Real-Time System for Driver Fatigue Detection by RGB-D Camera

LIYAN ZHANG, University of California, Irvine
FAN LIU and JINHUI TANG, Nanjing University of Science and Technology

Drowsy driving is one of the major causes of fatal traffic accidents. In this article, we propose a real-time system that utilizes RGB-D cameras to automatically detect driver fatigue and generate alerts to drivers. By introducing RGB-D cameras, the depth data can be obtained, which provides extra evidence to benefit the task of head detection and head pose estimation. In this system, two important visual cues (head pose and eye state) for driver fatigue detection are extracted and leveraged simultaneously. We first present a real-time 3D head pose estimation method by leveraging RGB and depth data. Then we introduce a novel method to predict eye states employing the WLBP feature, which is a powerful local image descriptor that is robust to noise and illumination variations. Finally, we integrate the results from both head pose and eye states to generate the overall conclusion. The combination and collaboration of the two types of visual cues can reduce the uncertainties and resolve the ambiguity that a single cue may induce. The experiments were performed using an inside-car environment during the day and night, and theyfully demonstrate the effectiveness and robustness of our system as well as the proposed methods of predicting head pose and eye states.

Categories and Subject Descriptors: C.1.2 [**Systems and Applications**]: Multimedia and Vision Systems

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: RGB-D cameras, driver fatigue detection system, head pose, eye state

ACM Reference Format:
Liyan Zhang, Fan Liu, and Jinhui Tang. 2015. Real-time system for driver fatigue detection by RGB-D camera. ACM Trans. Intell. Syst. Technol. 6, 2, Article 22 (March 2015), 17 pages.
DOI: http://dx.doi.org/10.1145/2629482

1. INTRODUCTION

The increasing number of traffic accidents due to drowsy driving has become a serious issue affecting people's lives. According to statistical analysis, 10%–20% of traffic accidents involve drivers with a diminished vigilance level caused by fatigue. In the trucking industry, approximately 60% of fatal truck accidents are related to driver fatigue [Ji and Yang 2002; Bergasa et al. 2006]. Therefore, developing a system that can automatically detect driver fatigue and issue an alert as soon as the driver loses adequate attention to the traffic conditions is an essential action to prevent accidents.

This work was supported in part by the 973 Program under grant 2014CB347600, in part by the National Nature Science Foundation of China under grants 61103059 and 61173014, and in part by the Program for New Century Excellent Talents in University under grant NCET-12-0623.

Authors' addresses: L. Zhang, Department of Computer Science, University of California, Irvine, Irvine, California, USA, 92697; email: liyanz@uci.edu; F. Liu and J. Tang (corresponding author), Department of Computer Science, Nanjing University of Science and Technology, Nanjing, China, 210094; emails: fanliu.njust@gmail.com, tangjh1981@acm.org.
Author's current address: L. Zhang, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics.

DOI: http://dx.doi.org/10.1145/2629482

In the past two decades, much research effort has been dedicated to the development of systems for detecting driver fatigue. Different techniques have been proposed and leveraged in these systems, including physiological measures (e.g., brain waves, heart rate, pulse rate, and respiration) and vehicle parameter monitoring (e.g., speed, lateral position, and steering wheel movements). However, these techniques are either infeasible or insufficient for the real applications, because physiological measure–based methods are intrusive and the vehicle behaviors are usually subjected to limitations such as the vehicle type, driver experience, and driving conditions.

More recently, the advancement of computer vision techniques makes it possible to detect driver fatigue by observing and analyzing drivers' facial features such as eye state, head pose, and face pose [Ji and Yang 2002; Smith et al. 2003; Bergasa et al. 2006; Tang et al. 2012; Zhang et al. 2013a]. The visual-based system reveals its superiority in convenience and nonintrusion; however, the accuracy of automatic facial feature extraction, such as head pose estimation and eye state detection, is far from satisfaction in the real applications due to the large variations in illumination conditions and possible face poses. In addition, most current visual-based systems merely rely on the single visual cue, which may encounter difficulties when the required visual features cannot be acquired accurately or reliably in the real applications.

To handle these issues, we propose leveraging the emerging RGB-D cameras in the driver fatigue detection system. Compared to the conventional RGB cameras, the RGB-D cameras can offer extra depth data, which can benefit the task of head detection and head pose estimation and is also insensitive to the variation of illumination and conditions. Therefore, in this article, we propose a real-time system to detect driver fatigue by RGB-D cameras, where two important visual cues (head pose and eye state) are leveraged simultaneously. We first explore the effective methods to extract high-level facial features (like head motions and eye states) utilizing both RGB image data and depth data, and then we systematically combine these features to generate the overall conclusion.

In this work, we first present a real-time 3D head motion estimation method utilizing both RGB image data and depth data obtained from the RGB-D cameras. Based on a rigid-body motion model, we derive the linear depth and optical flow constraint equations, respectively. These constraints are combined into a single linear system, from which the head motion vector is recovered by minimizing least squares. To enhance the robustness of this method, the weights between the RGB image data and depth data can be adjusted according to their quality in real conditions. Thus, our system is still applicable even when only depth or RGB data is available.

We then present an efficient eye state detection method leveraging the Weber local binary pattern (WLBP) feature, which is a novel local descriptor proposed in our prior work [Liu et al. 2013a] combining the advantages of the Weber local discriptor (WLD) [Chen et al. 2010] and the local binary pattern (LBP) [Ojala et al. 2002]. Given an eye image, the WLBP histogram is extracted, with each bin of the histogram regarded as a feature of the eye. Leveraging the extracted eye features and support vector machines (SVMs), we trained a nonlinear classifier to recognize the eye state. Experimental results demonstrate the significant superiority of WLBP in effectiveness and robustness compared to that of WLD and LBP.

We finally integrate the results from the two visual cues, head motion estimation, and eye state detection, and then generate the overall conclusion. Although uncertainty exists in performing each visual cue individually, by integrating multiple cues systematically we can reduce the uncertainty and resolve the ambiguity present in the information from a single source. Thus, the combination and collaboration of the two different types of visual cues can dramatically improve the reliability and robustness of our system. The experiments using the inside-car environment demonstrate that the integration of eye state and head pose for driver fatigue detection can achieve more robust results than by using each one individually—for example, the nodding behaviors that cannot be detected by eye state detection can easily be detected by head pose estimation.

The contributions of this article are as follows:

(1) We propose a real-time driver fatigue detection system utilizing RGB-D cameras, where two important visual cues (head pose and eye state) are leveraged simultaneously.
(2) We present a real-time 3D head motion estimation method utilizing both RGB image data and depth data.
(3) We present an efficient eye state detection method leveraging the novel WLBP features.
(4) Experiments using the inside-car environment demonstrate the effectiveness of the proposed system.

