國立交通大學

電機與控制工程研究所

碩士論文

利用模糊積分於昏睡偵測

Drowsiness Detection Using Fuzzy Integral Based Information Fusion

研 究 生：張欽漢

指導教授：張志永

中華民國九十四年七月

利用模糊積分於昏睡偵測

Drowsiness Detection Using Fuzzy Integral Based Information Fusion

研 究 生：張欽漢　　Student：Chin-Han Chang

指導教授：張志永　　Advisor：Jyh-Yeong Chang

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master in

Electrical and Control Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國 九十四 年 七 月

利用模糊積分於昏睡偵測

學生: 張欽漢　　　　　　指導教授: 張志永博士

國立交通大學電機與控制工程研究所

摘要

當人們在工作中或是在駕駛的環境中，打瞌睡是造成意外事故最常見的因素之一，為了避免類似的意外發生，我們提出了一個非侵入式的昏睡偵測演算法來避免因為侵入式的方法而造成受測者的不舒適感。本文是根據駕駛者眼睛閉合的程度與眨眼頻率兩種偵測資訊來判斷出受測者的昏睡程度。我們首先研究觀察的時間間隔對於利用眼睛閉合程度與眨眼頻率來偵測昏睡狀態的影響，並且找出最佳的觀察時間間隔。為了提高偵測的準確度，我們利用模糊積分的觀念，發展出上述兩種偵測資訊整合的技術，此技術可解決兩種偵測資訊在判斷上發生衝突與模稜兩可的情況。我們也將本文所提出的方法與眼睛閉合程度、眨眼頻率二種方法做比較，根據結果顯示，我們提出的方法的準確率高達 95.1%。我們也將所提的方法應用在偵測駕駛者的精神狀態，由結果證明，此實驗是非常成功且有效率的。另一方面，有許多駕駛者有戴墨鏡的習慣，尤其是在夏天，所以我們針對墨鏡的區域來做影像增強以去除墨鏡對眼睛偵測與昏睡偵測的干擾。

Drowsiness Detection Using Fuzzy Integral Based Information Fusion

STUDENT: CHIN-HAN CHANG ADVISOR: Dr. JYH-YEONG CHANG

Institute of Electrical and Control Engineering
National Chiao-Tung University

ABSTRACT

Drowsiness is often reported to be one of the most important factors causing danger on various occasions such as work fields and vehicle driving. To avoid this danger, we thesis in this thesis a non-intrusive vision-based drowsiness detection algorithm. Visual techniques are adopted such that we can prevent people from feeling uncomfortable due to intrusive signal acquisition. In this study, we utilize the Long Duration Blink Frequency (LDBF) and the PERcentage of eyelid CLOSure (PERCLOS) as features of drowsiness detection, which are commonly used in visual drowsiness detection system. We first investigated the effect of Observing Time Interval (OTI) on the separability of sample distributions of LDBF and PERCLOS under drowsy and conscious states to select the best OTI. In order to increase the accuracy of drowsiness detection, we use fuzzy integral to combine two different information sources from LDBF and PERCLOS features. The proposed fuzzy integral approach can resolve the conditional unreliability and uncertainty encountered in using LDBF or PERCLOS singly. To show the superiority of our method in drowsiness detection accuracy, we compared our proposed method to LDBF and PERCLOS, respectively. According to the experiment result, the proposed algorithm has the best average detection accuracy of 95.1%. In practice, we also implemented our algorithm to determine people's vigilance in a driver monitor and warning system. The test in driver drowsiness detection and warning was successful and satisfactory. On the other hand, many drivers have the needs and habits to wear sunglasses, especially in summer. We also develop image enhancement techniques to eliminate the effect caused by sunglasses in eye detection and drowsiness detection.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Jyh-Yeong Chang. Without his patient guidance and inspiration during the two years, it is impossible for me to complete the thesis. In addition, I am thankful to all my Lab members for their discussion and suggestion.

Finally, I would like to express my deepest gratitude to my family and my Lord. Without their strong support, I could not go through the two years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Driver fatigue is a significant factor in a large number of vehicle accidents. The ever-increasing number of traffic accidents in the US due to a diminished driver's vigilance level has become a problem of serious concern to society. Drivers with a diminished vigilance level suffer from a marked decline in their abilities of perception, recognition, and vehicle control, and therefore pose serious danger to their own life and the lives of other people. Statistics show that a leading cause for fatal or injury-causing traffic accidents is due to drivers with a diminished vigilance level. In the trucking industry, 57% fatal truck accidents are due to driver fatigue. It is the number 1 cause for heavy truck crashes. Seventy percent of American drivers report driving fatigued. With the ever-growing traffic conditions, this problem will further deteriorate. For this reason, developing systems actively monitoring a driver's level of vigilance and alerting the driver of any insecure driving conditions is essential to prevent accidents.

1.1 Previous Work

There are various methods that can be used to detect drowsy driving. These detection methods can be broadly divided into two categories. One category makes use of information about the driver such as psychological signals; the other category uses information about the vehicle such as the operation of vehicle motions. Typical examples of physiological signals include brain waves, eye electric potential, heart rate, and skin electric potential. Driving operations such as the movement of the steering wheel or the operation of the accelerator or brake pedal can be sensed, and characteristics for identifying drowsy driving can be detected. Detectable vehicle motions stemming from drowsy driving include the vehicle speed, lateral acceleration, yaw rate, and lateral displacement.

Measurement of physiological signals provides a rather accurate means of detecting drowsy driving, but it is necessary to attach sensors directly to the driver's body. Methods based on the use of vehicle information offer the advantage of noncontact detection, but they are subject to severe limitations depending on the characteristics of the vehicle or the driving environment.

Taking these issues into account, it was decided to use image processing technology [1]-[3] to detect drowsy driving. The reason for this choice is that it offers the advantages of not causing the driver any discomfort or annoyance and of providing high detection accuracy.

1.2 Our Approach

A flowchart of the major functions of the drowsiness detection system is shown in Fig. 1.1. The functions of the system can be broadly divided into an eye and eyeball detection function, comprising the first half of the processing routine, and a drowsiness detection function, comprising the second half of this thesis. Our experimental results show the feasibility of our approach and the validity for the method is extended to drivers wearing sunglasses.

1.2.1 Eyeball Detection Module

We uses color as a feature [4] to identify a human face in an image. This is feasible because human faces have a special color distribution that differs significantly (although not entirely) from those of the background objects. After locating a face area, we try to extract the eye region such that we could measure eye features for drowsiness detection. A brief explanation is given here of the eyeball detection procedure [5]-[7]. After inputting a facial image, preprocessing is first performed to binarize the image and remove noise, which makes it possible for the image to be accepted by the image processor.

The maximum width of the face is then detected so that the right and left edges of the face can be identified. After that, the vertical position of each eye is detected independently within an area defined by the center line of the face width and lines running through the outermost points of the face. On that basis, the area in which each eye is present is determined. The degree of eye openness is output with the establishment of the area of eye presence. That value is used to determine whether the eyes are open or closed.

1.2.2 Drowsiness Detection Module

(1) Determination of whether eyes are open/closed

A window is defined on the basis of the eye. The area of the eye indicates the degree of eye openness and is used as the basis to determine whether the eyes are open or closed.

(2) Criterion for estimating eye open/closed state

A threshold value is established for each driver to estimate whether the person's eyes are open or closed. That criterion [8]-[9] is based on the percentage of the eye closure observed for a certain time interval.

(3) Criterion for estimating the eye blinking rate

As the level of alertness drops, rapid blinking gives way to the appearance of long intervals when the eyes are closed, which provided a basis for detecting drowsiness by Long Duration Blink Frequency (LDBF).

(4) On determining alertness level

To integrate into knowledge from the two sources (PERCLOS and long duration blink), a fuzzy integral based method [10]-[12] will be devised to determine the level of alertness.

1.3 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, the face location detection and the eye position detection will first be described. Then, the method of determining whether the driver’s eyes are open or closed will be introduced. In Chapter 3, we will introduce two drowsiness detection measures—PERCLOS and long duration blink frequency, and to combine PERCLOS and the long duration blink frequency using fuzzy integral is also described. In Chapter 4, we show our experimental results and compare the detection accuracy of our proposed method with that of PERCLOS and long duration blink frequency, respectively. Finally, some concluding remarks will be drawn in Chapter 5.

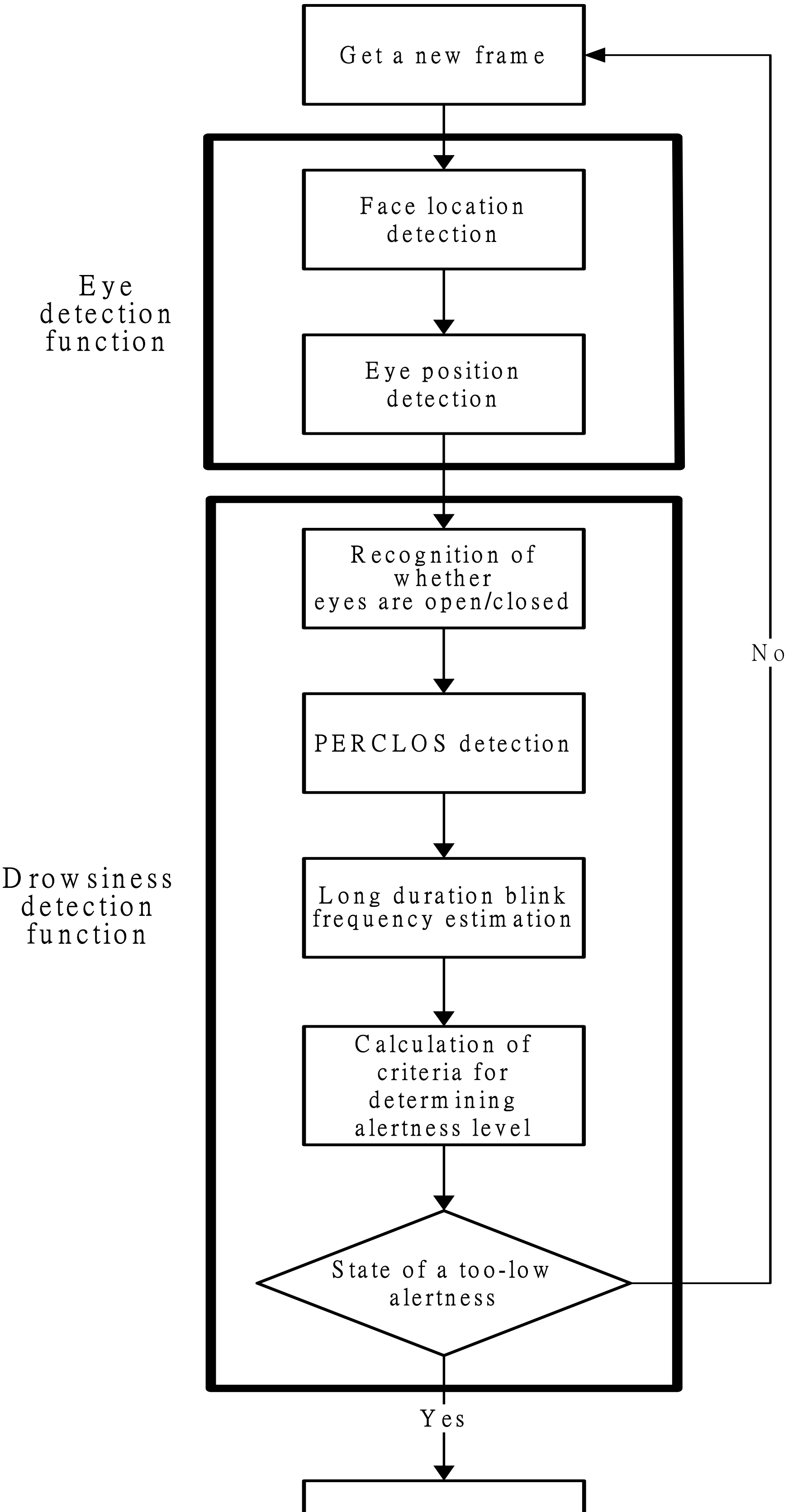

Get a new frame
Eye detection function
Face location detection
Eye position detection
Drowsiness detection function
Recognition of whether eyes are open/closed
PERCLOS detection
Long duration blink frequency estimation
Calculation of criteria for determining alertness level
State of a too-low alertness
No
Yes
Warning

Fig. 1.1. Flow chart of system.

Chapter 2

Feature Extraction for Drowsiness Detection

2.1 Introduction to Feature Extraction

For drowsiness detection, it is necessary to preprocess the captured image sequences for a subject. The preprocessing operation includes face detection and eye detection aiming for eye area detection to estimate the degree of eye openness.

2.2 Face Segmentation Algorithm

The algorithm in [4] is an unsupervised segmentation algorithm, and hence no manual adjustment of any design parameter is needed in order to suit any particular input image. The only principal assumption is that the person’s face must be present in the given image, since we are locating and not detecting whether there is a face. The algorithm we use is consists of four stages, as outlined in Fig. 2.1.

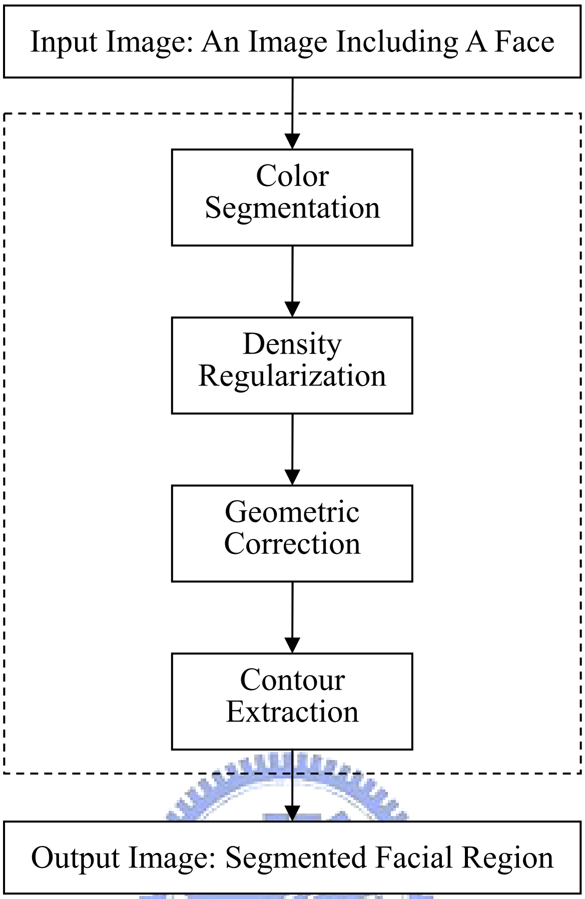


Fig. 2.1. Outline of face-segmentation algorithm.

A. Color Segmentation

The first stage of the algorithm is to classify the pixels of the input image to skin region and non-skin region. To do this, we reference a skin-color reference map in YC_rC_b color space. We denote R_{C_r} and R_{C_b} as the respective ranges of C_r and C_b values that correspond to skin color, and they have tested are $R_{C_r} = [133, 173]$, and $R_{C_b} = [77, 127]$. With the skin-color reference map, we got the color segmentation result O_A as

$$O_A(x, y) = \begin{cases} 1, & \text{if } [C_r(x, y) \in R_{C_r}] \bigcap [C_b(x, y) \in R_{C_b}] \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where x = 0, … , M/2–1 and y = 0, … , N/2–1 and M, N are the height and width of the picture respectively. An example to illustrate the classification of the original image Fig. 2.2 is shown in Fig. 2.3.

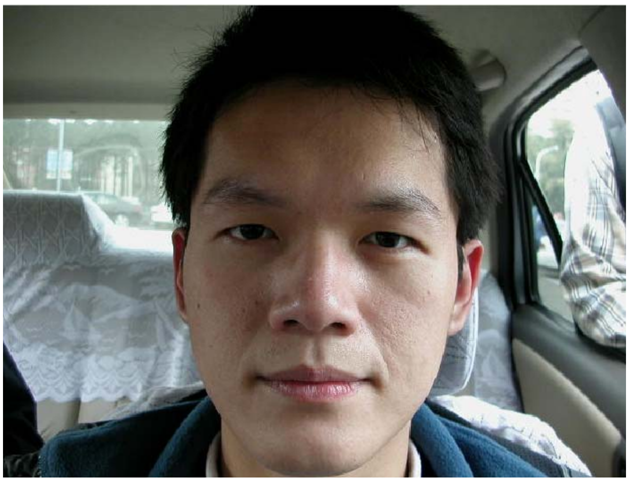


Fig. 2.2. Original image.

