both daytime and nighttime, as well as when wearing sunglasses or eyeglasses.

Table 14 Processing time of proposed driver monitoring system.

Process	Processing time (ms/frame)
Eye adaboost	23.1
Adaptive template matching	0.7
Blob detector	1.2
Eye validation by using SVM	2.2
Total eye detection	14.5
Eye states detection	2.4
Head pose estimation[a]	14.1
Total processing time	31

[a]Reference 39.

The second step is a process for classifying eyes as opened or closed. For this purpose, appearance-based features are extracted from the detected eye image by PCA + LDA, and edge-based features are obtained on the basis of sparseness and kurtosis. The detected eyes are then classified as open or closed by an SVM using the combined features. We confirmed that performance of the combined system is better than that of a single system. Experimental results revealed that the classification error rate in the databases was $<3\%$, during both daytime and nighttime, as well as when wearing sunglasses or eyeglasses.

The third step is a process for measuring the drowsiness and distraction levels of the driver. In this process, we proposed a method for checking these two conditions simultaneously. Two measures, PERCLOS and PERLOOK, are used to calculate the drowsiness and distraction levels, respectively, and they yield satisfactory results.

In the future, we plan to conduct tests with a greater number of users and over longer periods of time in order to obtain actual drowsy and distracted behaviors. From the viewpoint of gaze estimation, we need to develop a precise gaze-estimation method using both head-pose and eye-movement information. In the drowsy driver–detection method, facial expression recognition must be employed to distinguish eyes closed due to fatigue from those closed due to vigorous laughter. Moreover, because eye size, eye blinking frequency, and driving habits vary among drivers, we need to develop a driver-specific monitoring system using an online training method. This system will exhibit better performance if it is integrated with other sensors, such as the accelerator, steering wheel, and lane-position sensor. Finally, an integrated driving safety system will be more effective when combined with a pedestrian-recognition system and a lane-keeping support system.

Acknowledgments

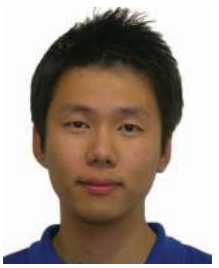
This work was supported in part by Mando Corporation Ltd. And, in part, by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (Grant No. 2011-0015321).

References

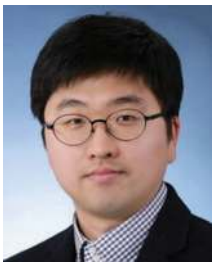
1. M. R. Rosekind, E. L. Co, K. B. Gregory, and D. L. Miller, "Crew factors in flight operations XIII: a survey of fatigue factors in corporate/executive aviation operations," NASA, Ames Research Center, NASA/TM-2000-209 610 (2000).
2. Awake Consortium (IST 2000-28062), "System for effective assessment of driver vigilance and warning according to traffic risk estima-