The rest of this article is organized as follows. We start by introducing the related work in Section 2. In Section 3, we describe the proposed system in detail, including four major modules: (a) data acquisition, (b) head detection, (c) head pose detection, and (d) eye state detection. The proposed approaches are empirically evaluated in Section 4. We conclude in Section 5 by highlighting key points of our work.

2. RELATED WORK

In the past two decades, the development of driver fatigue monitoring systems has attracted the attention of researchers. Different types of techniques have been proposed and investigated. Among them, the techniques based on physiological measures such as brain waves, heart rate, pulse rate, and respiration can achieve the most accurate performance. Two representative projects following this research direction are MIT Smart Car [Healey and Picard 2000] and the Advanced Safety Vehicle (ASV) [Albert et al. 2002]. However, these techniques are infeasible in practice, as they require physical contact with drivers (e.g., attaching electrodes), which usually cause annoyance to drivers. Other equipment monitoring eye and gaze movements using a helmet or special contact lenses [Anon 1999] have been leveraged; however, these techniques are still not acceptable in the real applications even though they are less intrusive.

Another research direction focuses on analyzing the indirect behaviors of the vehicle, such as speed, lateral position, and steering wheel movements, to estimate the drivers' state of vigilance. These techniques are implemented nonintrusively, but they are subject to several limitations, such as the vehicle type, driver experience, and driving conditions [Hiroshi et al. 1994].

Recently, the advancement in computer vision techniques provides a natural and nonintrusion solution to handle the fatigue detection problem [Ji and Yang 2002; Smith et al. 2003; Zhang et al. 2013b]. People in fatigue show some visual behaviors that are easily observable from changes in their facial features (e.g., eyes, head, and face). These visual characteristics can be extracted and monitored by analyzing the images captured from the camera placed in front of drivers using computer vision techniques. It has been concluded that computer vision represents the most promising noninvasive technology for fatigue detection in drivers [Ji and Yang 2002]. Many efforts have been reported on developing vision-based fatigue detection systems [Ji and Yang 2002; Smith et al. 2003; Bergasa et al. 2006]. The work can be classified into two categories according to the sensor type: popular color video camera and near-infrared (NIR) camera.

Despite the success of the existing approaches/systems, fatigue detection is still a difficult task, even for humans, because there are many factors involved, such as changing illumination conditions and a variety of possible face poses. In addition,

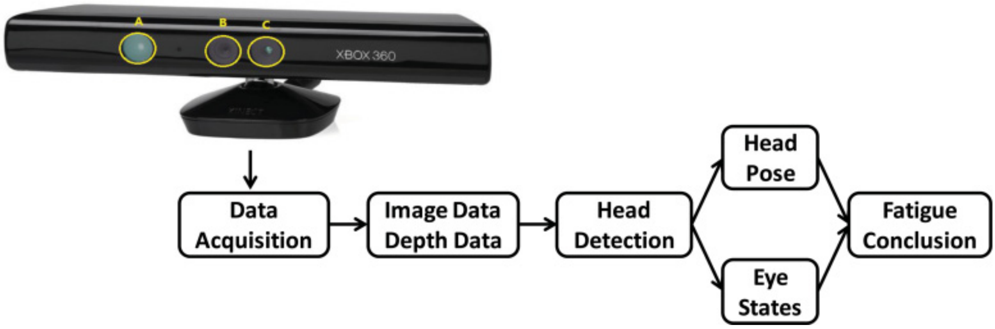

Fig. 1. The General Framework of the Driver Fatigue Detection System. (Data is acquainted from Mircosoft Kinect: (A) laser projector, (B) RGB camera, (C) monochrome CMOS camera.)

systems relying on a single visual cue may encounter difficulties when the required visual features cannot be acquired accurately or reliably, which often happens in real conditions. However, all of those visual cues—imperfect individually—can reduce the uncertainty and resolve the ambiguity present in the information from a single source if combined systematically.

Research in head pose estimation can be conveniently categorized by the type of input data on which they rely, such as RGB image data or depth data. However, the conventional RGB image-based head pose estimation methods encounter many difficulties due to pose and illumination variations. Moreover, the annotation of head poses from RGB images is an error-prone task itself. With the development and availability of depth-sensing technologies, researchers have focused on using depth information to solve the problem of head pose estimation, either as a unique cue [Fanelli et al. 2011; Breitenstein et al. 2008] or in combination with RGB image data [Cai et al. 2010; Seemann et al. 2004; Liu et al. 2013b]. The work proposed by Cai et al. [2010] aimed to handle the face tracking problem using RGB-D cameras; the work proposed by Seemann et al. [2004] explored the head pose estimation problem utilizing stereo video data; the work proposed by Wu et al. [2010] aimed to estimate the body orientation using RGB-D sensors. None of these techniques can be directly applied to solve our problem.

3. THE PROPOSED DRIVER FATIGUE DETECTION SYSTEM

In the proposed driver fatigue detection system, we utilize the emerging RGB-D cameras, which can provide the RGB image data as well as the depth data. By analyzing these data, we can automatically extract the driver's head pose and eye state information, based on which the overall fatigue conclusion can be generated. Figure 1 demonstrates the general framework of this system, including the four major modules of data acquisition, head detection, head pose detection, and eye state detection, which will be described in detail as follows.

3.1. Data Acquisition

The RGB image and depth data are captured from the Kinect, which is a motion-sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. The device consists of a multiarray microphone, an RGB camera, a monochrome CMOS camera, and an infrared laser projector (Figure 1). The laser projector produces a structured light pattern in the scene, which is imaged by the CMOS camera. The device is capable of outputting RGB and range images with pixels at 30 frames per second. Microsoft has released a noncommercial Kinect software development kit (SDK)

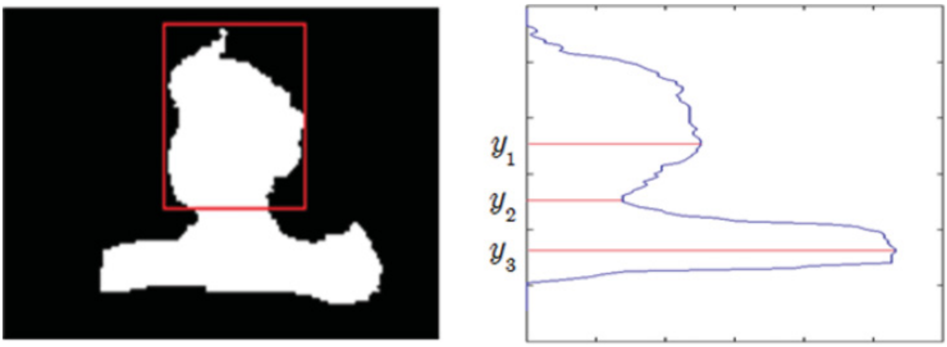

Fig. 2. Head detection: binary image of the upper body (left) and horizontal projection of the histogram (right).

for Windows. It provides Kinect capabilities to developers who build applications with C++, C#, or Visual Basic by using Microsoft Visual Studio 2010.