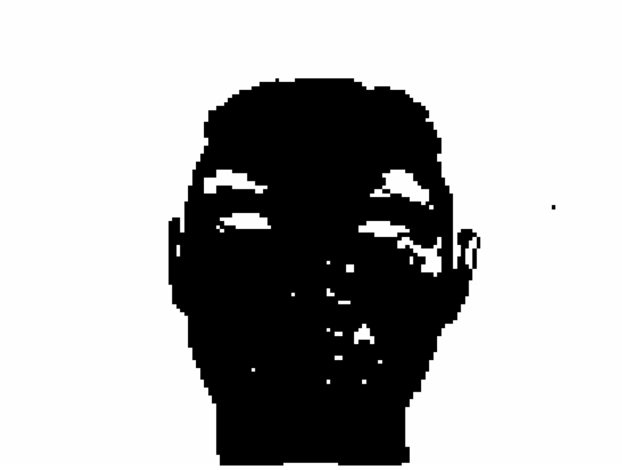


Fig. 2.3. Image after filtered by skin-color map in stage A.

B. *Density Regularization*

This stage considers the bitmap produced by the previous stages to contain the facial region that is corrupted by noise. The noise may appear as small holes on the facial region due to undetected facial features such as eyes, mouth, even glasses, or it may also appear as objects with skin-color appearance in the background scene.

Therefore, this stage performs simple morphological operations such as dilation to fill in any small hole in the facial area and erosion to remove any small object in the background area. A density map is calculus as follows.

$$D(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} O_A(4x + i,\ \ 4y + j) \tag{2.2}$$

According to the density value, we classify each point into three types, namely, zero ($D = 0$), intermediate ($0 < D < 16$), and full ($D = 16$). The density map of an example with three density classifications is depicted in Fig. 2.4. The point of zero density is shown in white, intermediate density in green, and full density in black.

Once the density map is derived, we can then begin the process that we termed as density regularization. This involves the following three steps.

1) Discard all points at the edge of the density map, i.e., set $D(0, y) = D(M/8–1, y) = D(x, 0) = D(x, N/8–1)$ for all $x = 0, \dots, M/8–1$ and $y = 0, \dots, N/8–1$.
2) Erode any full-density point (i.e., set to zero) if it is surrounded by less than five other full-density points in its local 3×3 neighborhood.
3) Dilate any point of either zero or intermediate density (i.e., set to 16) if there are more than two full-density points in its local 3×3 neighborhood.

After this process, the density map is converted to the output bitmap of stage B as

$$O_B(x, y) = \begin{cases} 1, & \text{if } D(x, y) = 16 \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

for all x = 0, …, M/8–1 and y = 0, …, N/8–1.

The result of the previous example is displayed in Fig. 2.5.

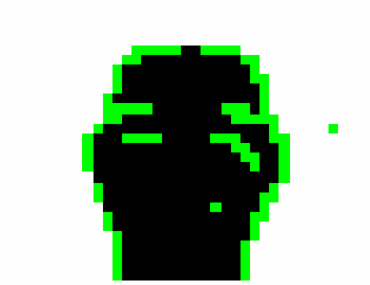


Fig. 2.4. Density map after classified to three classes.

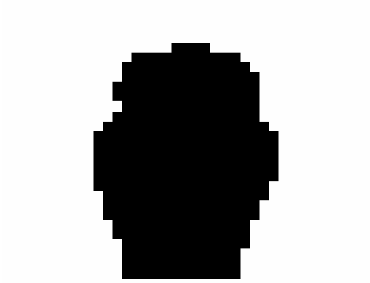


Fig. 2.5. Image produced by stage B.

C. Geometric Correction

We performed a horizontal and vertical scanning process to identify the presence of any odd structure in the previously obtained bitmap, $O_B(x,y)$, and subsequently removed it. A pixel in $O_B(x,y)$ with the values of one will remain as a detected pixel if there are more than three other pixels, in its local 3x3 neighborhood, with the same value. At the same time, a pixel in $O_B(x,y)$ with a value of zero will be reconverted to a value of one (i.e., as a potential pixel of the facial region) if it is surrounded by more than five pixels, in its local 3x3 neighborhood, with a value of one.

We then commence the horizontal scanning process on the “filtered” bitmap. We search for any short continuous run of pixels that are assigned with the value of one. Any group of less than four horizontally connected pixels with the value of one will be eliminated and assigned to zero. A similar process is then performed in the vertical direction. As a result the output bitmap of this stage should contain the facial region with minimal or no noise, as demonstrated in Fig. 2.6.

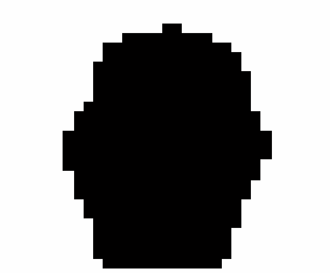


Fig. 2.6. Image produced by stage C.

D. Contour Extraction

In this final stage, we convert the output bitmap of stage C back to the extracted face region. To achieve the increase in spatial resolution, we utilize the edge information that is already made available by the color segmentation in stage A. The representative output bitmap of this final stage of the algorithm is shown in Fig. 2.7.

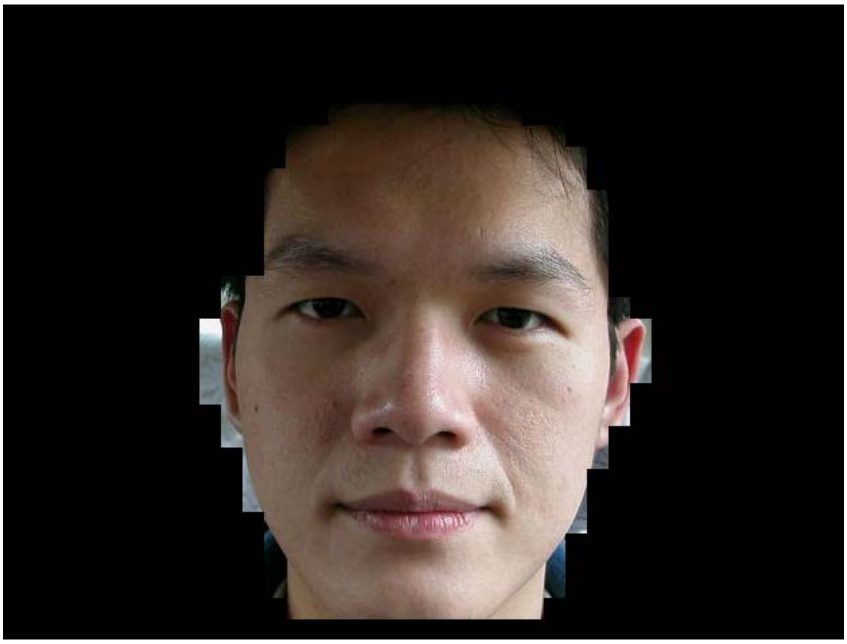


Fig. 2.7. Image produced by stage D.

2.3 Eye Position Detection

After locating a face area, we try to extract the eye position such that we could measure eye features for drowsiness detection.

In order to determine the position where the eyes are present, the maximum width of the subject's face on the captured image must be estimated first. Based on the fact that the eyeball is about one-fourth of the facial width, we can easily obtain the lateral eye position on the face. Next, we explain how to determine the vertical positions of the eyes. To this end, we draw two vertical lines, or more if necessary, around one-fourth of the facial width: one line X_a is not across the bulb of the eye and another line X_b is across the bulb of the eye as shown in Fig. 2.8. Fig. 2.9 depicts the gray-level variations versus the position of the scaling lines of X_a or X_b of the facial image. In particular, we observe the two darkest candidates along these two lines. Whenever the gray-level value falls to the local minimum, it may correspond to the position where the eyebrow or eyeball possibly locates. Since the eyebrow corresponds to a lower gray value, we can know that the valleys A_1 and B_1 are the eyebrows in Figs. 2.9(a) and 2.9(b), respectively. It is also known that the eyeballs would have the darkest in the gray value. In consequence, we can expect that

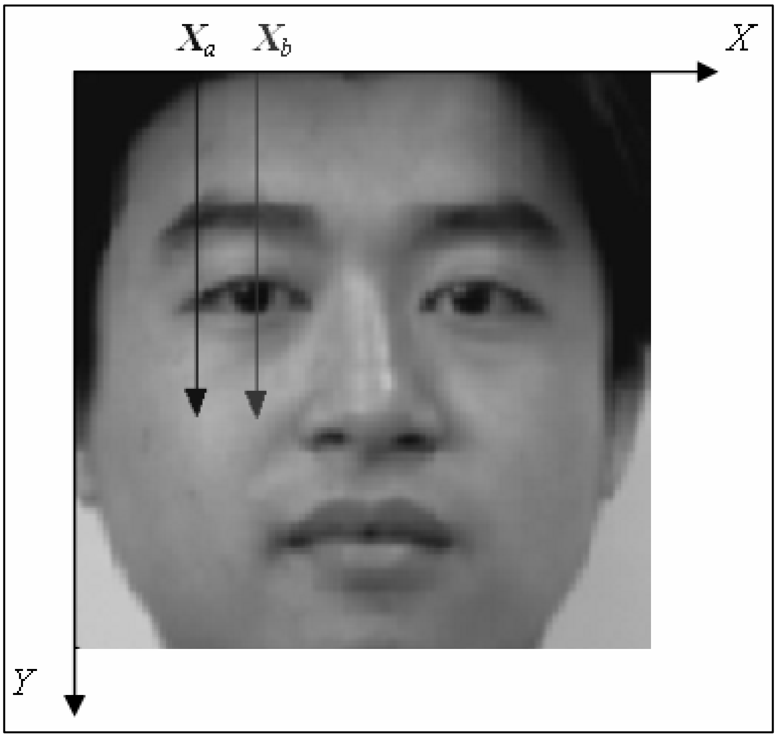
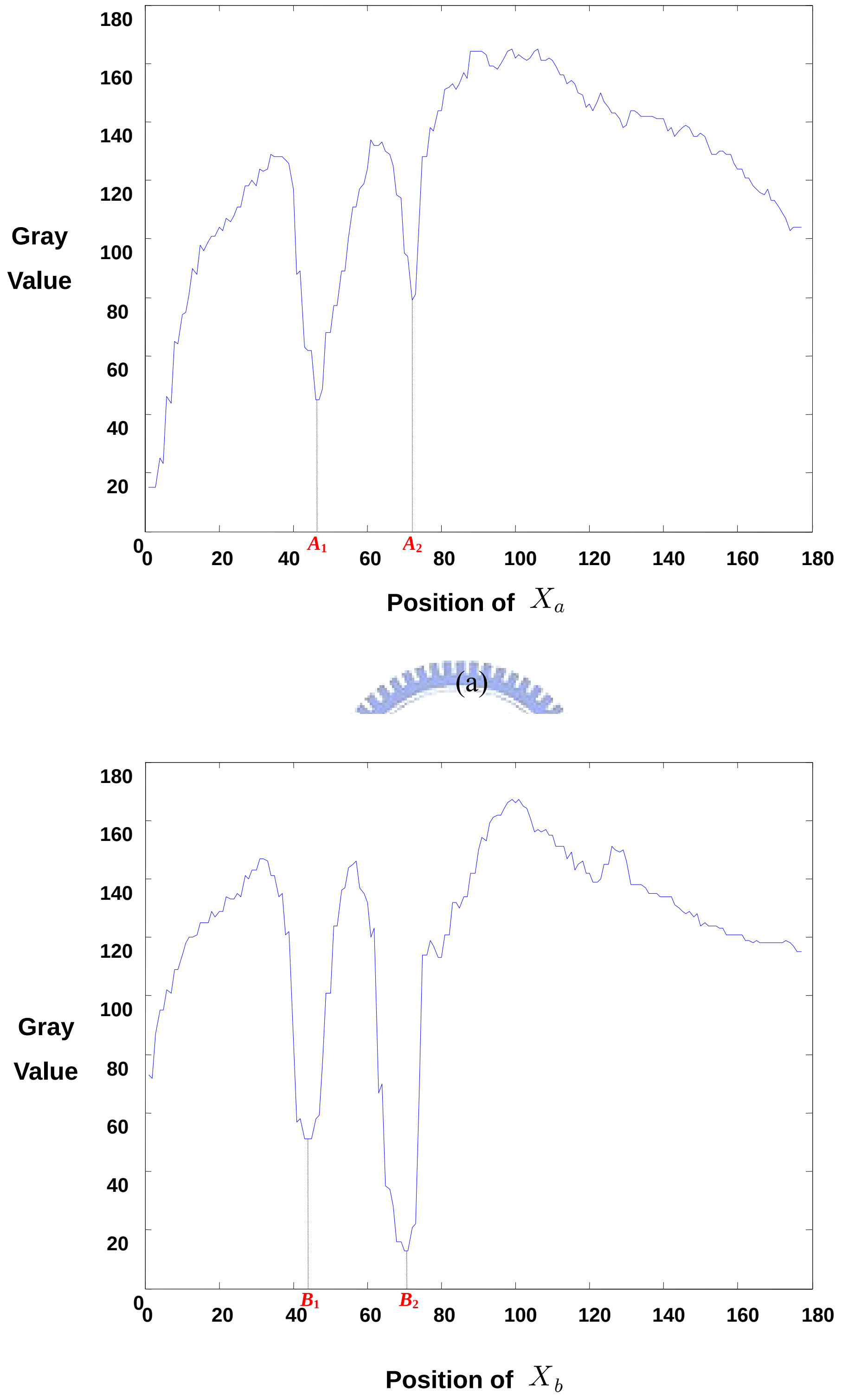


Fig. 2.8. The image of a subject's face: line X_a does not cross the eyeball and line X_b crosses the eyeball.



(a)

(b)

Fig. 2.9. Gray-level value variations along lines (a) X_a and (b) X_b.

2.4 Eye Detection with Sunglasses

The presence of sunglasses will affect the performance of extracting eye feature because the sunglasses may overlap with the eyes such that we could not separate the eyes from the detected eye regions easily. In this thesis, we apply image processing techniques to eliminate the effect of caused by sunglasses in eye detection.

2.4.1 Sunglasses Segmentation

We detect the sunglasses area from the input image. We select the seeded region growing (SRG) [13] technique for sunglasses segmentation. It has been demonstrated that, for several sets of images, region growing processes best perform than clustering or thresholding approaches because they deal with spatial repartition of color information. Seeded region growing algorithms typically start with seed pixels, then iteratively add to regions unassigned neighboring pixels which satisfy one or several homogeneity criteria. Thus, we can define a region as being a set of connected pixels which satisfy some homogeneity criteria. The SRG algorithm is described as follows.

Seeded region growing algorithm is a new approach which is based on conventional postulate of region growing algorithms where the criteria of similarity of pixels is applied, but the mechanism of growing regions is closer to the watershed algorithm. Instead of controlling region growing by tuning homogeneity parameters, SRG is controlled by choosing a usually small number of pixels, known as seeds. So, we start with a number of seeds which have been grouped into n sets, say, $A_1,\ A_{2,}\ \ldots,\ A_n.$ Sometimes, individual sets will consist of single points. It is in the choice of seeds that the decision of what is a feature of interest and what is irrelevant or noise is embedded. Given the seeds, SRG then finds a tessellation of the image into regions with the property that each connected component of a region meets (nonempty intersection with) exactly one of the A_i and, subject to this constraint, the regions are chosen to be as homogeneous as possible.

At each step of the algorithm we add one pixel to some of the sets $A_1,\ A_{2,}\ \ldots,\ A_n.$ Each step of the algorithm involves the addition of one pixel to one of the above sets. Consider now the state of sets A_i after *m* steps. Let *T* be the set of all as-yet unallocated pixels which border at least one of the regions

$$T = \left\{ x \notin \bigcup_{i=1}^{n} A_i \,\middle|\, N(x) \cap \bigcup_{i=1}^{n} A_i \neq \phi \right\} \tag{2.4}$$

where $N(x)$ is the set of immediate neighbors of the pixel x. In the examples to be presented in this correspondence, we will use a rectangular grid with immediate neighbors being those which are 8-connected to the pixel x. If, for $x \in T$ we have that $N(x)$ meets just one of the A_i, then we define $i(x) \in \{1, 2, \ldots, n\}$ to be that index such that $N(x) \cap A_{i(x)} \neq \phi$ and define $\delta(x)$ to be a measure of how different x is from the region it adjoins. The simplest definition for $\delta(x)$ is

$$\delta(x) = \left| g(x) - mean_{y \in A_{i(x)}} [g(y)] \right| \tag{2.5}$$

where $g(x)$ is the gray value of the image point x. If $N(x)$ meets two or more of the A_i, we take $i(x)$ to be a value of i such that $N(x)$ meets A_i and $\delta(x)$ is minimized. We then take a $z \in T$ such that

$$\delta(z) = {}^{min}_{x \in T} \{\delta(x)\} \tag{2.6}$$

and append z to $A_i(z)$.