tion (AWAKE)," September 2001–2004. <http://www.awake-eu.org> (April 16, 2011).
3. U.S. Department of Transportation, "2007 FARS/GES annual report (final ed.)," <http://www.nhtsa.dot.gov/portal/site/nhtsa/menuitem.6a6eaf83cf719ad24ec86e10dba046a0/> (April 16, 2011).
4. W. Wierwille, L. Tijerina, S. Kiger, T. Rockwell, E. Lauber, and A. Bittne, "Final report supplement—heavy vehicle driver workload assessment—task 4: review of workload and related research," USDOT, Tech. Rep. No. DOT HS 808 467(4) (1996).
5. H. Saito, T. Ishiwaka, M. Sakata, and S. Okabayashi, "Applications of driver's line of sight to automobiles—what can driver's eye tell," in *Proc. of Vehicle Navigation and Information Systems Conf.*, Yokohama, pp. 21–26 (1994).
6. H. Ueno, M. Kaneda, and M. Tsukino, "Development of drowsiness detection system," in *Proc. of Vehicle Navigation and Information Systems Conf.*, Yokohama, pp. 15–20 (1994).
7. S. Boverie, J. M. Leqellec, and A. Hirl, "Intelligent systems for video monitoring of vehicle cockpit," in *Proc. of Int. Congress and Exposition ITS, Advanced Controls and Vehicle Navigation Systems*, pp. 1–5 (1998).
8. M. Kaneda, H. Ueno, and M. Tsukino, "Development of a drowsiness warning system," in *Proc. of 11th Int. Conf. on Enhanced Safety of Vehicle*, Munich (1994).
9. R. Onken, "Daisy, an adaptive knowledge-based driver monitoring and warning system," in *Proc, of Vehicle Navigation and Information Systems Conf.*, Yokohama, pp. 3–10 (1994).
10. T. Ishii, M. Hirose, and H. Iwata, "Automatic recognition of driver's facial expression by image analysis," *Journal of the Society of Automotive Engineers of Japan*, Vol. 41, pp. 1398–1403 (1987).
11. K. Yammamoto and S. Higuchi, "Development of a drowsiness warning system," *Journal of the Society of Automotive Engineers of Japan*, Vol. 46, pp. 127–133 (1992).
12. 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).
13. R. Grace, V. E. Byrne, D. M. Bierman, and J. M. Legrand, "A drowsy driver detection system for heavy vehicles," in *Proceedings of Digital Avionics Systems Conference*, 17th DASC, Vol. 2, pp. I36/1–I36/8 (1998).
14. D. Cleveland, "Unobtrusive eyelid closure and visual of regard measurement system," Conference on Ocular Measures of Driver alertness, April (1999).
15. J. Fukuda, K. Adachi, M. Nishida, and E. Akutsu, "Development of driver's drowsiness detection technology," *Toyota Tech. Rev.* **45**, 34–40 (1995).
16. J. H. Richardson "The development of a driver alertness monitoring system," in *Fatigue and Driving: Driver Impairment, Driver Fatigue and Driver Simulation*, L. Harrtley, Ed., Taylor & Francis, London (1995).
17. T. D'Orazio, M. Leo, C. Guaragnella, and A. Distante, "A visual approach for driver inattention detection," *Pattern Recogn.* **40**, 2341–2355 (2007).
18. E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(4), 607–626 (2009).
19. D. W. Hansen and Q. Ji, "In the eye of the beholder: a survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 478–500 (2010).
20. A. Hattori, S. Tokoro, M. Miyashita, I. Tanakam, K. Ohue, and S. Uozumi, "Development of forward collision warning system using the driver behavioral information," presented at 2006 SAE World Congress, Detroit, Michigan (2006).
21. E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, "Head pose estimation for driver assistance systems: a robust algorithm and experimental evaluation," in *Proc. of 10th Int. IEEE Conf. Intelligent Transportation Systems*, pp. 709–714 (2007).
22. P. Smith, M. Shah, and N. da Vitoria Lobo, "Determining driver visual attention with one camera," *IEEE Trans. Intell. Transp. Syst.* **4**(4), 2058–2218 (2003).
23. J. Y. Kaminski, D. Knaan, and A. Shavit, "Single image face orientation and gaze detection," *Mach. Vis. Appl.* **21**, 85–98 (2009).
24. J. C. Stutts, D. W. Reinfurt, L. Staplin, and E. A. Rodgman, "The role of driver distraction in traffic crashes," AAA Foundation for Traffic Safety, Washington, DC, (2001).
25. K. Torkkola, N. Massey, and C. Wood, "Driver inattention detection through intelligent analysis of readily available sensors," in *Proc. of IEEE Conf. on Intelligent Transportation Systems*, Washington, DC, pp. 326–331 (2004).
26. G. Yang, Y. Lin, and P. Bhattacharya, "A driver fatigue recognition model using fusion of multiple features," in *Proc. of IEEE Int. Conf. on Syst.*, Man and Cybernetics, Hawaii Vol. 2, pp. 1777–1784 (2005).
27. S. Park and M. M. Trivedi, "Driver activity analysis for intelligent vehicles: issues and development framework," in *Proc. of IEEE Intelligent Vehicles Symp.*, Las Vegas, pp. 795–800 (2005).
28. E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. S. Bartlett, and J. R. Movellan, "Drowsy driver detection though facial movement analysis," *IEEE Int. Conf. on Computer Vision–Human Computer Interaction*, pp. 6–18 (2007).
29. M. Saradadevi and P. Bajaj, "Driver fatigue detection using mouth and yawning analysis," *Int. J. Comput. Sci. Netw. Security* **8**(6), 183–188 (2008).
30. L. M. Beragasa, J. Nuevo, M. A. Sotelo, R. Barea, and E. Lopez, "Real-time system for monitoring driver vigilance," *IEEE Trans. Intell. Transport. Syst.* **7**(1), 1524–1538 (2006).
31. R. Senaratne, D. Hardy, B. Vanderaa, and S. Halgamuge, "Driver fatigue detection by fusing multiple cues," *Lect. Notes Comput. Sci.* **4492**, 801–809 (2007).
32. K. Kircher, A. Kircher, and F. Claezon, "Distraction and drowsiness—a field study," Tech. Rep., VTI (The Swedish National Road and Transport Research Institute) (2009).
33. S. J. Lee, J. Jo, H. G. Jung, K. R. Park, and J. Kim, "Real-time gaze estimator based on driver's head orientation for forward collision warning system," *IEEE Trans. Intell. Transp. Syst.* **12**(1), 254–267 (2011).
34. Specification of illuminator, <http://roithner-laser.com/multi/led_multi_to.html≥ (July 25, 2011).
35. Camera (EC650), <http://www.pyramidimaging.com/specs/Prosilica/700009AA-EC650%20User%20Manual.pdf> (August 25, 2011).
36. Camera lens (ML0614), <http://www.phoeniximaging.com/ml-0614.htm≥ (July 25, 2011).
37. Narrow bandpass interference filter (NT43-148), <http://www.edmundoptics.com/onlinecatalog/displayproduct.cfm?productid=3198&PageNum=5&Sort=displayOrder&Order=asc#products> (July 25, 2011).
38. SUV (GIA motors Sportage), <http://www.kia.com/#/sportage/explore/> (August 25, 2011).
39. Sedan (Hyundai Motors Grandeur), <http://grandeur.hyundai.com≥ (August 25, 2011).
40. T. Ito, S. Mita, K. Kozuka, T. Nakano, and S. Yamamoto, "Driver blink measurement by the motion picture processing and its application to drowsiness detection," in *Proc. of Int. Conf. Intelligent Transportation Systems*, Singapore, pp. 168–173 (2002).
41. K. Ohue, Y. Yamada, S. Uozumi, S. Tokoro, A. Hattori, and T. Hayashi, "Development of a new pre-crash safety system," SAE Tech. Paper Series, Doc. No. 2006-01-1461 (2006).
42. Specification of motion tracking device called Patriot, <http://www.polhemus.com/polhemus_editor/assets/Tech%20compare%20with%20ResolutionvsRange%20graphs.pdf> (February 15, 2011).
43. W. J. Scheirer, A. Rocha, B. Heflin, and T. E. Boult, "Difficult detection: a comparison of two different approaches to eye detection for unconstrained environments," in *Proc. of IEEE 3rd Int. Conf. on Biometrics: Theory, Applications, and Systems*, pp. 1–8 (2009).
44. C. Whitelam, Z. Jafri, and T. Bourlai, "Multispectral eye detection: a preliminary study," in *Proc. of 2010 Int. Conf. on Pattern Recognition*, pp. 209–212 (2010).
45. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of IEEE CVPR*, Vol. 1, pp. I-511–I-518 (2001).
46. S. D Wei and S. H Lai, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," *IEEE Trans. Image Process.* **17**(11), 2227–2235 (2008).
47. Open source of blob detection, <http://opencv.willowgarage.com/wiki/cvBlobsLib#Algorithm> (July 30, 2011).
48. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., pp. 523–532, Prentice-Hall, Englewood Cliffs, NJ (2002).
49. G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, 974–989 (1999).
50. P. N. Bellhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 711–720 (1997).
51. A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 228–233 (2001).
52. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuro-sci.* **3**, 71–86 (1991).
53. H. Deng, L. Jin, L. Zhen, and J. Huang, "A new facial expression recognition method nased on local gabor filter bank and PCA plus LDA," in *Proc. of Int. Conf. on Intelligent Computing*, August 23–26, Hefei, China, pp. 6–7 (2005).
54. V. Vapnik, "Support vector estimation of functions," in *Statistical Learning Theory*, pp. 375–570, Wiley, Hoboken, NJ (1998).
55. C. Hsu, C. Chang, and C. Lin. "A practical guide to support vector classification," Tech. Rep., Department of Computer Science and Information Engineering, National Taiwan University, Taipei, <http://w.csie.org/˜cjlin/papers/guide/guide.pdf> (March. 23, (2011).
56. L. Lang and H. Qi, "The study of driver fatigue monitor algorithm combined PERCLOS and AECS," in *Proc. of Int. Conf. on Comput. Sci. and Software Eng.*, pp. 349–352 (2008).