3.2. Head Detection

Head detection is a key component of head pose estimation. It is also a necessary preprocessing step for eye detection. Although many sophisticated head detection methods are available, they usually use intensity images as the input to the detection system [Nguyen and Smeulders 2006; Zhao and Nevatia 2004] and are easily affected by illumination variations.

In this article, we perform a simple head detection method leveraging the depth data, which is robust against illumination variations. In our application, we assume that the experimental environment is relatively constrained, where the background data can be easily obtained and the simple approach can work effectively. First, we extract the background depth data, which can be regarded as an initialization step. Then a difference matrix is computed by subtracting background from a new depth array. If the difference is below a threshold, the pixel is set to be zero and otherwise it will be retained, resulting in a matrix containing the depth information of the foreground. According to the prior knowledge about human body, a pixel and its neighbors are considered to be on the same object if the depth difference between them is less than a threshold (0.1 to 0.2 m). Any segmented object that contains fewer pixels than a particular number is considered as nonhuman and discarded.

To get a more accurate head detection result, we perform a simple head/shoulder separation algorithm based on a vertical projection signature operation by effectively leveraging the depth image. A similar method was employed in Nguyen and Smeulders [2006]. As illustrated in Figure 2, the binary image is projected to the vertical direction, where y_1 and y_3 are two peaks in the projection histogram, and y_2 is the minimal value between the two peaks. It is straightforward that y_2 corresponds to the position of the head bottom area. Therefore, the bottom boundary of the head area is segmented by finding the minimal value between the two peaks of the projection histogram. Then the head area is eventually located by finding the top-, left-, and right-most pixel in the area above the bottom boundary. The four pixels constitute the boundaries of the rectangle containing the head.

3.3. Head Pose Estimation

To recover the head pose, we assume that the sensor is rigid and the global motion of the head is a kind of rigid motion. Then the motion of the head relative to the sensor can be parameterized by a rigid motion vector μ, which includes the 3D rotation

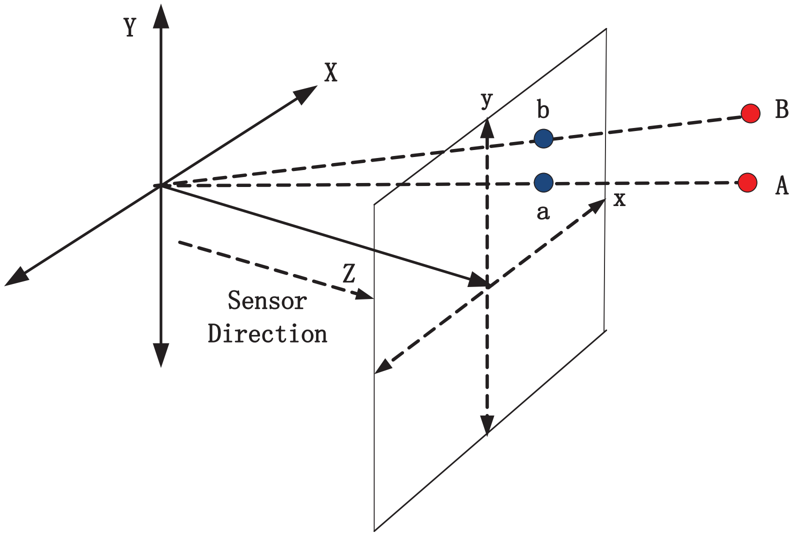

Fig. 3. Relationship between 3D motion from A to B and 2D motion from a to b.

angles (W_X, W_Y, W_Z) and the 3D translation vector (T_X, T_Y, T_Z). As shown in Figure 3, when the 3D coordinate of a head point is $A = (X_A, Y_A, Z_A)^T$ in the camera-centered 3D coordinate system, then the new location of the point A transformed by the rigid motion vector μ is $B = (X_B, Y_B, Z_B)^T$. The relationship between A and B can be described by the following equation:

$$B = RA + T, \tag{1}$$

where the 3D translation vector $T \in \mathbb{R}^{3\times1} = (T_X, T_Y, T_Z)^T$ and the 3D rotation matrix $R \in \mathbb{R}^{3\times3}$ is related to the 3D rotation angles (W_X, W_Y, W_Z). When the rotation is small, the rotation matrix can be approximated as follows, based on the exponential map [Bregler and Malik 1998]:

$$R = \begin{bmatrix} 1 & -W_Z & W_Y \\ W_Z & 1 & -W_X \\ -W_Y & W_X & 1 \end{bmatrix}. \tag{2}$$

Then the 3D rigid motion between point A and B can be described by

$$\begin{bmatrix} \triangle X_{AB} \\ \triangle Y_{AB} \\ \triangle Z_{AB} \end{bmatrix} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} + \begin{bmatrix} 0 & -W_Z & W_Y \\ W_Z & 0 & -W_X \\ -W_Y & W_X & 0 \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}. \tag{3}$$

3.3.1. Depth Constraint. The time-varying depth map from the Kinect can be viewed as a function of the form $Z(X, Y, t)$. Taking a full time derivative of Z via the chain rule, the following equation is obtained:

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = \frac{\partial Z}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}t} + \frac{\partial Z}{\partial Y}\frac{\mathrm{d}Y}{\mathrm{d}t} + \frac{\partial Z}{\partial t}. \tag{4}$$

Equation (4) will be called the *depth rate constraint equation* [Kondori et al. 2011], in which the three partial derivatives of Z are denoted by

$$p = \frac{\partial Z}{\partial X}, q = \frac{\partial Z}{\partial Y}, Z_t = \frac{\partial Z}{\partial t}, \tag{5}$$

where the vector $(\frac{\mathrm{d}X}{\mathrm{d}t}, \frac{\mathrm{d}Y}{\mathrm{d}t}, \frac{\mathrm{d}Z}{\mathrm{d}t})$ denotes the differentiation with respect to time, which can be computed by