This completes step m+1. The process is repeated until all pixels have been allocated. The process commences with each A_i being just one of the seed sets. The definition (2.5) and (2.6) ensure that the final segmentation is into region as homogeneous as possible given the connectivity constraint.

In programming SRG, we make use of a data structure which we will term a Sequentially Sorted List (SSL). In that list only coordinates of the pixels and distance δ from their neighboring regions are stored. When we consider a new pixel we take it from the SSL and process it. Because of the fact that all pixels are neighbors of already created regions and that they are put into SSL and processed by taking first of them from the list we have an impression that all border pixels of all regions are processed in parallel. When we add pixel to the list we have to add it according to its distance from the neighboring region.

The algorithm for implementing SRG (boundary flagging case) is as follows:

Initialization:

Label seed points according to their initial grouping.

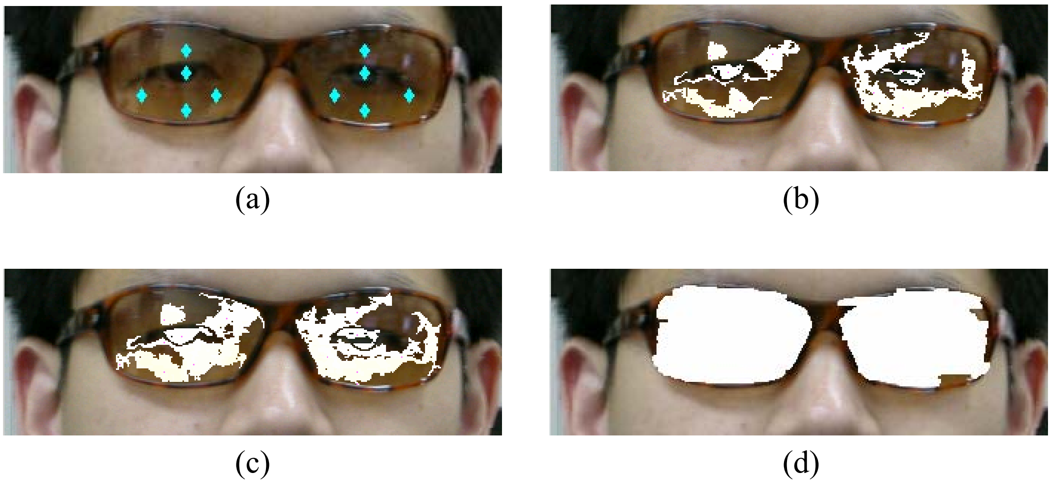
Put neighbors of seed points (the initial T) in the SSL.

```
Region Growing:
While SSL is not empty do
    Remove first pixel $y$ from the SSL.
```

```
Test the neighbors of this point:
    if all neighbors of $y$ which are already labeled (other than boundary label) have the same label –
        Set $y$ to this label.
        Update running mean of corresponding region.
        Add neighbors of $y$ which are neither already set nor already in the SSL to the SSL according to their value of $\delta$
    else
        Flag $y$ with the boundary label.
```

Note that previous entries in the SSL are not updated to reflect their differences from the new region mean. This leads to negligible difference in the results, but greatly enhanced speed.

This stepwise description shows that, in executing the algorithm, each pixels is visited just once, although at each visit we also view each of the neighbors. Hence, it makes for a very rapid program. Fig. 2.10 gives a visual demonstration of the region growing mechanism.



(a) (b)

(c) (d)

Fig. 2.10. Segmentation results of sunglasses. (a) Original image with 10 seeds marked (cyan diamonds). (b) The sunglasses region after 30% of pixels have been allocated by SRG. (c) The sunglasses region after 50% of pixels have been allocated. (d) The final sunglasses region.

2.4.2 Retinex Image Enhancement Technique

After segmenting the sunglasses regions by SRG, we try to enhance the sunglasses region such that we could extract the eye region. Many image enhancement methods have been proposed. Typical methods in such ones are an auto gain/offset correction method, a histogram equalization method, a homomorphic filtering method, and retinex image enhancement method. These methods have their different characteristics. Of these methods the retinex one is the most popular and most widely used because of its simplicity in use and its powerful ability to enhance images.

Land [14] first proposed the idea of retinex as a model of lightness and color perception of the human vision. Obviously it is not only a model, but also could be developed to algorithms of image enhancement. Land gave more contributions on retinex algorithm, evolving the concept from a random walk computation [15], to his latest version of a center/surround spatially opponent operation [16]. The center/surround opponent operation is related to the neurophysiological functions of neurons in the primate retina, lateral geniculate nucleus, and cerebral cortex. Hurlbert [17] studied the lightness theories and found that they have a common mathematical foundation. Also the leaning problems for artificial neural networks suggested a solution with center/surround form. But that is not enough. The human vision does not determine the relative reflectances, but rather content dependent relative reflectances for arbitrary illumination conditions.

Jobson and his coworkers [18] defined a Single-Scale Retinex (SSR), which is an implementation of center/surround retinex. But depending on the special scale, it can either provide dynamic range compression (small scale) or tonal rendition (large scale). Superposition of weighted different scale SSR is obvious a choice to balance these two effects. This is Multi-Scale Retinex (MSR) [19]. For color images, if the content is out of "gray world," which means the spatial averages of three color bands are far from equal, the output will be forced to be gray by MSR. This problem could be solved by introducing weight factor for different channels in Multi-Scale Retinex With Color Restoration (MSRCR) [20]. After MSRCR, generally the outputs will be out of the range of display. Auto gain/offset can be used to shift and compressed the histogram of MSRCR outputs to the display domain. But the histograms of MSRCR outputs show typical shapes and the gain/offset parameters could be "canonical".

In this thesis, we will implement MSRCR with gain/offset. We adjust the gain/offset parameters to adjust most of the pixels values to display domain and clap small part of the values to improve the contrast. The following content was excerpted from “Jobson *et al*. [20]”.

The basic form of MSR is given by

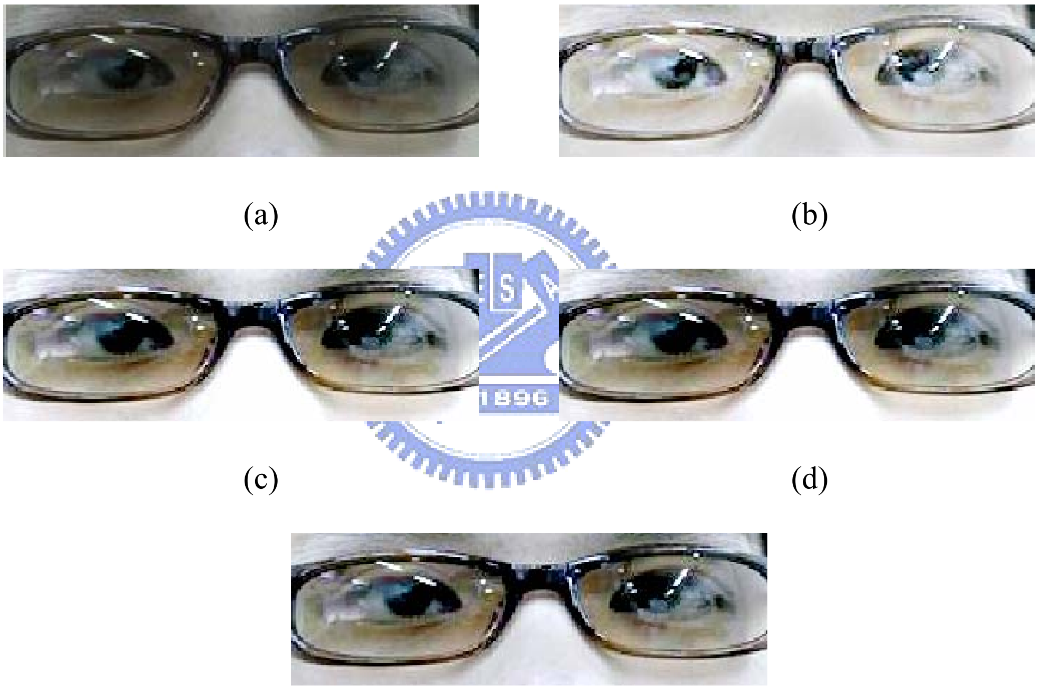
$$R_i(x_1,x_2)=\sum_{k=1}^{K}W_k\left(\log I_i(x_1,x_2)-\log\left[F_k(x_1,x_2)*I_i(x_1,x_2)\right]\right)\quad i=1,\ldots,N \tag{2.7}$$

where the sub-index i represents the *i*-th spectral band, *N* is the number of spectral bands—*N* = 1 for grayscale images and *N* = 3 for typical color images. In the latter case, $i \in R, G, B$—*I* is the input image, *R* is the output of the MSR process, F_k represents the *k*-th surround function, W_k are the weights associated with F_k, *K* is the number of surround functions, or scales, and $*$ represents the convolution operator. The surround functions, F_k are given as:

$$F_k(x_1,x_2)=k\exp\left[-(x_1^2+x_2^2)/\sigma_k^2\right] \tag{2.8}$$

where σ_k are the scales that control the extent of the surround – smaller values of σ_k lead to narrower surrounds and $k = 1/\left(\sum_{x_1}\sum_{x_2}F(x_1,x_2)\right)$.

As mentioned above, we found that multiple surrounds were necessary in order to achieve a graceful balance between dynamic range compression and tonal rendition. The number of scales used for the MSR is, of course, application dependent. We have found empirically, however, that a combination of three scales representing narrow, medium, and wide surrounds is sufficient to provide both dynamic range compression and tonal rendition. Fig. 2.11 shows the input image, the output of the MSR and the outputs when the different surround functions are applied to the original image. As is evident from Fig. 2.11, none of the individual scales attains the goal that we are trying to achieve: visual realism. The narrow and medium surround cases are self-explanatory. The wide-surround case deserves some discussion because it is a “nice” output image. However, the lack of dynamic range obscures the features that were visible to the observer, hence it fails the test. The MSR processed image uses features from all three scales to provide simultaneous dynamic range and tonal rendition.



(a)

(b)

(c)

(d)

(e)

Fig. 2.11. (a) The original input. (b) Narrow surround (15 pixels). (c) Medium surround (80 pixels). (d) Wide surround (250 pixels). (e) MSR output. The narrow-surround acts as a high-pass filter, capturing all the fine details in the image but at a severe loss of tonal information. The wide-surround captures all the fine tonal information but at the cost of dynamic range. The medium-surround captures both dynamic range and tonal information. The MSR is the average of the three renditions.

The general effect of retinex processing on images with regional or global gray-world violations is a “graying out” of the image either in specific regions or globally. This desaturation of color can, in some cases, be severe therefore we can consider the desired color computation as a color restoration, which should produce good color rendition for images with any degree of graying. In addition we would like for the correction to preserve a reasonable degree of color constancy since that is one of the basic motivations for the retinex. Color constancy is known to be imperfect in human visual perception, so some level of illuminant color dependency is acceptable provided it is much lower than the physical spectrophotometric variations. Ultimately this is a matter of image quality and color dependency is tolerable to the extent that the visual defect is not visually too strong.

We consider the foundations of colorimetry even though it is often considered to be in direct opposition to color constancy models and is felt to describe only the so-called “aperture mode” of color perception, i.e., restricted to the perception of color lights rather than color surfaces. The reason for this choice is simply that it serves as a foundation for creating a relative color space and in doing this uses ratios that are less dependent on illuminant spectral distributions than raw spectrophotometry. We compute a color restoration factor, α based on the following transform:

$$\alpha_i\left(x_1,x_2\right)=f\left(I_i\left(x_1,x_2\right)/\sum_{n=1}^{N}I_n\left(x_1,x_2\right)\right) \tag{2.9}$$

where $\alpha_i(x_1,x_2)$ is the color restoration coefficient in the *i*-th spectral band, *N* is the number of spectral bands, I_i is the *i*-th spectral band in the input image, and $f()$ is some mapping function. In a purely empirical fashion this was tested on several retinexed images to gain a sense of the visual impact. This proved to restore color rendition, encompassing both saturated and less saturated colors. Adding this to Equation (2.7), the MSRCR is given by:

$$R_i(x_1, x_2) = \alpha_i(x_1, x_2) \sum_{k=1}^{K} W_k \left(\log I_i(x_1, x_2) - \log\left[F_k(x_1, x_2) * I_i(x_1, x_2)\right]\right) \tag{2.10}$$

The results of applying this transformation to the "monochrome" images are shown in Fig. 2.12. Fig. 2.13 shows the sunglasses image enhanced using MSRCR.

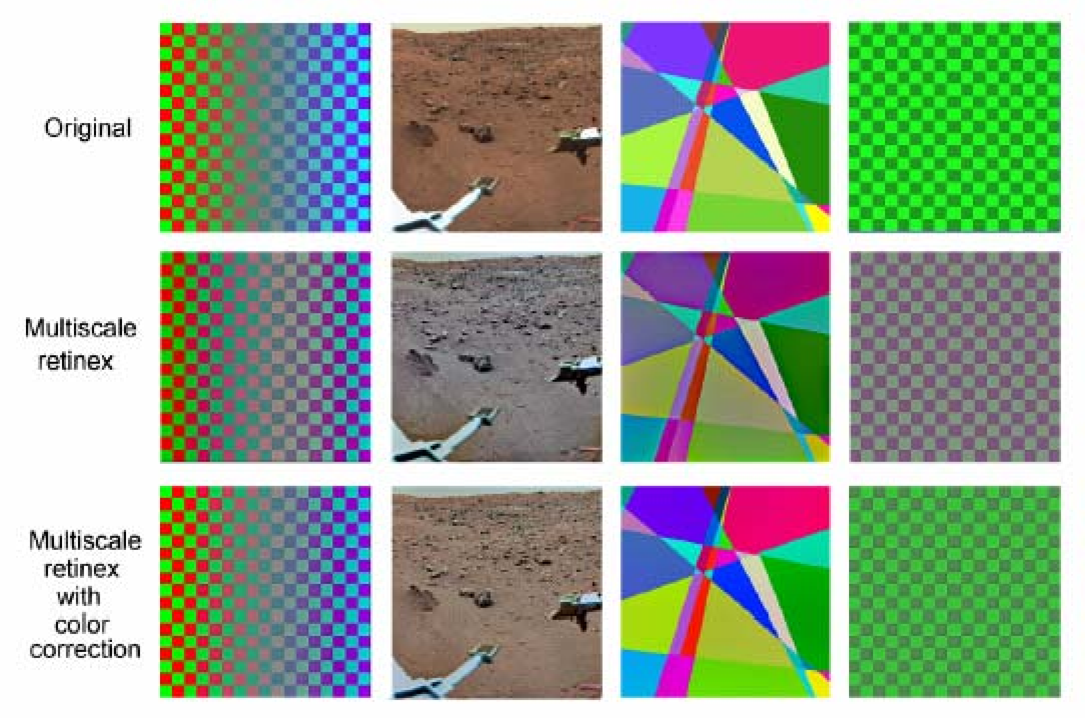
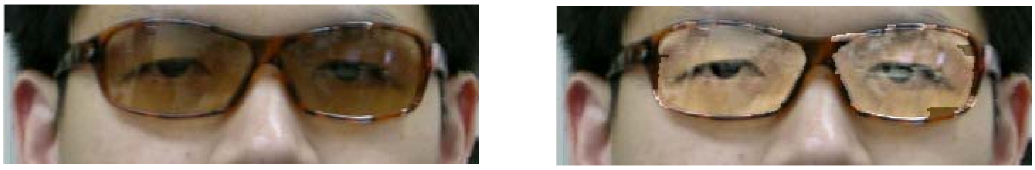


Fig. 2.12. (Top row) Scenes that violate the gray-world assumption; (Middle row) the MSR output; note the graying of large areas of monochromes; (Bottom row) The MSRCR output; note that color constancy is diluted in order to achieve correct tonal rendition.



(a) (b)

Fig. 2.13. The sunglasses image enhanced using MSRCR. (a) Original image. (b) The image after MSRCR.