57. L. Harley, T. Horberry, N. Mabbott, and G. Krueger, "Review of fatigue detection and prediction technologies," National Road Transport Commission, (2000).
58. R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, pp. 78–84 McGraw-Hill, New York (1995).
59. P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learning Res.* **5**, 1457–1469 (2004).
60. A. Hyvarinen and E. Oja, "Independent component analysis," *Neural Netw.* **13**(4–5), 411–430 (2000).
61. Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging* **8**(5), 357–377 (2002).
62. D. F. Dinges, M. 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," U.S. Department of Transportation: National Highway Traffic Safety Administration, DOT HS 808 762 (1998).
63. Smart Eye Pro, <http://www.smarteye.se> (25 July 2011).
64. S. Kawato and J. Ohya, "Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes," in *Proc. of 4th Int'l IEEE Conf. Automatic Face and Gesture Recognition*, pp. 40–45 (2000).
65. B. Ma, S. Shan, X. Chen, and W. Gao, "Head yaw estimation from asymmetry of facial appearance," *IEEE Trans. Syst., Man, Cybern. Part B* **38**(6), 1501–1512 (2008).
66. P. Watta, S. Lakshmanan, and Y. Hou, "Nonparametric approaches for estimating driver pose," *IEEE Trans. Vehicular Technol.* **56**(4), 2028–2041 (2007).
67. J. Wu and M. M. Trivedi, "A two-stage head pose estimation framework and evaluation," *Pattern Recogn.* **41**, 1138–1158 (2008).
68. E. Murphy-Chutorian and M. M. Trivedi, "HyHOPE: hybrid head orientation and position estimation for vision-based driver head tracking," in *Proc. of IEEE Intell. Vehicles Symp.*, pp. 512–517 (2008).
69. J. P. Batista, "A real-time driver visual attention monitoring system," *Lect. Notes Comput. Sci.* **3522**, 200–208 (2005).
70. N. K. Ratha and V. Govindaraju, *Advances in Biometrics: Sensors, Algorithms and System*, Springer, New York (2008).