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \frac{\triangle X_{AB}}{\triangle t}, \frac{\mathrm{d}Y}{\mathrm{d}t} = \frac{\triangle Y_{AB}}{\triangle t}, \frac{\mathrm{d}Z}{\mathrm{d}t} = \frac{\triangle Z_{AB}}{\triangle t}. \tag{6}$$

Therefore, Equation (4) can be written in the following form:

$$\triangle Z_{AB} = p\triangle X_{AB} + q\triangle Y_{AB} + Z_t\triangle t. \tag{7}$$

By substituting $\triangle X_{AB}$, $\triangle Y_{AB}$, and $\triangle Z_{AB}$ of Equation (3) into Equation (7), we obtain the following equation:

$$-pT_X - qT_Y + T_Z + rW_X + sW_Y + uW_Z = Z_t\triangle t, \tag{8}$$

where $r = Y_A + qZ_A$, $s = -X_A - pZ_A$, and $u = pY_A - qX_A$. If there are n pixels in the head area, the resulting n equations can be written in a matrix form as

$$\underbrace{\begin{bmatrix} -p_1 & -q_1 & 1 & r_1 & s_1 & u_1 \\ -p_2 & -q_2 & 1 & r_2 & s_2 & u_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -p_n & -q_n & 1 & r_n & s_n & u_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} T_X \\ T_Y \\ T_Z \\ W_X \\ W_Y \\ W_Z \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} (Z_t\triangle t)_1 \\ (Z_t\triangle t)_2 \\ \vdots \\ \vdots \\ \vdots \\ (Z_t\triangle t)_n \end{bmatrix}}_{b}. \tag{9}$$

3.3.2. Optical Flow Constraint. The 3D head rigid motion also forms a motion field in the head region of the RGB image plane from the Kinect. Optical flow, based on constant image brightness constraint over time, is used to estimate the motion filed between successive frames. If the time-varying RGB image from the Kinect is viewed as a function of the form $I(x, y, t)$, the well-known optical flow constraint equation [Horn and Schunck 1981] can be described as

$$\frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial I}{\partial t} = 0, \tag{10}$$

where the three partial derivatives of I can be denoted by

$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}. \tag{11}$$

Based on the pinhole camera model, we have the following equation,

$$x = \frac{F}{Z}X,\ y = \frac{F}{Z}Y, \tag{12}$$

where F is the camera focus length. By taking the derivative with respect to time, we have

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{F}{Z}\frac{\mathrm{d}X}{\mathrm{d}t} - \frac{FX}{Z^2}\frac{\mathrm{d}Z}{\mathrm{d}t} \\ \frac{\mathrm{d}y}{\mathrm{d}t} = \frac{F}{Z}\frac{\mathrm{d}Y}{\mathrm{d}t} - \frac{FY}{Z^2}\frac{\mathrm{d}Z}{\mathrm{d}t} \end{cases}. \tag{13}$$

By substituting Equation (3), Equation (6), and Equation (12) into Equation (13), we obtain the relationship between the instantaneous 3D motion vector and 2D motion vector:

$$\begin{cases} \triangle x = \frac{F}{Z}T_x - \frac{x_A}{Z}T_Z - \frac{x_A y_A}{F}W_x + (F + \frac{x_A^2}{F})W_Y - y_A W_Z \\ \triangle y = \frac{F}{Z}T_y - \frac{y_A}{Z}T_Z - (F + \frac{y_A^2}{F})W_x + \frac{x_A y_A}{F}W_Y + x_A W_Z \end{cases}. \tag{14}$$

By substituting Equation (12) and Equation (14) into Equation (10), we get the following equation:

$$pT_x + qT_y + kT_Z + rW_x + sW_Y + uW_Z = I_t \triangle t, \tag{15}$$

where

$$p = -\frac{I_x F}{Z}, q = -\frac{I_y F}{Z}, k = \frac{I_x x_A - I_y y_A}{Z}$$

$$r = \frac{x_A y_A}{F} + \left(F + \frac{y_A^2}{F}\right), s = -\frac{x_A y_A}{F} - \left(F + \frac{x_A^2}{F}\right), u = y_A - x_A.$$

Considering that there are n pixels in head area, the resulting n equations could be written in a matrix form as

$$\underbrace{\begin{bmatrix} p_1 & q_1 & k_1 & r_1 & s_1 & u_1 \\ p_2 & q_2 & k_2 & r_2 & s_2 & u_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_n & q_n & k_n & r_n & s_n & u_n \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} T_X \\ T_Y \\ T_Z \\ W_X \\ W_Y \\ W_Z \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} (I_t \triangle t)_1 \\ (I_t \triangle t)_2 \\ \vdots \\ \vdots \\ \vdots \\ (I_t \triangle t)_n \end{bmatrix}}_{c}. \tag{16}$$

3.3.3. Head Pose Estimation. The two linear systems according to depth and optical flow constraints can be combined into a single linear system:

$$y = Xw, \text{where } X = \begin{bmatrix} \lambda A \\ H \end{bmatrix}, y = \begin{bmatrix} \lambda b \\ c \end{bmatrix}. \tag{17}$$

The scaling factor, denoted as λ, controls the weighting of the depth constraints relative to the optical flow constraints. When depth data is more reliable than RGB image data, λ is set to a value higher than 1. When depth data is much noisier than RGB image data, λ is set to a small value. The value of λ is determined by the experimental environment. The experiments reveal that the depth data is robust to the variation of illumination but sensitive to the occlusions and driving vibrations. Besides, the depth data is not valid if the distance between the subject and sensors exceeds a certain value. The RGB data does not have this limitation but is very sensitive to the variations of illumination and poses. Therefore, in an environment where the illumination is relatively stable or the subject is very far from the sensors, the RGB information tends to be more reliable. On the other hand, if the illumination changes significantly, the depth data is more reliable.

In addition, the depth and optical flow constraints can also be used independently for pose estimation, which usually happens when one type of data, either depth or RGB image data, is not valid any more. The parameter w can be determined by solving the following optimization problem:

$$\min_{w} \parallel y - Xw \parallel^2 . \tag{18}$$

The least-squares solution to Equation (18) is given by

$$\hat{w} = (X^T X)^{-1} X^T y. \tag{19}$$

Consequently, the six degrees of freedom (DOF) of the head motion will be recovered. After 3D head motion between two successive image frames is solved by Equation (19), cumulative 3D head motion over time from the starting status can be calculated by summing up all of the rotation angles between every two successive image frames.