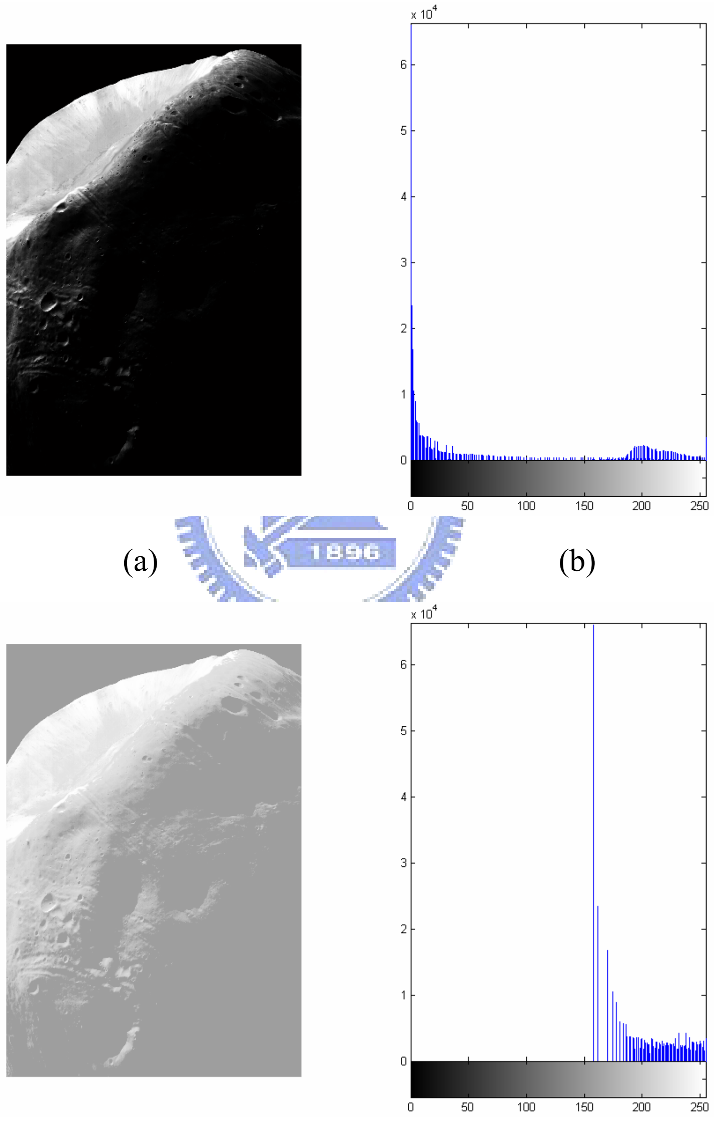
While we have called this additional color computation a “restoration” we have noticed in retrospect that depending upon the form of $f()$, this can be considered as a spectral analog to the spatial operation of the retinex itself. If we use

$$\alpha_i\left(x_1,x_2\right)=\log\left(I_i\left(x_1,x_2\right)/\sum_{n=1}^{N}I_n\left(x_1,x_2\right)\right) \tag{2.11}$$

then the internal form of the retinex process and the color restoration process is essentially the same. This mathematical and philosophical symmetry is intriguing since it suggests that there may be a unifying principle at work. Both computations are contextual in nature and highly relative and nonlinear. We can venture the speculation that the visual representation of wide ranging scenes must be a compressed mesh of contextual relationships even at the stage of lightness and color representation. This sort of information representation would certainly be expected at more abstract levels of visual processing such as form information composed of edges, links, and the like but is surprising for a representation so closely related to the raw image. Perhaps in some way this front-end computation can serve later stages in a presumed hierarchy of machine vision operations that would ultimately need to be capable of such elusive goals as resilient object recognition.

2.4.3 Histogram Equalization Enhancement Technique

In order to analyze the performance of the retinex image enhancement techniques, we compare the retinex result with histogram equalization technique. Histogram equalization technique is based on the idea of remapping the histogram of the scene to a histogram that has a near-uniform probability density function. This results in reassigning dark regions to brighter values and bright regions to darker values. As a result, the histogram equalization technique heavily depend on input images. Histogram equalization works well for scenes that have unimodal or weakly bi-modal histograms (i.e. very dark, or very bright), but not so well for those images with strongly bi-modal histograms (i.e. scenes that contain very dark and very bright regions). Fig. 2.14(a) show a bi-mode image, and Fig. 2.14(b) shows its histogram. The image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in the dark end of the gray scale. At first glance, one might conclude that histogram equalization would be a good approach to enhance this image, so that details in the dark areas become more visible. However, the result in Fig. 2.14(c), obtained using the command shows that histogram equalization in fact did not produce a particularly good result in this case. The reason for this can be seen by studying the histogram of the equalized image, show in Fig. 2.14(d). Here, we see that that the intensity levels have been shifted to the upper one-half of the gray scale, thus giving the image a washed-out appearance. The cause of the shift is the large concentration of dark components at or near 0 in the original histogram. In turn, the cumulative transformation function obtained from this histogram is steep, thus mapping the large concentration of pixels in the low end of the gray scale to the high end of the scale. Fig. 2.15 shows a sunglasses image that compares the output of the MSRCR with histogram equalization. As can be seen, the MSRCR provided the best overall visual quality. We have also tried local histogram equalization approach, but it usually cause too much block effect in the equalized image. By our experience, eye region enhancement by histogram equalization does not perform consistently, i.e., sometimes good and sometime bad. However the MSRCR approach performs constantly good.

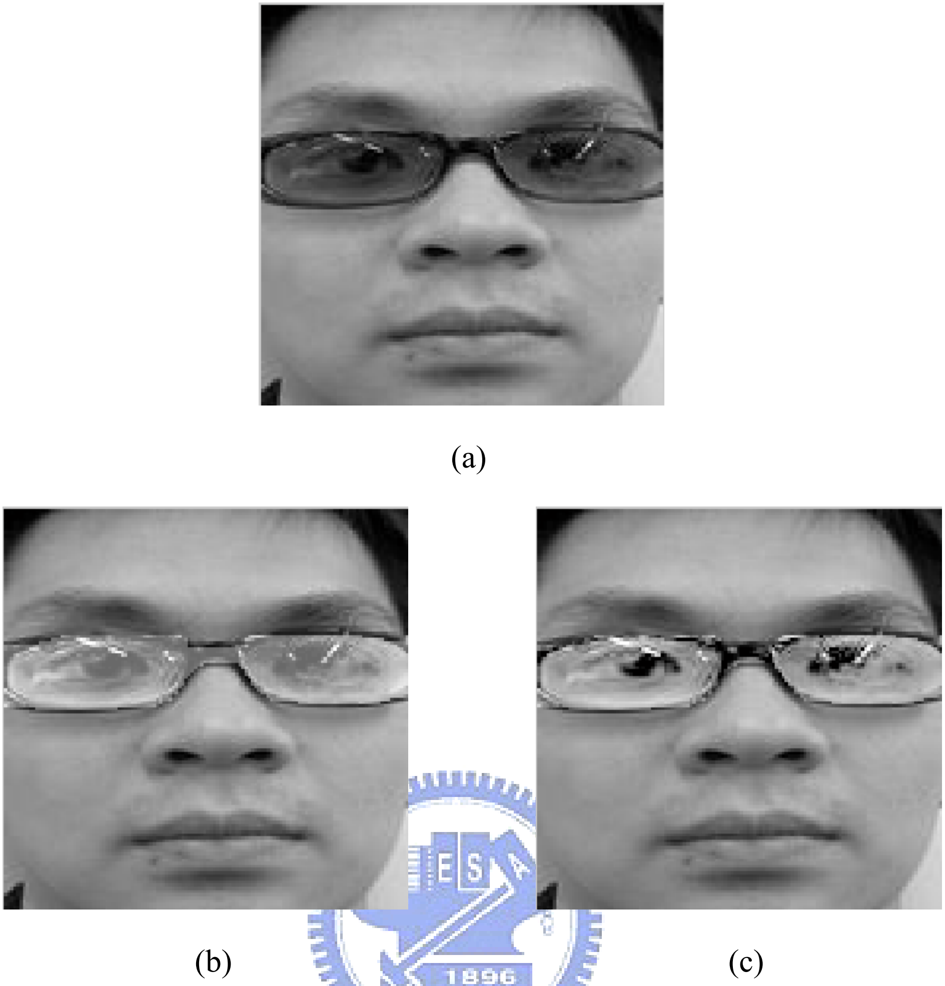


(a) (b)

(c) (d)

1896

Fig. 2.14. Illustration of histogram equalization. (a) Original image. (b) Histogram. (c) Histogram equalized image. (d) Histogram of (c).



(a)

(b)

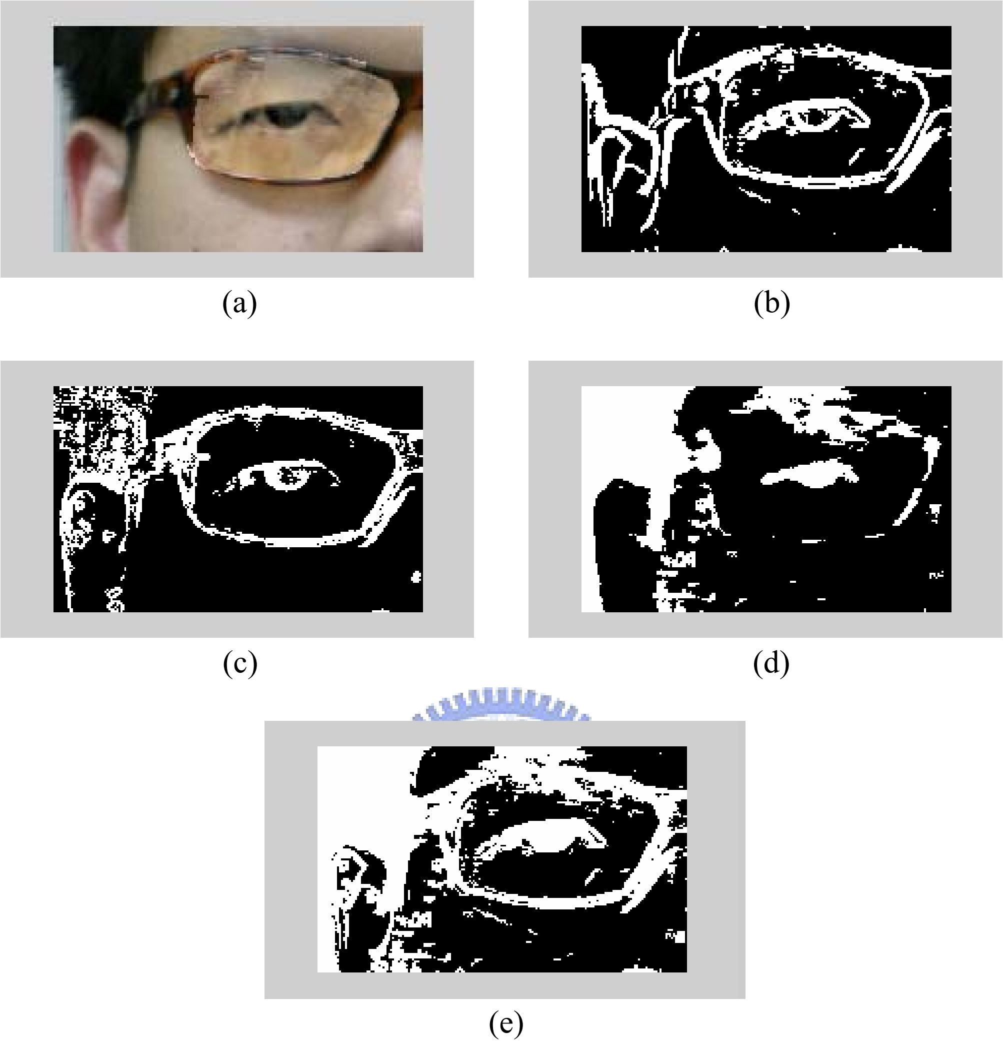
(c)

Fig. 2.15. A comparison of histogram equalization and the MSRCR. (a) Original image. (b) Histogram equalization. (c) MSRCR.

2.4.4 Eyeball Extraction

The remaining question is how to locate eyes without sunglasses. We begin with the region that contains the eyes and sunglasses. First, we compute the color edge of the eye region in HSV color model and do erosion and dilation to get the rough edge map of sunglasses. The color edge detection method proposed by Fan *et al.* [21] use entropic thresholding technique to obtain an optimal threshold that is adaptive to the image contents, and this technique has been proved to be highly efficient for two-class data classification problem [22]. Then, we convert the image of eye region from RGB color space to YC_rC_b color space. It is easy to see that in YC_rC_b color model domain, the intervals of the C_r and C_b components of skin-color are always very different from sunglasses and can easily be clustered to two classes. However, for kinds of sunglasses, such as metallic and thin-frame, the color of sunglasses frame sometimes lies in the skin-color interval in YC_rC_b color model because the metallic reflection and the noise caused by low resolution CCD. In order to deal with these problems, we add extra information of RGB gradient edge detector. Therefore, we combine the three evidences to guarantee that the sunglasses have completely been extracted, despite some noises caused by hair or eyebrows to be included in the map. Fig. 2.15 show an example of edge detection while one worn sunglasses.

After getting the edge map of detected eye region, we use geometry and projections to eliminate the sunglasses region and locate accurately the eyes position. When we apply erosion to the edge image of wearing sunglasses image, the edge will break into small pieces and then the eye can be separated from sunglasses contour easily by selecting the largest connected component which has the smallest standard deviation to each center of the component. The hair and eyebrows components also can be recognize because it is always from the top to the bottom and begin with the top of the eye region we have set. The extracted eye result from edge map of wearing sunglasses examples in Fig. 2.15 demonstrates in Fig. 2.16.



(a) (b)

(c) (d)

(e)

Fig. 2.16. An example of edge detection. (a) Original image. (b) Edge detection using gradient operator in RGB color space. (c) Edge detection using gradient operator in Hue component of HSV color space. (d) Non-skin-color region. (e) The resultant edge map union the previous three edge map.

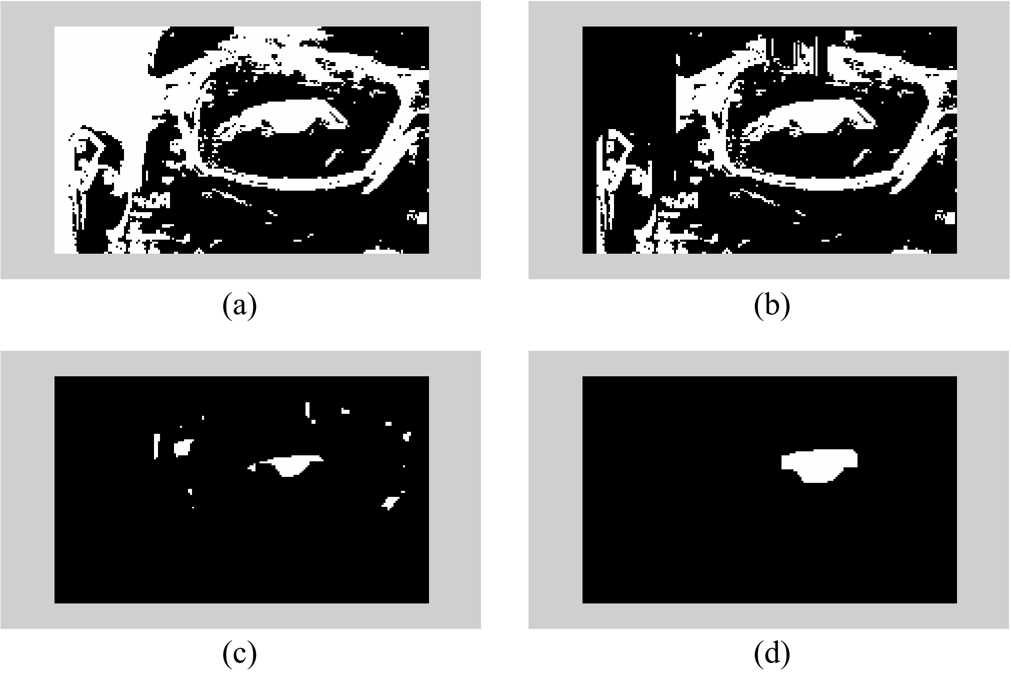


Fig. 2.17. Eye extraction from edge map of Fig. 2.16. (a) Edge map of eye region. (b) Edge map with eliminating the hair region. (c) Edge map which eliminating hair region with twice erosion. (d) Extraction the pupil position.

2.5 Detection of Open/Closed State of The Eyes

The degree of eye openness leading to define the opened/closed state of the eyes will be established for each concerned subject. Besides, the size of each subject's eyes also varies with the relative distance between the camera and the subject's face. Based on the eye detection routines stated in Section 2.3, we can circumscribe the eye by a rectangle as shown in Fig. 2.17. We then determine the degree of eye openness by their vertical pupil lengths. The openness of each subject's eye is the ratio value of eyeball expansion, d_2, normalized by the maximum, i.e., normal, vertical expansion, d_1, when his eye is fully open. As a result, the ratio is equivalent to the eye openness percentage, as shown in Fig. 2.17. A threshold value is established for considering whether the subject's eye is open or closed: if the ratio is smaller than 0.2, the threshold used in this study, of the normal expansion, the eye of the subject in this image frame is defined to be closed; otherwise, it is defined to be open.