Jaeik Jo received the BS in electrical and electronic engineering in 2008 from Yonsei University, Seoul, Korea, where he is currently working toward the MS-PhD joint degree. His current research interests include driver monitoring systems, 3-D face reconstruction, biometrics, pattern recognition, and computer vision.



Sung Joo Lee received the BS in electrical and electronic engineering and the MS in biometric engineering in 2004 and 2006, respectively, from Yonsei University, Seoul, Korea, where he received the PhD in electrical and electronic engineering in 2011. In 2011, he joined the Mobis Co., Ltd. where he is currently an assistant research engineer. His current research interests include moving object detection, driver monitoring system, biometrics, pattern recognition, and computer vision.

Ho Gi Jung received the BE, ME, and PhD in electronic engineering from Yonsei University, Seoul, Korea, in 1995, 1997, and 2008, respectively. He was with MANDO Corporation Global R&D H.Q., from 1997 to April 2009. He developed environmental recognition systems for intelligent parking assist system, collision warning and avoidance system, and active pedestrian protection system. From May 2009 to February 2011, he was with Yonsei University as a full-time researcher and research professor. He researched computer vision applications for intelligent surveillance systems and biometric systems. Since March 2011, he has been with Hanyang University as an assistant professor. He is researching recognition systems for intelligent vehicles. His interests are recognition system for intelligent vehicle, next generation vehicle, computer vision applications, and pattern recognition applications.

Kang Ryoung Park received his BS and MS in electronic engineering from Yonsei University, Seoul, Korea, in 1994 and 1996, respectively. He also received his PhD in computer vision from the Department of Electrical and Computer Engineering, Yonsei University, in 2000. He was an assistant professor in the Division of Digital Media Technology at Sangmyung University from March 2003 to February 2008. He has been an assistant and associate professor in the Division of Electronics and Electrical Engineering at Dongguk University since March 2008. He is also a research member of BERC. His research interests include computer vision, image processing, and biometrics.

Jaihie Kim received the BS in electronic engineering from Yonsei University, Republic of Korea, in 1979, the PhD in electrical engineering from the Case Western Reserve University, USA in 1984. Since 1984, he has been a professor in the School of Electrical and Electronic Engineering, Yonsei University. Currently, he is the Director of the Biometric Engineering Research Center in Korea, the Chairman of the Korea Biometric Association, a member of the National Academy of Engineering of Korea, and in 2008 he was the President of the IEEK (Institute of Electronics Engineers of Korea) and now an Emeritus President of the IEEK. His general research interests are biometrics, pattern recognition and computer vision. Specifically, some of his recent research topics include touchless fingerprint recognition, fake fingerprint detection, fake face detection, 2D-to-3D face conversion, face age estimation/synthesis. He was the author of many international technical journals which could be found at 'http://cherup.yonsei.ac.kr/'.