3.4. Eye State Detection

Eye states can be obtained by using features such as eye contour, irises, corners, and eyelids. Deng and Lai [1997] used improved deformable templates to locate eye features. However, this scheme is time consuming due to the process of energy minimizing and is hard to use as a real-time eye detector in image sequences. In Tian et al. [2000], the Gabor wavelet is applied to extract the features of eye corners, but the initial positions of the corners should be given manually in advance; this method is invalidated when the face is at a nonfrontal pose. In the work of Wang and Yang [2006], eye states are determined by detecting the circle of the iris. Researchers have begun to pay more attention to machine learning methods. In Senaratne et al. [2007], SVMs and the naive Bayes (NB) classifier are employed for comparison by simply using intensity values of eye blocks. In Xu et al. [2008] and Wu et al. [2010], Adaboost and SVM classifiers based on local binary pattern (LBP) histogram features are used to detect eye states.

However, in real-world conditions, the performance of the preceding algorithms is greatly influenced by noise and light changes. To improve the robustness to noise and light changes, we present a novel eye state detection method in this article. We consider the detection of eye states as a binary classification problem. The eyes can be classified into two categories: open (positive) or closed (negative). The eye features are extracted by WLBP, the novel local descriptor that we have proposed, which consists of two components: differential excitation and LBP. The differential excitation extracts perception features by Weber's law, whereas LBP can describe local features splendidly. By computing the two components, we obtain two images: the differential excitation image and the LBP image, from which the WLBP histogram features are extracted. We then trained an SVM classifier to classify the eye states as open or closed.

3.4.1. WLBP. To analyze the state of eyes, the first step is to detect their location. For face and eye detection, the FaceSDK of Luxand (available at http://www.luxand.com) was adopted because it has robust detection performance and allows $-30 \sim 30$ degrees of in-plane head rotation and $-10 \sim 10$ degrees of out-of-plane rotation.

After the detection of eyes, the next step is to extract local features as the representation. In this article, we choose to employ the WLBP feature, which is described in detail in Liu et al. [2013a]. As illustrated in Figure 4, for a given image, we compute the differential excitation value and the LBP value of every pixel by the uniform LBP operator $LBP_{P,R}^{u2}$ and the differential excitation operator, respectively. We then got two images, the differential excitation image and the LBP image, from which the 2D histogram $\{WLBP(s,t)\}$, $(s = 1, \ldots, S, t = 1, \ldots, T)$ will be constructed. Note that the size of this 2D histogram is $T \times S$, where S is the number of intervals of ξ and T is the total number of the LBP's patterns. In other words, in this 2D histogram, each column corresponds to a pattern t of the LBP, and each row corresponds to a differential excitation interval. Thus, the value of each cell corresponds to the frequency of the certain differential excitation interval and the LBP pattern t .

To enhance the discriminability, the 2D histogram is further encoded into a 1D histogram H. We use each row of the 2D histogram to form a 1D histogram $H(s)$, where $s = 1, \ldots, S$. Each subhistogram $H(s)$ corresponds to the differential

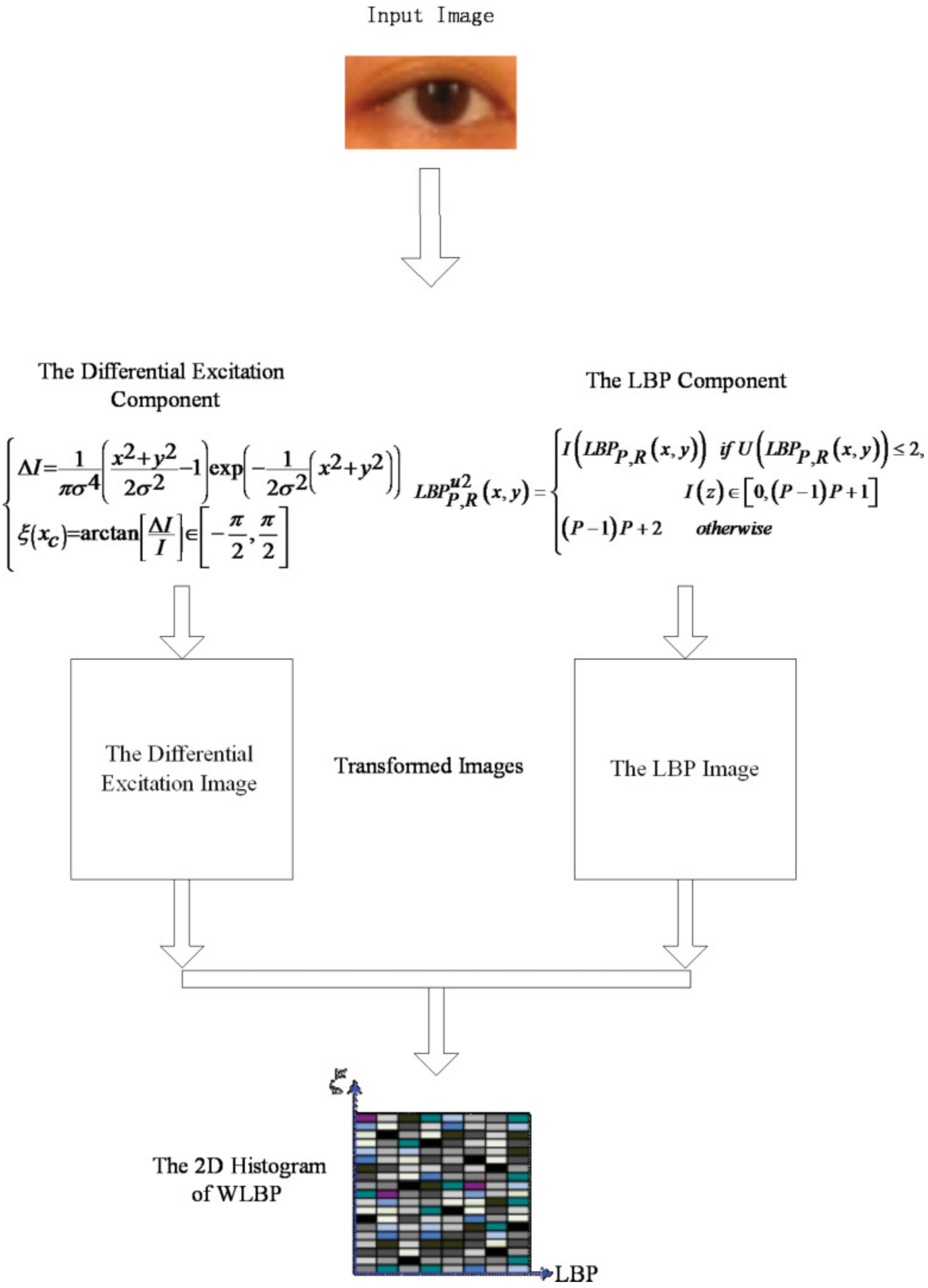
Input Image
The Differential Excitation Component
$\Delta I=\frac{1}{\pi\sigma^4}\left(\frac{x^2+y^2}{2\sigma^2}-1\right)\exp\left(-\frac{1}{2\sigma^2}\left(x^2+y^2\right)\right)$
$\xi(x_c)=\arctan\left[\frac{\Delta I}{I}\right]\in\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$
The LBP Component
$LBP^{u2}_{P,R}(x,y)=\begin{cases} I\left(LBP_{P,R}(x,y)\right) & if\ U\left(LBP_{P,R}(x,y)\right)\le 2, \\ & I(z)\in\left[0,(P-1)P+1\right] \\ (P-1)P+2 & otherwise \end{cases}$
The Differential Excitation Image
Transformed Images
The LBP Image
The 2D Histogram of WLBP
ξ
LBP