To proceed, the point where the eye center locates is set as the reference coordinates for consecutive facial image frame processing. It is not necessary to detect the eye every image frame because the eye position in the subsequent frame will correspond to the near-by eye coordinates of the eye detected in the previous image frame. Hence, we will, in general, carry the coordinate to the next facial image frame for quick eye locating. It is to be noted that the varying degree of the eye openness will be always within the range [0, 1]. If the computed eye openness value is outside this value, it can be concluded that the detected eye position is incorrect. In such a case, we have to restart the step of eye detection procedure of Section 2.3 over the current facial image frame.

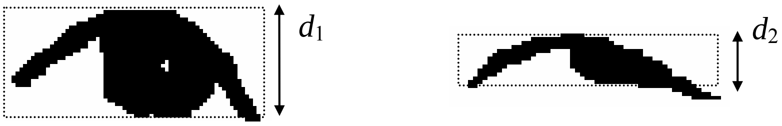


Fig. 2.18. The open/closed state of the eye openness. d_1 is the maximum vertical scale of the complete eye openness, and d_2 is another sampled openness of the same subject's eye.

Chapter 3

Drowsiness Detection

3.1 Introduction to Drowsiness Detection

Technology may be able to provide earlier or more robust warnings of degraded driving status so as to preclude a drowsy driving-related crash. Drowsy driver detection algorithms and approaches have been a topic of considerable research in recent years. A key ingredient in the development of such algorithms is to select an appropriate “criterion” measure for drowsiness. Such a measure of drowsiness should ideally be valid (i.e., relate to observed performance decrements in meaningful ways) and reliable (i.e., consistently vary with levels of sleepiness, circadian troughs, or time on task). Numerous physiological measures of driver arousal or wakefulness exist. These include electroencephalograms (EEGs), heart rate and heart rate variability, core body temperature, and various measures of the eyes

In this thesis, one measuring feature used for the drowsiness detection is PERCLOS [8], [9], [23]. The other measuring feature to be employed to determine the level of a driver’s alertness, is to calculate the Long Duration Blink Frequency of the driver’s eyes [24]. Information fusion is an important aspect of any intelligent system. Our proposed system is capable of integrating two different evidence sources (PERCLOS and Long Duration Blink Frequency) into a knowledgeable decision, to determine a driver is drowsy or not. The fuzzy integral combines objective evidence with the system’s expectation of the importance of that evidence.

3.2 Long Duration Blink Frequency Feature for Drowsiness Detection

Ogawa and Shimotani [24] studied the relationship between the conscious/drowsy state of a testee and his blinking behavior by testing ten subjects in a driving simulator. The blinking duration is quite short when a subject is in a conscious condition. On the contrary, the blink duration becomes longer in the drowsy condition [25], [26]. To evaluate the drowsiness detection performance in these tests, they measured LDBF of the eye closure over certain time duration. LDBF is defined as the number of long eye blinks, in which the eye is closed exceeding a duration longer than usual, within a certain time interval. The eye closure is defined as the eye is more than 80% closed with respect to its usual, which is identical to the definition stated in Section 2.5. To this end, we define a long duration blink if one's eye closure time lasts for more than 0.5 second inclusive in this research. It is to be remarked that the LDBF of the eye closure varies with the observation time interval and the processing frame rate of the captured video sequences.

1896

3.3 PERCLOS Feature for Drowsiness Detection

Eyelid closure has been recently proven to be a very reliable and meaningful indicator of drowsiness while driving. Wierwille *et al.* [9] found that it is indicative of a subject at the onset of drowsiness leading to poor response. If a driver's eyelids are closed during driving, his ability to operate a vehicle would be greatly hampered. The researchers demonstrated that PERCLOS, defined as the proportion of time that the eyes of the subject are closed over a specified period, can be used as a physiological indicator of drowsiness. Accordingly, a subject can be said to be drowsy if he has a high PERCLOS value. For example, suppose the eye of one subject is detected to be in the closed state, 80% closed as defined previously, for six seconds within one minute. The PERCLOS will be 6/60 = 10%. That is, the subject closes his eyes in 10% of one minute.

Figs. 3.1 and 3.2 show a typical night of observed drowsiness for these two drivers. Each plot spans the portion of the night's run where the bout was observed (time is indicated on the horizontal axis), and the peaks and troughs of the plot itself represent the proportion of time PERCLOS was 80% or greater within each minute.

E S A 1896

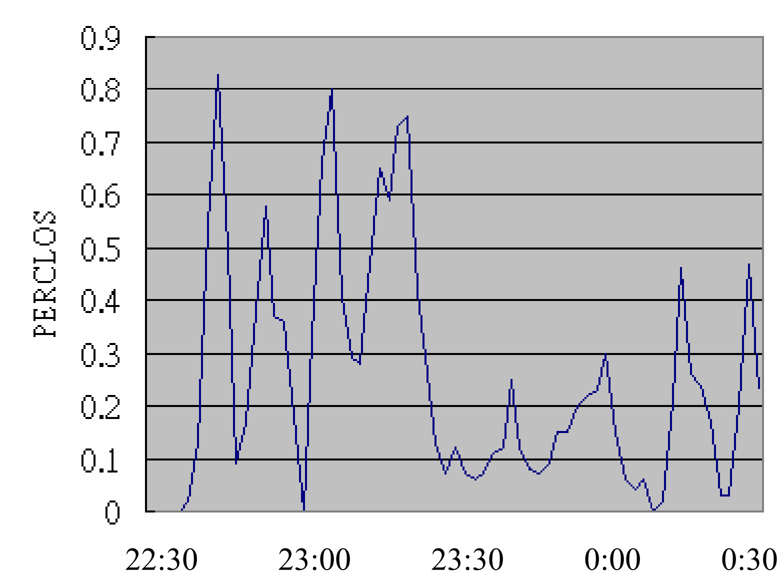


Fig. 3.1. One night of observed drowsiness for Driver A.

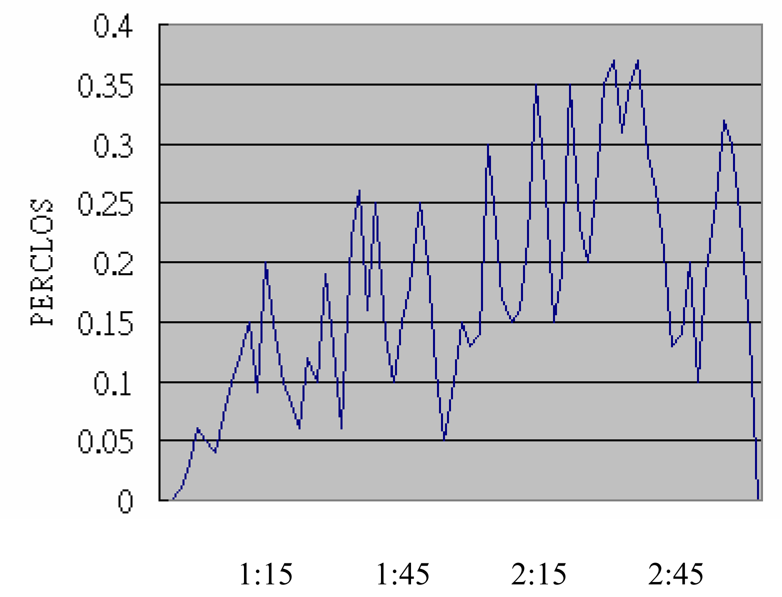


Fig. 3.2. One night of observed drowsiness for Driver B.

3.4 Information Fusion Using Fuzzy Integral

LDBF or PERCLOS feature alone face their own difficulties in reliably detecting drowsiness. First, for LDBF approach, subjects in severely drowsy state will have LDBF value near zero due to their seldom eye blinks. However, subjects in conscious state also have LDBF value near zero because their eye blink time is so short that almost all the blinks will not last exceeding the duration specified in Section 3.2. Consequently, low LDBF can easily cause false detection. Secondly, PERCLOS is also weak in some cases to detect drowsiness according to the Pearson correlation coefficients as reported in the Dinges and Grace's work [25]. Fuzzy integral based information fusion, on the other hand, can make use of the two feature values of PERCLOS and LDBF cooperatively to remedy the limitation from one dimensional thresholding. That is, a decision boundary is established such that either LDBF or PERCLOS can become more discriminative in classifying a subject's vigilance state depending on the relative importance of the features' evidences. As a result, LDBF and PERCLOS can support each other to increase the drowsiness detection accuracy. The fuzzy integral is defined as follows.

Definition: Let (X,Ω) be a measurable space and let: $X \rightarrow [0,1]$ be a Ω-measurable function. The fuzzy integral over $A \subseteq X$ of the function *h* with respect to a fuzzy measure *g* defined by

$$\int_A h(x) \circ g(\cdot) = \sup_{\alpha\in[0,1]} [min(\alpha, g(A \bigcap F_\alpha))],$$

where

$$F_\alpha = \{x : h(x) \geq \alpha\}.$$

The following is the interpretation of the fuzzy integral that will be adopted in this paper. Suppose that an object is evaluated from the point of view of a set of sources *X*. Let $h(x) \in [0,1]$ denote the decision for the object when source $x \in X$ is considered and let $g(\{x\})$ denote the degree of importance of this source. Now, suppose the object is evaluated using sources from $x \in X$. It is reasonable to consider a quantity

$$W(A) = \min_{x \in A} h(x),$$

is the best security decision that the object provides and *g*(*A*) expresses the grade of importance of this subset of sources. The value obtained from comparing these two quantities in terms of the min operator is interpreted as the grade of agreement between real possibilities, *h*(*x*), and the expectations, *g*. Hence, fuzzy integration is interpreted as searching for the maximal grade of agreement between the objective evidence and the expectation.

The following properties of fuzzy integral are easy to prove.

1) If $h(x) = c$, for all $x \in X, 0 \leq c \leq 1$, then

$$\int_X h(x) \circ g(\cdot) = c.$$

2) If $h_1(x) \leq h_2(x)$, for all $x \in X$, then

$$\int_X h_1(x) \circ g(\cdot) \leq \int_X h_2(x) \circ g(\cdot).$$

3) If $A \subset B$, then

$$\int_A h(x) \circ g(\cdot) \leq \int_B h(x) \circ g(\cdot).$$

4) Let $\{A_i : i = 1, \cdots, n\}$ be a partition of the set *X*. Then

$$\int_X h(x) \circ g(\cdot) \geq max(e_1, \cdots, e_n),$$

where e_i is the fuzzy integral of *h* with respect to *g* over *A*$_i$. The interpretation of all these properties related to the fuzzy integral as an information fusion technique should be obvious.

The calculation of the fuzzy integral when *X* is a finite set is easily given. Let $X = \{x_1, x_2, \cdots, x_n\}$ be a finite set and let *h*: *X*→[0,1] be a function. Suppose $h(x_1) \geq h(x_2) \geq \cdots \geq h(x_n)$, (if not, *X* is rearranged so that this relation holds). Then a fuzzy integral, *e*, with respect to a fuzzy measure *g* over *X* can be computed by

$$e = \max_{i=1}^{n}[\min(h(x_i), g(A_i))], \tag{3.1}$$

where $A_i = \{x_1, \cdots, x_i\}$.

Note that when *g* is a $g_\lambda -$fuzzy measure, the values of $g(A_i)$ can be determined recursively as

$$g(A_1) = g(\{x_1\}) = g^1. \tag{3.2}$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), \qquad \text{for} \quad 1 < i \leq n. \tag{3.3}$$

Thus the calculation of the fuzzy integral with respect to a g_λ −fuzzy measure would only require the knowledge of the density function, where *i*-th density, g^i, is interpreted as the degree of importance of the source x_i for *i*=1,2,⋯,*n*. Furthermore, the degree of importance may be interpreted as a belief if

$$\sum_{i=1}^{n} g^i < 1 \ ,$$

and a plausibility value if this sum is greater than 1.

Chapter 4

Simulation and Results

The eye-detection algorithm was tested on a number of people in order to confirm the validity. First, we capture frontal face images of people by a CCD camera, and then we will validate the algorithm via these images. In Section 4.1, the algorithm will be tested of people wearing sunglasses. We will give the step-by-step result of finding the eyes. In Section 4.2, we will first evaluated the drowsiness detection rate by LDBF and PERCLOS, respectively. Then we tested our proposed algorithm as a drowsiness detection system.

4.1 Experiment Results of Eye Detection with Sunglasses

We take frontal face images of four students in laboratory to test the eye extraction algorithm. The size of images is 640×480 and the simulation is processing on a Pentium IV 2.4GHz personal computer. Figs. 4.1– 4.4 show these four examples. In each example, we will show the original face image in sub-image (a), and face extraction result in (b). The input and output images of Seeded Region Growing (SRG) are given in sub-image (c) and (d) respectively. Sub-image (e) and (f) are the enhanced images using histogram equalization and MSRCR respectively, while (g) and (h) show the eye extraction result. The accuracy of the result for finding eye location is 100%.

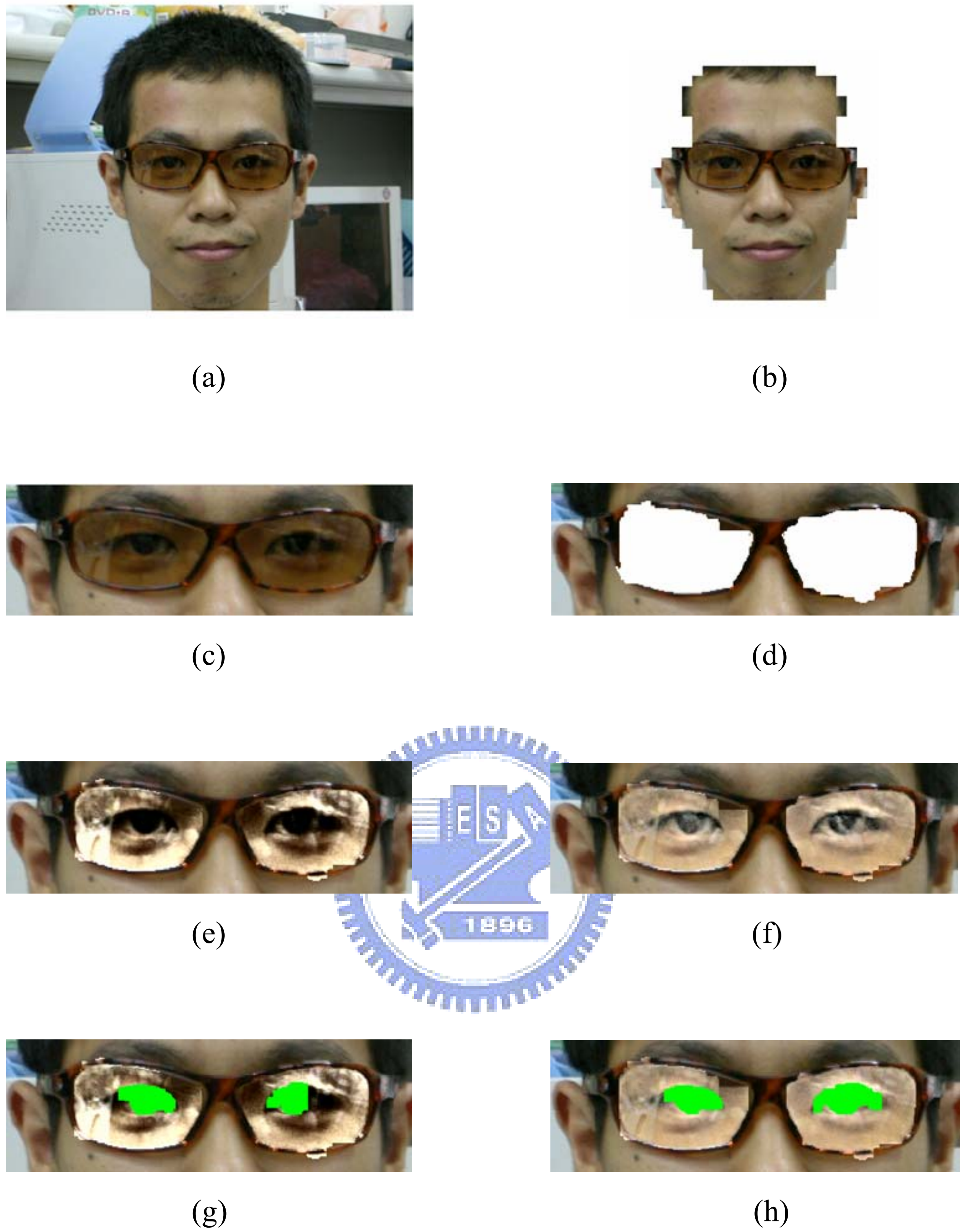
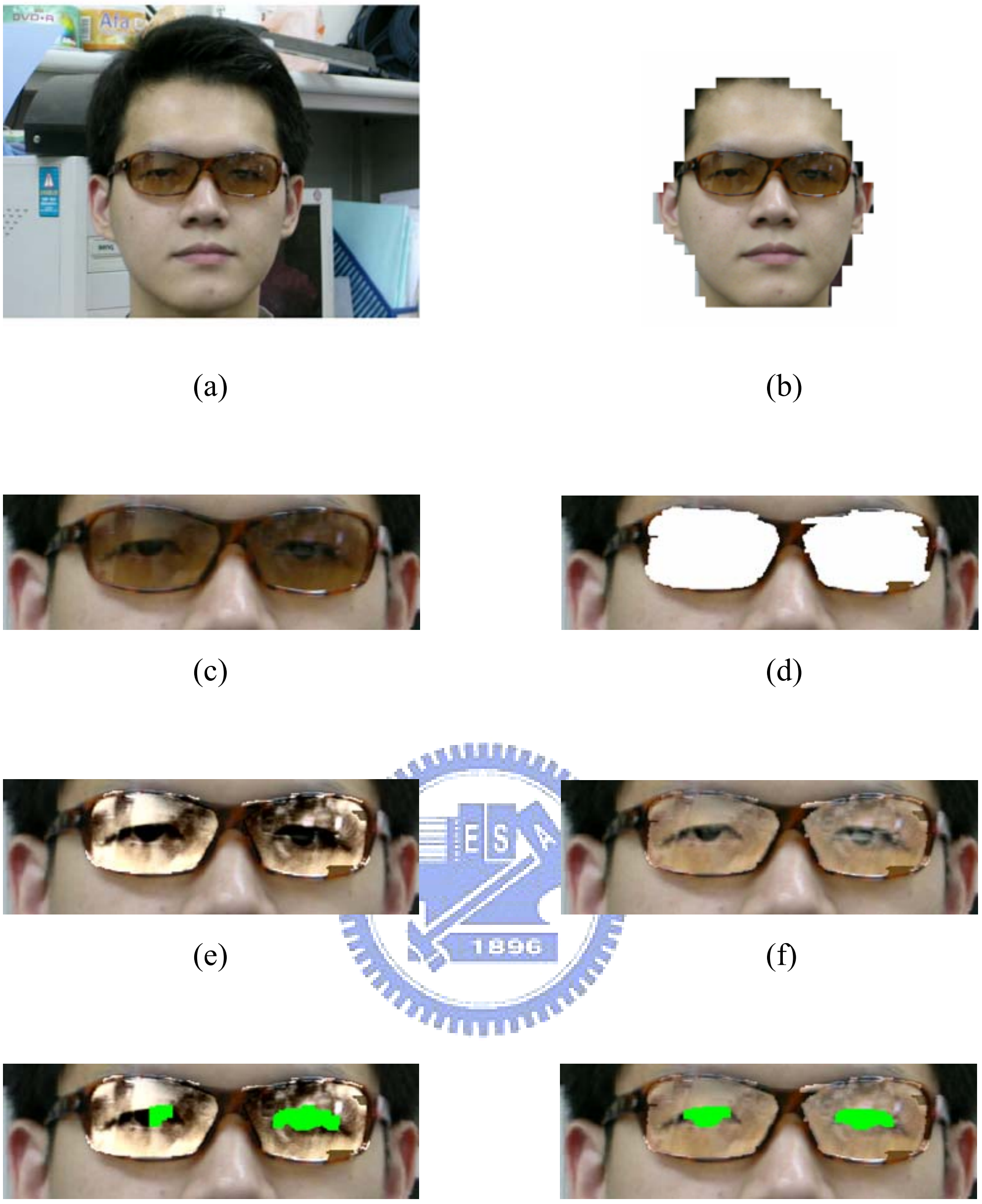