Fig. 4. Illustration of the computation of the WLBP descriptor.

excitation interval. Concatenating the S subhistograms, we obtain the 1D histogram $H = \{H_s\}, s = 1, \ldots, S$. For the eye region, the WLBP histogram is calculated to form a single feature vector of the eye.

3.4.2. Eye State Classification. There are two default eye states: open and closed. We define the state of the eye to be open if the iris and the sclera of eye can be seen. Otherwise, if the iris and sclera are difficult to distinguish or even not visible, we define the eye to be closed.

Classification of the open and closed state is complex due to many factors, such as the changing shape of the eye, the changing position and the rotating of the face, and variations of twinkling and illumination. All of these factors make it difficult to analyze eye states in a reliable manner. To solve the problems, we leverage the SVM classifier to classify two eye states, as it can achieve high generalization performance without prior knowledge, even when the dimension of the input space is very high [Tzotsos and Argialas 2006].

SVM is a classical binary classification algorithm developed by Vapnik and his team [Vapnik 1999]. By minimizing the structural risk, it finds the optimal linear decision surface between classes. The decision surface is the weighted combination of elements of the training set. These elements are called *support vectors* and characterize the boundary between the two classes. However, there are still many cases that are not linearly separable. For such classification problems, the input vectors should be nonlinearly mapped to a high-dimensional feature space. In this feature space, the two classes are considered to be linearly separable. The separating hyperplane is now defined as a linear function of vectors drawn from the high-dimensional feature space rather than the original input space. However, it is difficult to compute the high-dimensional spaces. The kernel method gives the solution to this problem. Several possible inner product kernels can be applied, such as Gaussian radial basis function (RBF), polynomial functions, and sigmoid polynomials.

In this work, we employed the SVM algorithm implemented in LIBSVM [Chang and Lin 2011], along with an RBF kernel. Therefore, given an image of the eye, we first extract the WLBP feature and then leverage the SVM classifier along with the RBF kernel to determine the state of the eyes.

3.5. Fatigue Detection

Sections 3.4.1 and 3.4.2 describe the techniques for estimating head pose and eyelid movement, which are two important visual behaviors that reflect a person's level of fatigue. The eye-blinking frequency is a good indicator for characterizing eyelid movement, and thus we set a threshold of consecutive eye-closed frames to represent the eye-blinking frequency. If the number of consecutive eye-closed frames is larger than the threshold, the system issues an alarm. With regard to head pose, if the driver's head deviates excessively from its normal orientation for an extended period of time or too frequently, it will be regarded as a symptom of fatigue. If the system detects that the head pose is not frontal, an alarm is also issued to alert the driver of a danger situation.

The fatigue parameters computed from these visual cues can subsequently be combined probabilistically to form a composite fatigue index that can robustly, accurately, and consistently characterize ones' fatigue level. For example, Lan et al. [2002] proposed using Bayesian networks for fatigue information fusion. In addition, Bergasa et al. [2006] proposed using the license-free knowledge base configuration tool (KBCT) Alonso et al. [2004] to implement fuzzy systems. This article focuses on the development of algorithms to extract the visual cues of eye state and head pose. The issue of fuzzy systems will be discussed in future work.

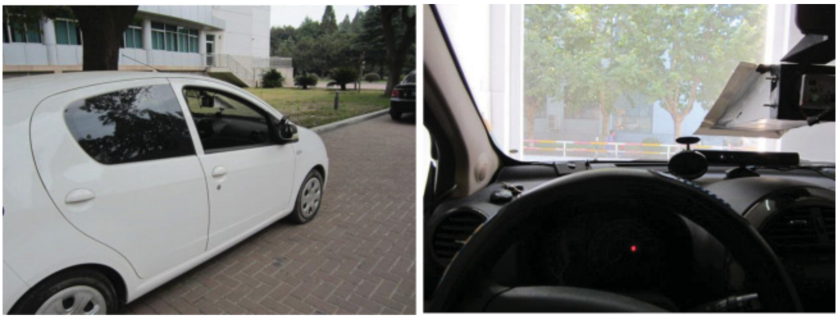

Fig. 5. The experimental environment for driver fatigue monitoring.

4. EXPERIMENTAL ANALYSIS

To evaluate the effectiveness of the proposed driver fatigue detection system, a series of experiments are performed using the inside-car environment. As illustrated in Figure 5, the Kinect camera is mounted on the right front of the dashboard.

We first perform experiments to test the techniques for head pose estimation and then the methods for eye state estimation. Finally, the two types of visual cues are combined to determine the state of driver fatigue.

4.1. Experiment for Head Pose Estimation

Experiments were designed to evaluate the performance of the head pose estimation method. The first experiment is performed in a constrained environment where we assume that the subject rotates his head as smoothly as possible. The captured video consisting of a sequence of RGB and depth data is processed with a core Intel T9300 at 2.0GHz. Since both RGB and depth data are reliable in this environment, we set λ to 1, which means that they have the same importance for pose estimation. The total processing speed is about 20 frames per second. Consequently, our system is fast enough to operate in real-time applications. The user must maintain a frontal pose for initialization before the system has begun. The six DOF of the head motion are recovered and used to manipulate a 3D head model. As shown in Figure 6, the position and orientation of the head model is updated whenever the subject moves his head.

We also conduct experiments where only one type of constraint (the depth constraint or optical flow constraint) is leveraged individually to estimate the head pose. The experimental results demonstrate that each has strong and weak points. First, depth constraint is more robust than optical flow constraint because depth data is less sensitive to illumination variation and shading effects. However, the valid sensing distance of the depth sensor in the Kinect is limited, only ranging from 0.6 to 6 meters, whereas the RGB image data does not have this limitation. Besides, head pose estimation based on the depth constraint is also influenced by self-occlusion (e.g., the neck by the chin), as it creates large depth gradients that do not remain consistent with head pose estimation during motion. Therefore, the combination of RGB image data and depth data is necessary and can achieve better performance.