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4.1. Images of Example 1 for face detection and eye location. (a) The input image. (b) Face extraction result. (c) The input image for SRG. (d) The output image of SRG. (e) The sunglasses image enhanced using histogram equalization. (f) The sunglasses image enhanced using MSRCR. (g) The eye extracted from (e). (h) The eye extracted from (f).



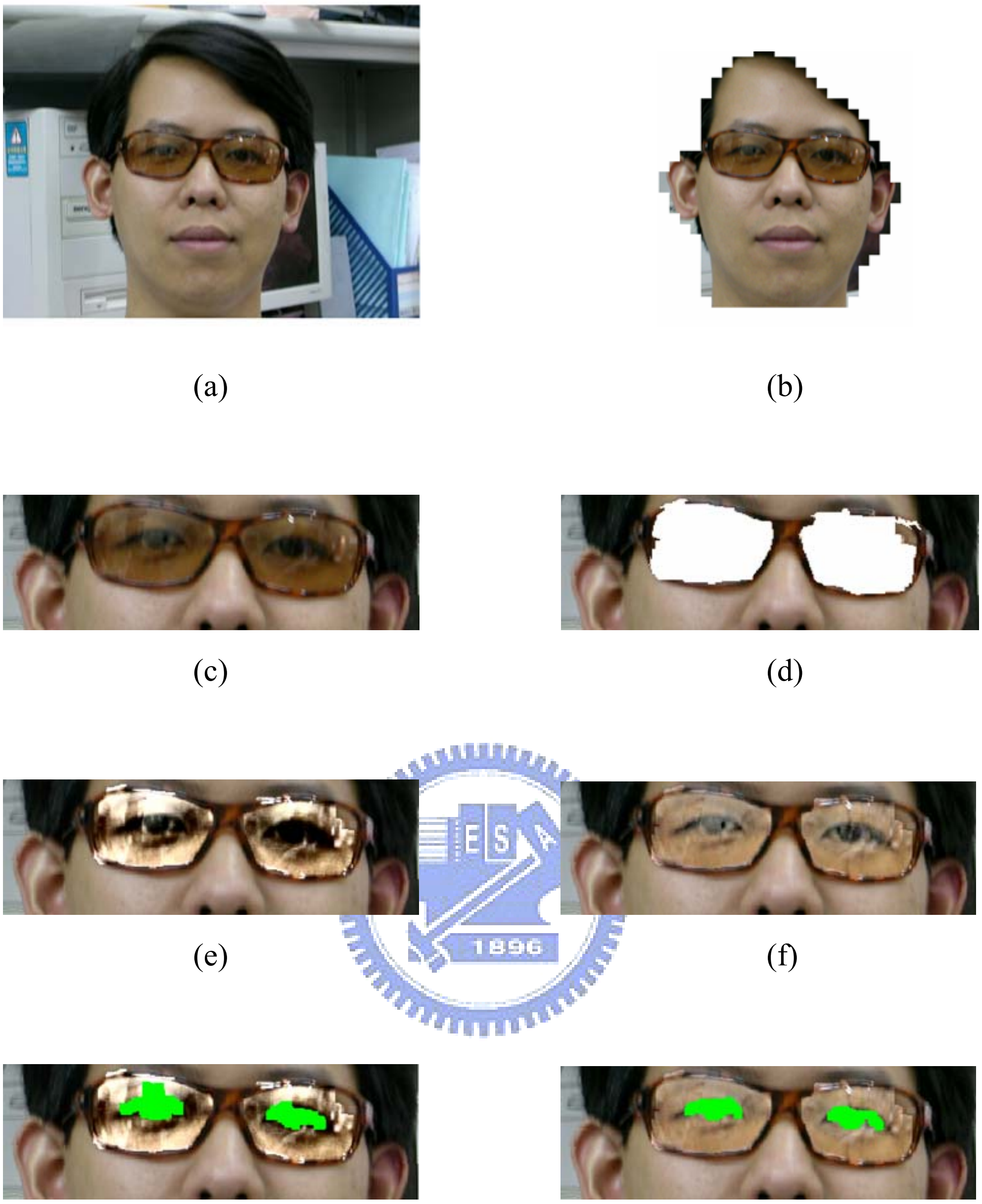
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 4.2. Images of Example 2 for face detection and eye location. (a) The input image. (b) Face extraction result. (c) The input image for SRG. (d) The output image of SRG. (e) The sunglasses image enhanced using histogram equalization. (f) The sunglasses image enhanced using MSRCR. (g) The eye extracted from (e). (h) The eye extracted from (f).



(a) (b)

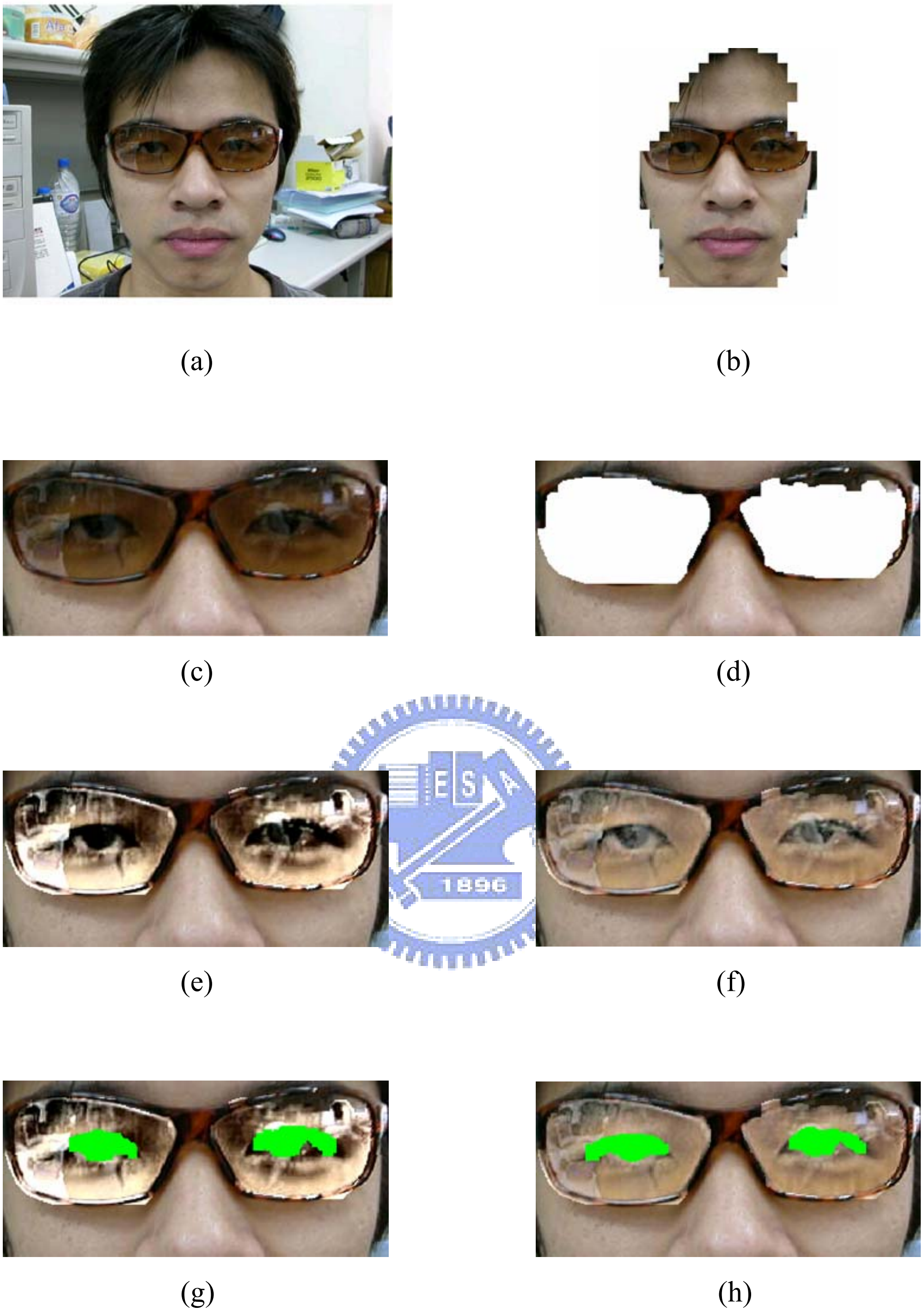
(c) (d)

(e) (f)

(g) (h)

1896

Fig. 4.3. Images of Example 3 for face detection and eye location. (a) The input image. (b) Face extraction result. (c) The input image for SRG. (d) The output image of SRG. (e) The sunglasses image enhanced using histogram equalization. (f) The sunglasses image enhanced using MSRCR. (g) The eye extracted from (e). (h) The eye extracted from (f).



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 4.4. Images of Example 4 for face detection and eye location. (a) The input image. (b) Face extraction result. (c) The input image for SRG. (d) The output image of SRG. (e) The sunglasses image enhanced using histogram equalization. (f) The sunglasses image enhanced using MSRCR. (g) The eye extracted from (e). (h) The eye extracted from (f).

4.2 Experiment Results of Drowsiness Detection

For safety and imitating the drowsiness condition in a sense of reality, we selected 18 testees as subjects and did our experiments in a driving simulator which consisted of a car body and a projector. The car body was the control unit included a wheel-house with a driver seat, a steering wheel, a throttle, and brakes. The projector would project road scene images of driving. As shown in Fig. 4.5, a driver was steering the simulator having night-time highway scenery projected on the screen and controlled the steering wheel in the vehicle when the car diverging from the lane in the scene. We mounted on the dashboard a CCD camera with infrared lamps, which acted as auxiliary light sources to compensate for insufficient illumination, to capture the subjects' face image sequences during the driving period of two hours: all subjects were first under consciousness and finally became drowsiness. The resolution of frames and the video sampling rate were set to be 320×240 pixels and 15 frames per second, respectively. Some captured image sequences of the subjects were shown in Fig. 4.6. Image frames were firstly preprocessed using the steps described in Chapter 2. For illustration, two typical results after the eye open/closed state determination procedure of Chapter 2 under conscious state are shown in Table I. Similarly, two processed sequences under drowsy state of a subject are shown in Table II. Note that the eye state detection accuracy of the image frames processed by the method stated in Section 2.3 is also checked by manual inspection and demonstrates a high consistence rate of 97.1%.

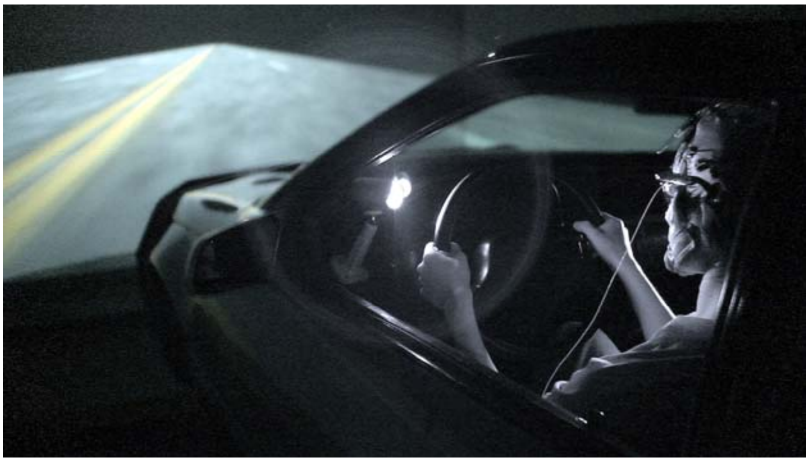


Fig. 4.5. A subject was steering the driving simulator through night-time highway scenery projected on the screen.

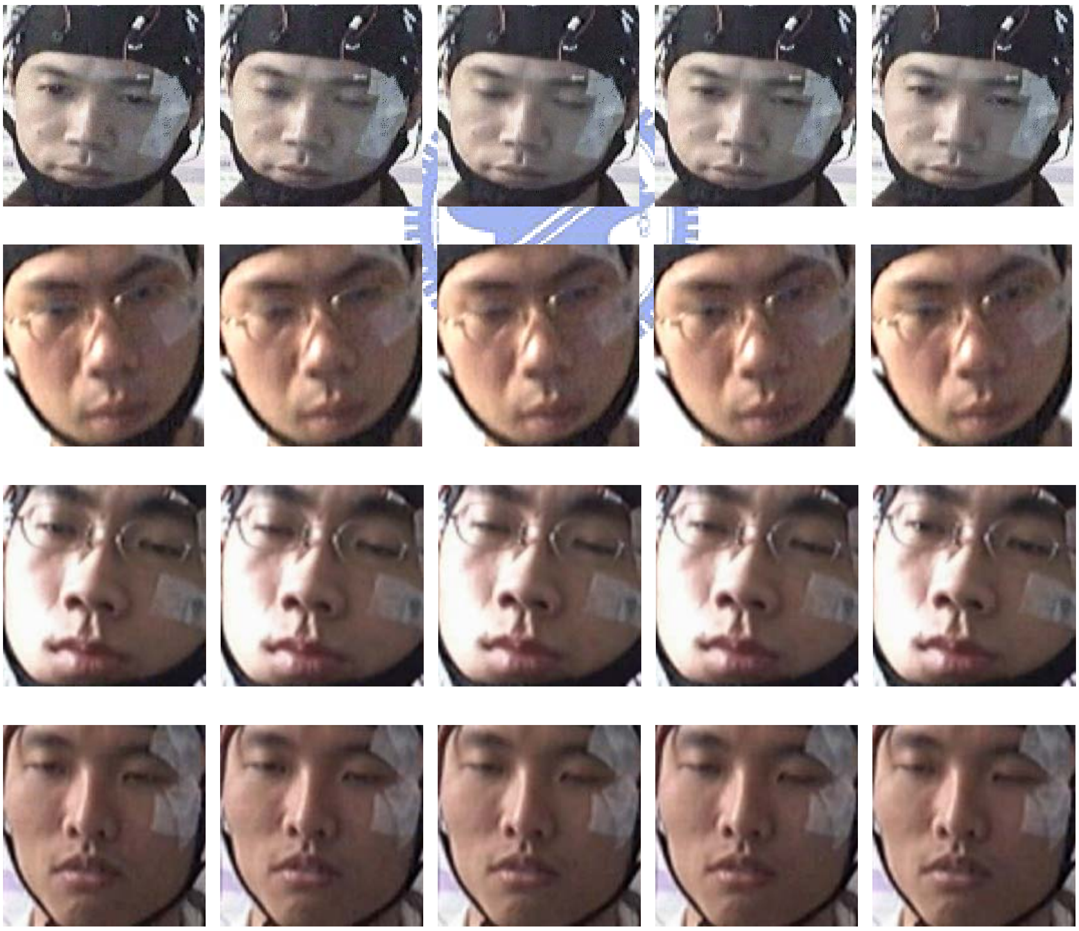


Fig. 4.6. Some captured image sequences of the 18 subjects in this experiment.

Table I. Two eye's open/closed state sequences of the testee under consciousness

Frame	227	228	229	230	231	232	233	234
Eye	open	open	open	open	close	close	open	open
frame	265	266	267	268	269	270	271	272
eye	open	open	close	close	open	open	open	open

Table II. Two eye's open/closed state sequence of the testee under drowsiness

frame	205	206	207	208	209	210	211	212
eye	open	close	close	close	close	close	open	open
frame	310	311	312	313	314	315	316	317
eye	open	close	close	close	close	close	close	close

4.3 The Best OTI Determination for LDBF and PERCLOS

Determining a suitable Observing Time Interval (OTI) to compute LDBF and PERCLOS for drowsiness evaluation is very important because, in practical application such as a driver drowsiness detection system, we need a reliable and fast algorithm which can in time alert people to prevent them from possible dangers caused by the loss of vigilance. To this end, we first analyzed respectively LDBF and PERCLOS distributions of drowsiness and consciousness of testees to select the best OTI leading to the highest discrimination from the distribution curves. Moreover, we also adopted a sliding window data update formation to speed up the drowsiness detection output rate.

4.3.1 The Best OTI Determination Procedure

In the following, we first describe the procedure of selecting the best OTI for LDBF feature. For each OTI changing from one second to ten seconds, we counted respectively LDBF under drowsiness and consciousness states of the image frames of testees and plotted the distributions of these two states. We had found that the best separability between drowsiness and consciousness states of the testees occurred when OTI was between 8 to 9 seconds. Continuing this strategy, the search procedure above was applied between 8 and 9 seconds and repeated up to a quarter of second accuracy. It leaded to the OTI of 8.25 seconds to yield the best drowsiness/consciousness separation by the of LDBF feature. To show the effectiveness of the OTI selection, we have plotted in Fig. 4.7 the distribution of LDBF for the best OTI equal to 8.25 seconds, in comparison to Fig. 4.8 of other random selection of OTI equal to 2 seconds. From these two figures, the superiority of

OTI equal to 8.25 seconds is obvious. Following similar procedure above, we could also find the best OTI of PERCLOS feature for drowsiness detection. We have found that the good OTI range for PERCLOS was wider and fell between 6 to 9 seconds. To be consistent with LDBF feature, we have chosen the OTI for PERCLOS equal to 8.25 seconds, too. The probability of PERCLOS feature under drowsiness and consciousness states of the testees for OTI equal to 8.25 and 2 seconds are plotted in Figs. 4.9 and 4.10, respectively. Likewise, the superiority of Fig. 4.9, in discriminating drowsiness from consciousness is evident. For this best OTI, it follows from Figs. 4.7 and 4.9 that the thresholds of LDBF and PERCLOS can be chosen to be 0.8 and 0.202, respectively, as shown in the first two columns of Table III.

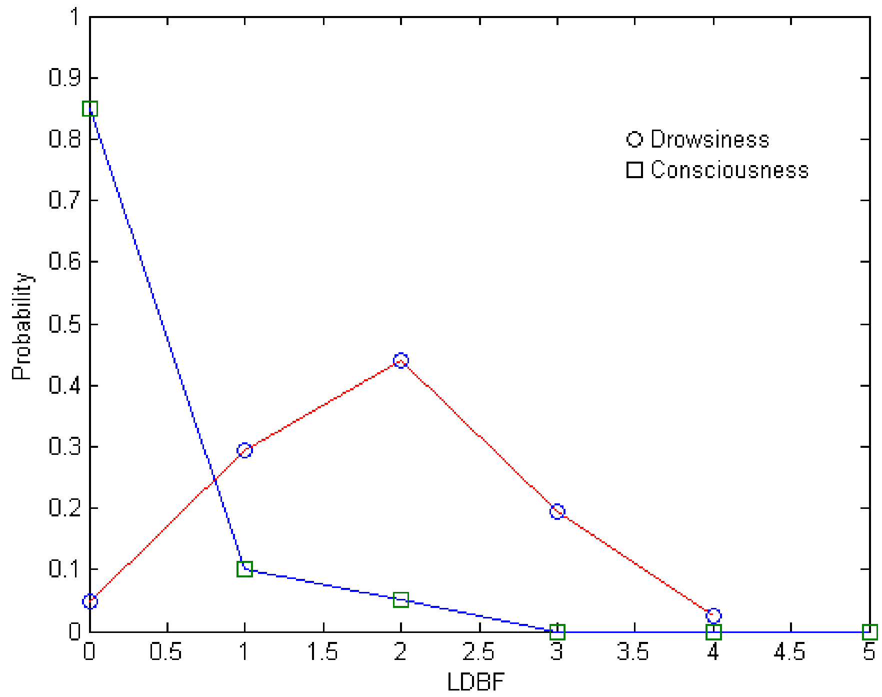


Fig. 4.7. The histograms of LDBF under consciousness and drowsiness for OTI = 8.25 seconds

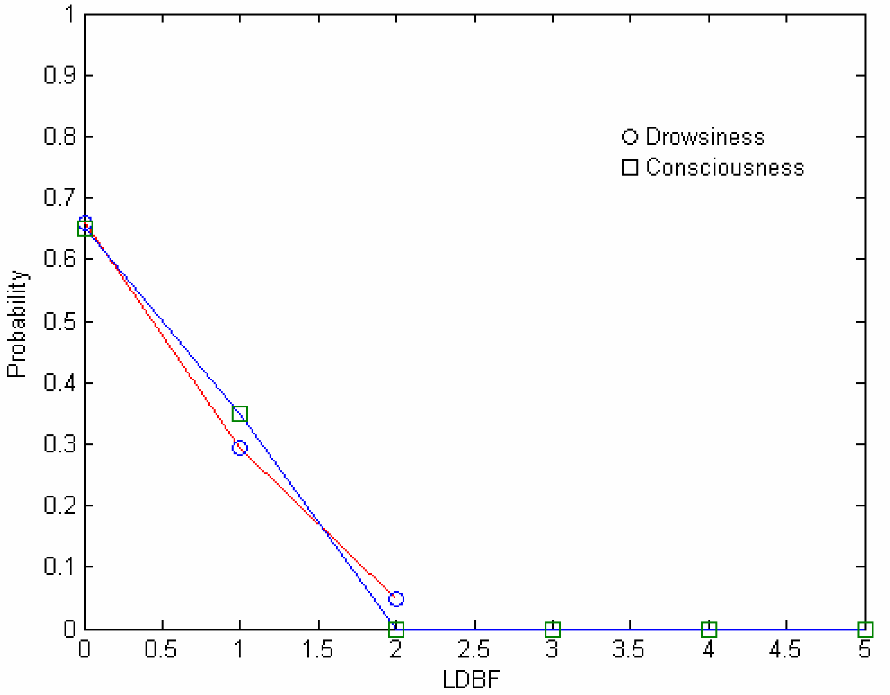


Fig. 4.8. The histograms of LDBF under consciousness and drowsiness for OTI = 2 seconds

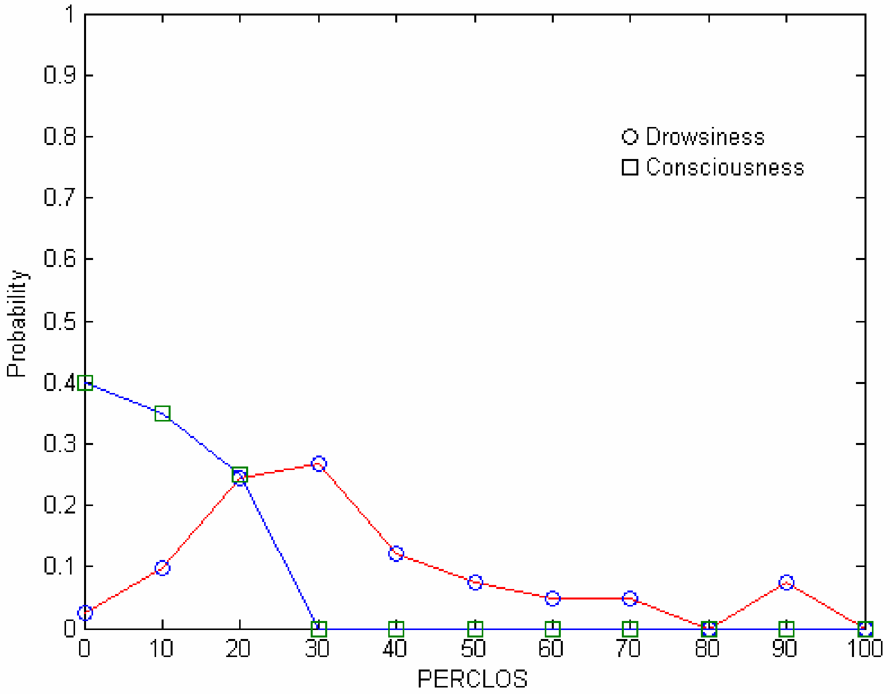


Fig. 4.9. The histograms of PERCLOS under consciousness and drowsiness for OTI = 8.25 seconds.

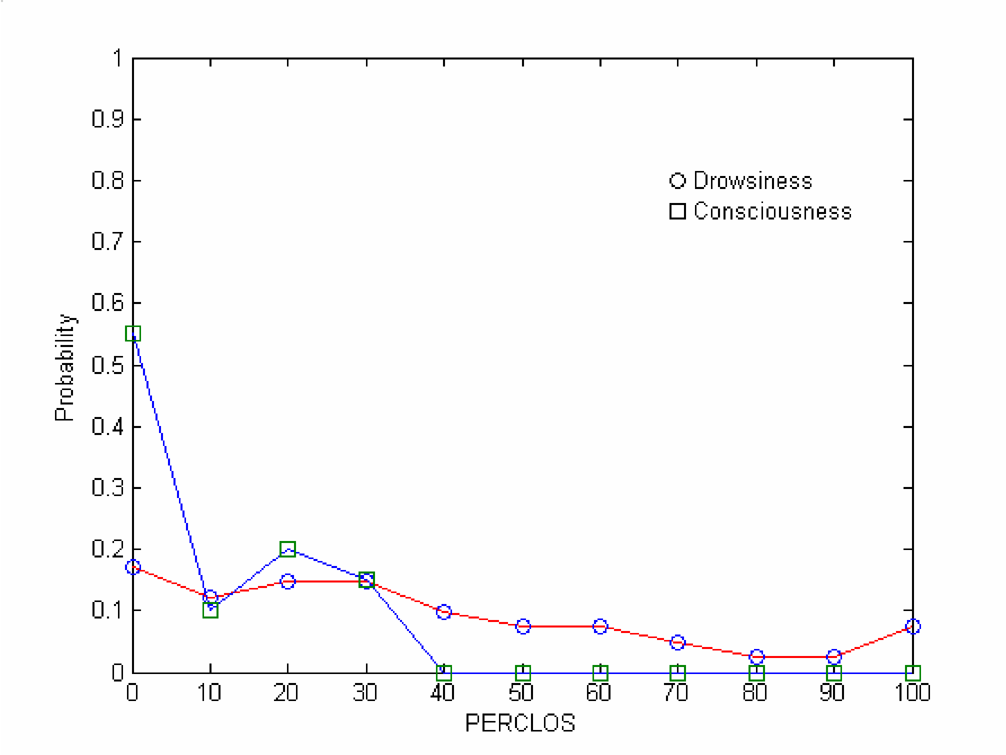


Fig. 4.10. The histograms of PERCLOS under consciousness and drowsiness for OTI = 2 seconds.

Table III. Parameter settings of the three algorithms for conscious/drowsy states

Method Subject State	LDBF	PERCLOS	Our Method by Fuzzy Integral *e*
Conscious	< 0.8	< 0.202	< 0.380
Drowsy	≥ 0.8	≥ 0.202	≥ 0.380

4.3.2 Drowsiness Detection Throughput Rate Speeding Up Using Sliding Window Strategy

For on-line drowsiness detection, in each processing iteration of the image frame, we deleted the oldest eye open/closed state, shifted all the remaining eye states one position left, and stored the most recent eye open/closed states of a subject for 8.25 seconds, which was the best OTI we have chosen above. Then, with our frame processing rate of 15 frames per second in mind, the LDBF and PERCLOS could be computed from the new 124 open/closed state of a subject and subsequently his consciousness/drowsiness state could thus be determined. Under this strategy, we could update LDBF and PERCLOS values and then make a conscious/drowsy state determination at image frame rate of $\frac{1}{15}$ seconds rather than OTI of 8.25 seconds. It is to be noted that the drowsiness detection throughput rate by our sliding window strategy via image frame updating scheme was much quicker than those reported in [24] and [9].

4.4 The Drowsiness Detection Accuracy Comparison

To combine information from LDBF and PERCLOS, we first computed the two means of the drowsiness and consciousness classes for each feature and then obtained the corresponding objective evidence function, $h(x)$ [27]. By assigning the desired sum of fuzzy densities equal to 1.2 and using the LDBF and PERCLOS features' drowsiness detection accuracies as a basis [27], we obtained the degrees of importance as $g(\{\text{LDBF}\})$ = 0.625 and $g(\{\text{PERCLOS}\}) = 0.575$, respectively. With the above g and h functions defined, we fused LDBF and PERCLOS features and plotted the drowsiness and consciousness data of test subjects versus fuzzy integral values, as shown in Fig. 4.11. From this figure, the subject was conscious if the fused fuzzy integral value e was less than or equal to 0.380, whereas the subject was drowsy if the fuzzy integral value e was larger than 0.380. The detection accuracy of the proposed method was 95.1% and the equivalent decision boundary by the fuzzy integral approach is shown in Fig. 4.12. This boundary was dominantly determined by PERCLOS value of 0.202 if LDBF was inside the range [0, 0.8], and the sample points summarizing the processed video image frames of subjects were almost classified to drowsiness category if LDBF value was larger than 0.8. The drowsiness recognition accuracies of LDBF, PERCLOS, and the proposed method were 91.8%, 86.8%, and 95.1%, respectively, as summarized in Table IV. Our method outperformed LDBF and PERCLOS by 3.3% and 8.3%, respectively. More insights into the function of fuzzy integral method will be given below.

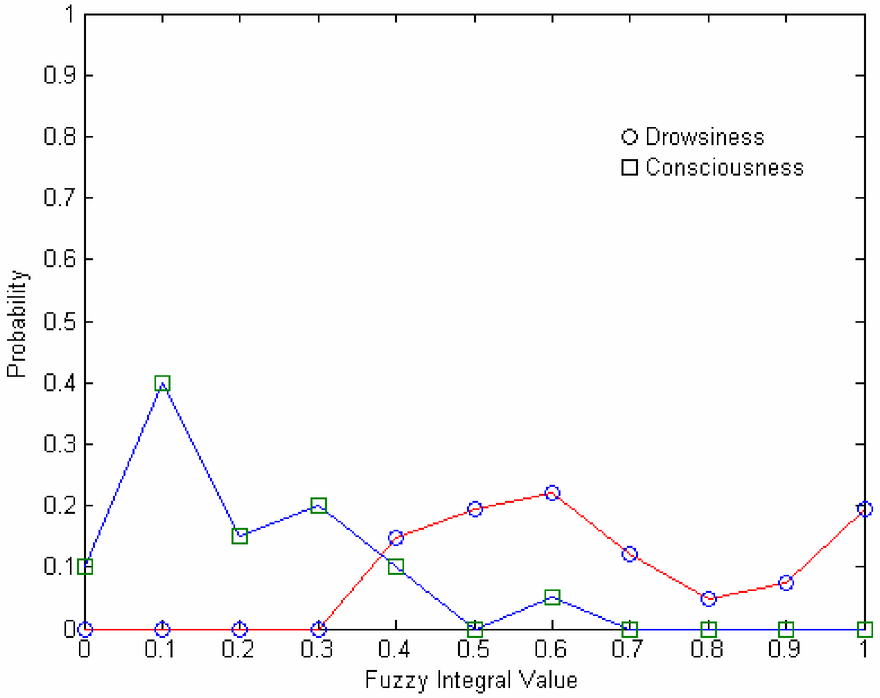


Fig. 4.11. The histograms of the fuzzy integral value *e* under consciousness and drowsiness.