Furthermore, we also test our method in the real-time system for monitoring a driver's fatigue. The experimental environment is shown in Figure 5. We assume that the normal face orientation while driving is frontal. We have experimentally observed that when nodding takes place, the driver's head goes down, flowing a rise of the pitch or roll angle (α, β). Figure 7 shows four different nodding states detected by the head pose. If the driver's head deviates excessively from its normal orientation for an extended period of time or too frequently, it will be regarded as a symptom of fatigue.

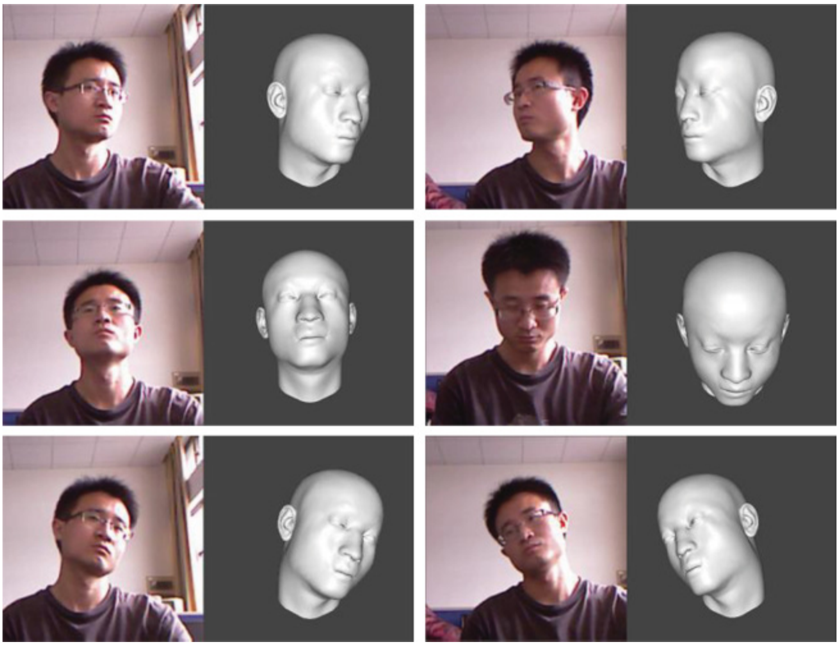

Fig. 6. The three rows show the rotation in yaw, pitch, and roll, respectively.

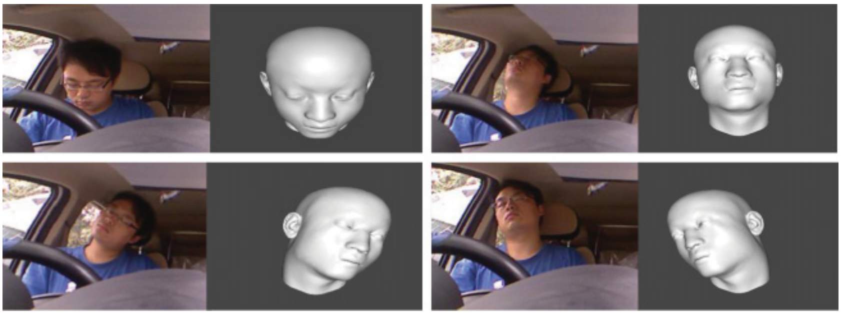

Fig. 7. Four different nodding states and estimated head pose results.

In our experiment, we set two thresholds to determine whether the driver is nodding. If $|\alpha| \geq 10^{\circ}$ or $|\beta| \geq 10^{\circ}$, the system will determine that the driver is nodding and issue an alarm. These thresholds can be selected by leveraging cross validation. However, the performance of head pose estimation will drop as time continues due to error accumulation. This is because in our method, the least-squares solution is leveraged to recover the head motion between two successive frames, about which estimation errors are usually inevitable. When computing the cumulative 3D head motion over time from the starting status, all rotation angles between every two successive image frames will be summed up, and the errors will be accumulated correspondingly.

4.2. Experiment for Eye State Detection

To evaluate the performance of the eye state detection method, we collected 7,686 eye images including 3,546 open eye images and 4,140 closed eye images. These eye images were collected from the RPI ISL Eye Training Database [Wang and Ji 2005], consisting

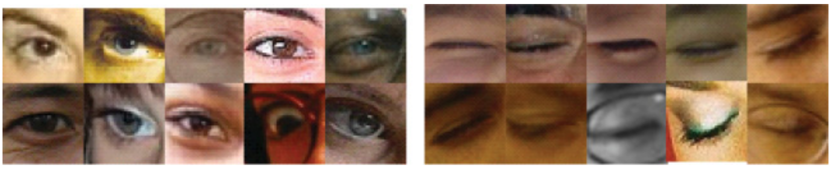

Fig. 8. Eye samples.

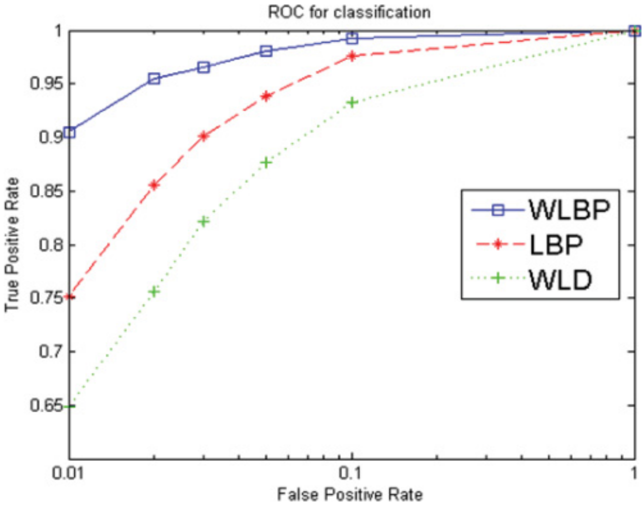

Fig. 9. The ROC curves of the classifiers using features WLBP, LBP, and WLD, respectively.