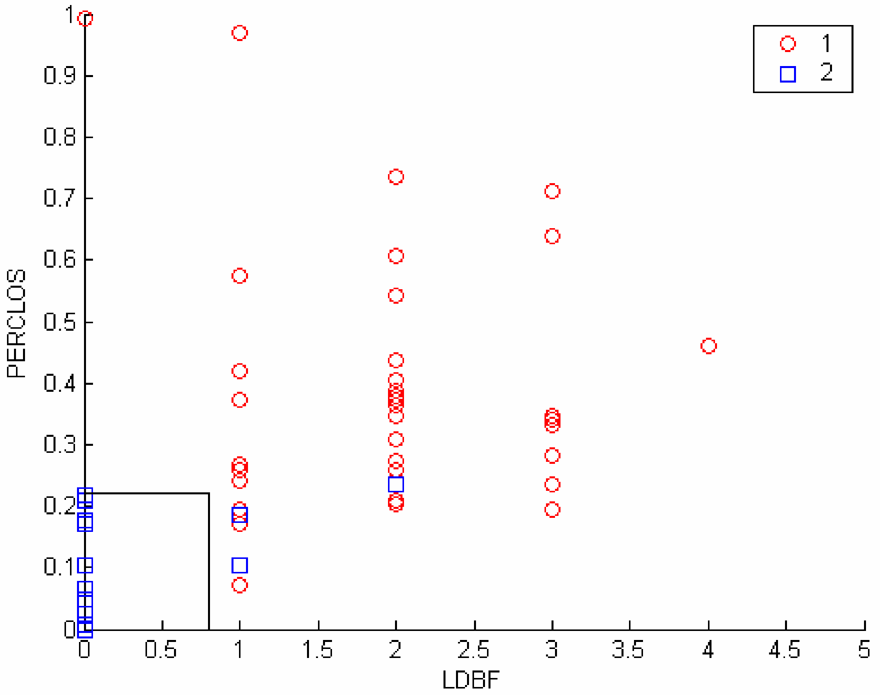


Fig. 4.12. The decision boundary between drowsiness and consciousness classes for the fuzzy integral method. ○ and □ represent drowsiness and consciousness samples, respectively.

Table IV. The drowsiness detection accuracy comparison

	Number of Samples	Number of Errors	Error Rate(%)	Accuracy (%)
LDBF	122	10	8.2	91.8
PERCLOS	122	16	13.2	86.8
Our Method	122	6	4.9	95.1

4.4.1 Examples to Illustrate the Fuzzy Integral Improvement in Drowsiness Detection Accuracy

In the following, we will demonstrate that by either LDBF or PERCLOS method may fail due to single feature utilization, and it could be corrected by incorporating the other feature through fuzzy integral approach. To this end, we illustrate our fuzzy integral improvement by six drowsy samples and six conscious ones among the 18 test subjects as shown in Table V. According to the experiments, LDBF method failed with one test sample, sample L in drowsy condition, and PERCLOS failed with two samples, sample B in conscious condition and sample J in drowsy condition. Our proposed method, however, outperformed these two methods producing no erroneous recognition. Sample L, in severely drowsy condition, was falsely determined to be conscious by the LDBF feature alone and could be accurately classified to be drowsy owing to its high PERCLOS value by our fuzzy integral. Sample B was falsely detected to be drowsy by PERCLOS feature only due to its PERCLOS value greater than 0.202, and this error could be corrected using its LDBF value. Sample J was falsely detected to be conscious due to its low PERCLOS value only. Likewise, this mis-classification could be corrected by its proper LDBF value.

Table V. A few comparisons of sampled testees by the three detection algorithms

Subject State Method	Conscious Condition					
Subject	A	B	C	D	E	F
LDBF	0	0	0	0	0	0
PERCLOS	0.169	0.218	0.048	0.177	0.105	0.02
Our Method	0.177	0.346	0.164	0.205	0.097	0.198
Subject State Method	Drowsy Condition					
Subject	G	H	I	J	K	L
LDBF	2	3	1	3	4	0
PERCLOS	0.735	0.638	0.420	0.194	0.460	0.993
Our Method	1.000	1.000	0.575	0.625	1.000	0.575

4.5 The Implementation for a driver drowsiness detection system

Owing to the satisfactory testing result above, the proposed algorithm was implemented in a real-world driver drowsiness detection and warning system. As shown in Fig. 4.13, the system was realized on a Centrino 1.5 GHz notebook with Sony EVI-D100 camera, which is mounted on the dashboard of the car. The captured and processed images of the real-time monitoring of a driver’s vigilance are shown in Fig. 4.14. To alert the driver to drowsiness, the system would keep on alarming by beepers if he is in sleepy state.

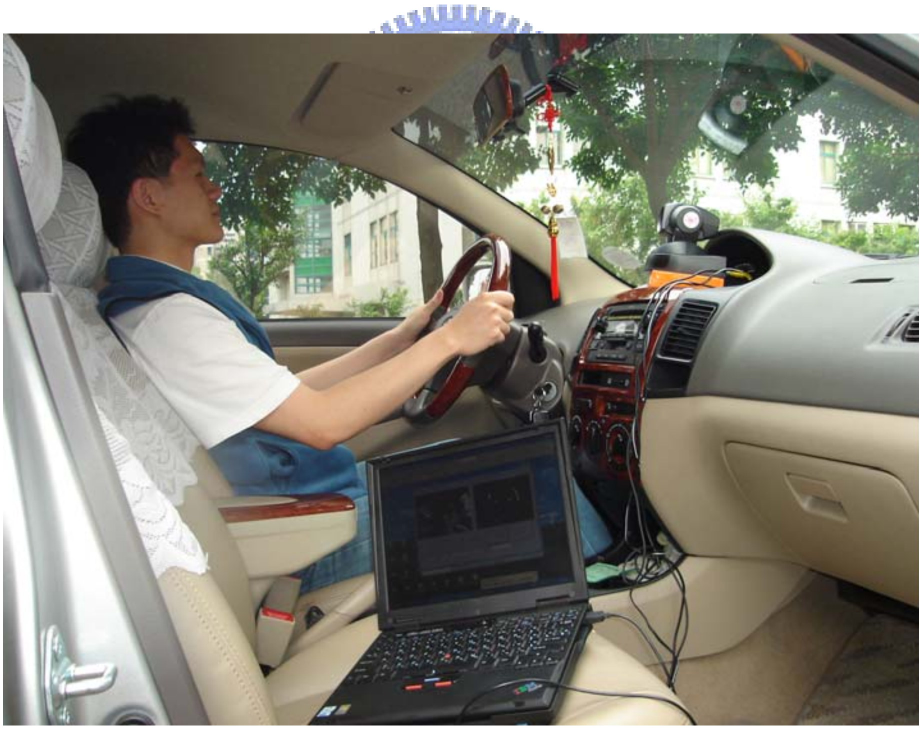


Fig. 4.13. The setups of driver drowsiness detection system: a CCD camera mounted on the dashboard and connected to a notebook.

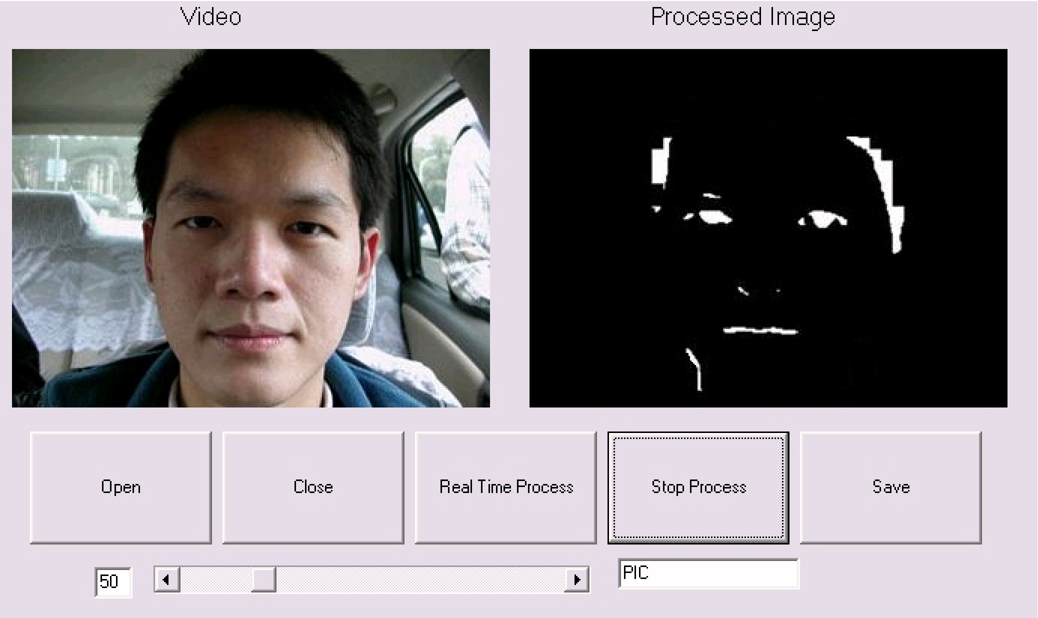


Fig. 4.14. The real-time monitoring of a driver’s vigilance: as shown in the program window, a driver’s facial image is captured and processed further to locate one of the eyeballs.

Chapter 5

Conclusion

This thesis presented a method to drowsiness detection system via eye detection, which is also applicable to wearing glasses or sunglasses causes. In particular the eye detection module consists of three parts, sunglasses area detection, sunglasses region enhancement, and eyeball extraction, by image processing techniques such as image segmentation, image enhancement, and edge detection. Extensive experiments have been carried out to demonstrate the effectiveness of the proposed method. Besides, our drowsiness detection system utilizes LDBF and PERCLOS features to determine the driver's visual attention level. We first investigated the conscious/drowsy distributions of these two features under different OTI's and then selected the best OTI for these two features altogether. For real-time drowsiness detection, the drowsiness detection throughput rate is greatly enhanced by our sliding window strategy via every image frame updating strategy. Afterwards, we employed a fuzzy integral to combine these two features and raised the drowsiness detection accuracy. Our fuzzy integral based approach outperformed LDBF and PERCLOS approaches by 3.3% and 8.3%, respectively. The proposed method provides a safe, reliable, and effective means for drowsiness detection. In practice, we have also applied our algorithm in the real-world driver safety monitoring systems. The test result of drivers' drowsiness detection is successful and satisfactory.

For future work, we try to solve the noise caused by shadows, light reflection of lenses and frames of metallic sunglasses, and the overlapping of eyes, eyebrows, and sunglasses. Besides, there are still a number of issues that remain to be addressed in the drowsiness detection system. These include improvement of its adaptability to changes in ambient brightness, assurance of reliability and attainment of a compact system design.

References

[1] K. Ogawa and M. Shimotani, “Drowsiness detection system,” *Technical Report of Mitshbish Electric*, Mar. 1997. (http://www.mitsubishielectric.com)

[2] H. Ueno, M. Kaneda, and M. Tsukino, “Development of drowsiness detection system,” in *Proc. IEEE Conf. On Vehicle Navigation & Information Systems*, May 1994.

[3] M Kaneda, H Iizuka, H Ueno, M Hiramatsu, and M Taguchi, “Development of a drowsiness warning system,” in *Proc. 14th Int. Conference on Enhanced Safety of Vehicles*, Munich, May 1994.

[4] D. Chai and K. N. Ngan, “Face segmentation using skin-color map in videophone applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 551–564, 1999.

[5] J. Y. Chang and J. L. Chen, “Automated facial expression recognition system using neural networks,” *J. of the Chinese Institute of Engineers*, vol. 24, no. 3, pp. 345–356, 2001.

[6] I. Cohen, N. Sebe, A. Garg, M. S. Lew, and T. S. Huang, “Facial expression recognition from video sequences,” in *Proc. IEEE Multimedia and Expo Conf.*, vol. 2, Aug. 2002.

[7] M. Kirby and L. Sirovich, “Application of the Karhunen-Loeve procedure for the characterization of human faces,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 1, pp. 103–108, 1990.

[8] C. D. Wylie, J. C. Shultz, M. M. Miller, and R. R. Mackie, “Commercial motor vehicle driver fatigue and alertness study,” *Project Report* (Report No. FHWAMC-97-002), Washington, D. C: Federal Highway Administration Office

of Motor Carroers, Oct. 1996.

[9] W. W. Wierwille, S. S. Wreggit, C. L. Kim, L. A. Ellsworth, and R. J. Fairbanks, "Research on vehicle-based driver status/performance monitoring: development, validation, and refinement of algorithms for detection of driver drowsiness," *Tech Report,* (No. DOT HS 808 638), Washington, D. C: National Highway Traffic Safety Administration, Dec. 1994.

[10] J. Y. Chang, S. T. Lo, and C. F. Li, "Fuzzy type identification and recognition of handwritten Chinese characters," *Int. J. of Fuzzy Systems*, vol. 2, no. 4, pp. 291–299, Dec. 2000.

[11] S. B. Cho and J. H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 2, pp. 380–384, Feb. 1995.

[12] J. Keller, H. Qiu, and H. Tahani, "Fuzzy integral and image segmentation," in *Proc. North American Fuzzy Information Processing Soc.*, New Orleans, pp. 334–338, June 1986.

[13] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 641–647, 1994.

[14] E. Land, "An alternative technique for the computation of the designator in the retinex theory of color vision", in *Proc. Nat. Acad, Sci.*, vol.83, pp. 3078–3080, 1986.

[15] E. Land, "Recent advances in retinex theory and some implications for cortical computations," in *Proc. Nat. Acad. Sci.*, vol. 80, pp. 5163–5169, 1983.

[16] E. Land, "Recent advances in retinex theory," *Vis. Res.*, vol. 26, pp. 7–21, 1986.

[17] A.C. Hurlbert, "Formal connections between lightness algorithms," *J. Opt. Soc. Amer. A*, vol. 3, pp. 1684–1693, 1986.

[18] D. J. Jobson, Z. Rahman, and G. A. Woodell, “Properties and performance of a center/surround retinex,” *IEEE Trans. Image Processing*, vol. 6, pp. 451–462, Mar. 1997.

[19] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A multi-scale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Tran. Image Processing*, vol. 6, pp. 965-976, July 1997.

[20] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Retinex processing for automatic image enhancement," in *IS&T/SPIE Electronic Imaging 2002. The Human Vision and Electronic Imaging VII Conference*, 2002, vol. 4662, pp. 390–401.

[21] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, “Automatic image segmentation by integrating color-edge extraction and seeded region growing,” *IEEE Trans. Image Processing*, vol. 10, pp. 1454–1466, 2001.

[22] J. Fan, R. Wang, L. Zhang, D. Xing, and F. Gan, “Image sequence segmentation based on 2-D temporal entropy,” *Pattern Recognition Lett.*, vol. 17, pp. 1101–1107, 1996.

[23] W. W. Weirwille, “Overview of research on driver drowsiness definition and driver drowsiness detection,” in *Proc. 14th International Technical Conference on Enhanced Safety of Vehicles*, Munich, May 1994.

[24] K. Ogawa and M. Shimotani, “A drowsiness detection system,” *Mitsubishi Tech. Reports*, vol. 78, pp. 13–16, 1997.

[25] P. P. Caffier, U. Erdmann, and P. Ullsperger, “Experimental evaluation of eye-blink parameters as a drowsiness measure,” *Eur. J. Applied Physiology*, vol. 89, pp. 319–325, 2003.

[26] D. F. Dinges and R. Grace, “PERCLOS: a valid psychophysiological measure of alertness as assessed by psychomotor vigilance,” *Federal Highway*

Administration, Office of Motor Carriers, USA, 1998.

[27] H. Tahani and J. M. Keller, “Information fusion in computer vision using the fuzzy integral,” *IEEE Trans*. *Syst.*, *Man*, *Cybern*., vol. 20, pp. 733–741, 1990.