Table I. Recognition Rate of Eye States

	Recognition Rate			
	SNR = 0	SNR = 5	SNR = 10	SNR = 15
WLBP	96.63%	92.60%	88.41%	81.71%
LBP	94.50%	89.38%	85.48%	78.59%
WLD	91.99%	87.38%	84.05%	74.38%

of many types of eye appearances (different skin colors, changing illuminations, various orientations, with and without glasses, etc.) as illustrated in Figure 8. All of these images were normalized to the size of 40 × 20. In the experiments, we selected 1,576 open eye images and 1,840 closed eye images as training data, and the remaining images as test data.

We compared the performance of the proposed WLBP feature with two conventional features: LBP and WLD. Histograms about the three types of features were extracted from the eye images, respectively, with the length of feature vector being 177 (WLBP), 59 (LBP), and 48 (WLD). The ROC curves of the trained classifiers based on the three features are compared in Figure 9. The comparison demonstrates that the proposed WLBP feature dramatically outperforms the other two features, LBP and WLD. Although the dimension of the WLBP feature is relatively higher than LBP and WLD, it is much smaller than many improved features, such as the 68,853 dimensions of feature vector proposed in Xu et al. [2008].

To demonstrate the robustness of our method to noise, we conducted some experiments on the eye database with Gaussian noise, which are illustrated in Table I. Notice that WLBP outperforms LBP and WLD for every noise level. Besides, WLBP still gets

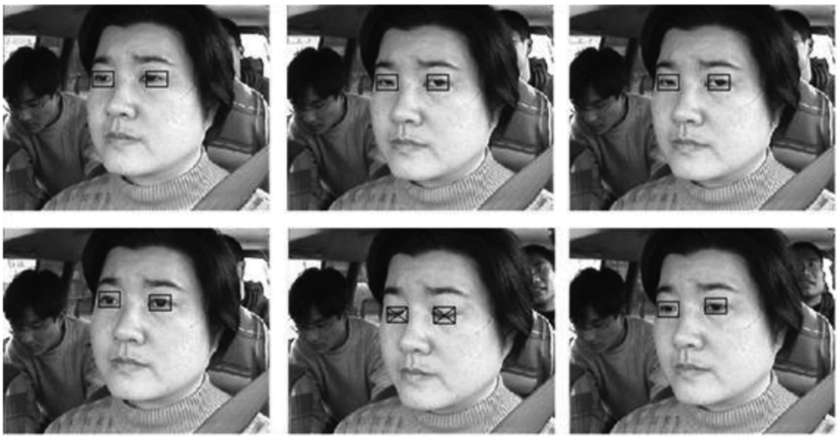

Fig. 10. Some detection results of inside-car video sequences.

the best recognition rate of 81.71% even with the highest noise level. In addition, WLBP is able to achieve high performance despite the large variations of skin color, illumination, face pose, and subjects in the testing dataset. Therefore, the results demonstrate the priority of WLBP in robustness. This is because WLBP has been developed to suppress the effects of illumination variations. In the LBP component, neighbor pixels and the current pixel are compared. Thus, a brightness change in which a constant is added to each image pixel will not affect the pattern values. On the other hand, in the differential excitation component, it performs the division between the differences $\triangle I$ and the current pixel x_c. Thus, a change in image contrast in which each pixel value is multiplied by a constant will be canceled by the division.

4.3. Experiment for Driver Fatigue Detection

In this section, we review experiments using the inside-car environment to evaluate the whole driver fatigue detection system. We collected the real-world driving videos and selected six sequences with 6,000 frames as the experimental dataset. Each frame was processed to be a gray-level image with a size of 320 × 240 pixels. We manually labeled the ground truth where 30 periods are labeled as the fatigue state. The experiment demonstrates that our system is able to detect 28 fatigue periods correctly. Figure 10 illustrates the detection results of six frames of the video sequences. From the experiments, we discover that in most cases, the fatigue samples can be detected by eye states. However, in some conditions, the detection of eyes might fail when drivers appear with different poses or expressions, such as the lateral nodding pose. These occasions can be solved by head pose estimation. Therefore, the combination of head pose and eye state yields more robust and accurate results.

5. CONCLUSIONS

In this article, we have proposed a real-time system for driver fatigue detection using RGB-D cameras. Considering the limitation of conventional color video cameras and NIR cameras, we utilized the emerging RGB-D cameras as a new exploration in driver fatigue detection. It consisted of two important components: head pose estimation and eye state detection. We first proposed a real-time 3D head pose estimation method leveraging both RGB and depth data. The combination of RGB and depth data improved the reliability and robustness of performance. We then presented a novel feature descriptor, WLBP, to represent eye images to detect eye state, leveraging SVM as the classifier. Experiments demonstrated the superiority of WLBP over the traditional features of WLD and LBP in both effectiveness and robustness to noises and illumination variations. Finally, the two visual cues were integrated systematically to generate the conclusion. The experiments using the inside-car environment were conducted to demonstrate that the combination of head pose and eye state can improve the effectiveness and robustness of the system.

REFERENCES

Kircher Albert, Marcus Uddman, and Jesper Sandin. 2002. Vehicle control and drowsiness. Retrieved February 16, 2015, from http://www.vti.se/en/publications/pdf/vehicle-control-and-drowsiness.pdf.

Jose M. Alonso, Luis Magdalena, and Serge Guillaume. 2004. KBCT: A knowledge extraction and representation tool for fuzzy logic based systems. In *Proceedings of the Conference on Fuzzy Systems*. 989–994.

Anon. 1999. *Perclos and Eyetracking: Challenge and Opportunity*. Technical Report. Applied Science Laboratories, Bedford, MA.

Luis Miguel Bergasa, Jesús Nuevo, Miguel Ángel Sotelo, Rafael Barea, and María Elena López Guillén. 2006. Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems* 7, 1, 63–77.

Christoph Bregler and Jitendra Malik. 1998. Tracking people with twists and exponential maps. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. 8–15.

Michael D. Breitenstein, Daniel Küttel, Thibaut Weise, Luc J. Van Gool, and Hanspeter Pfister. 2008. Real-time face pose estimation from single range images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Qin Cai, David Gallup, Cha Zhang, and Zhengyou Zhang. 2010. 3D deformable face tracking with a commodity depth camera. In *Proceedings of the European Conference on Computer Vision: Part III*. 229–242.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3, 27.

Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, Matti Pietikäinen, Xilin Chen, and Wen Gao. 2010. WLD: A robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9, 1705–1720.

Jyh-Yuan Deng and Feipei Lai. 1997. Region-based template deformation and masking for eye-feature extraction and description. *Pattern Recognition* 30, 3, 403–419.

Gabriele Fanelli, Juergen Gall, and Luc J. Van Gool. 2011. Real time head pose estimation with random regression forests. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. 617–624.

Jennifer Healey and Rosalind W. Picard. 2000. SmartCar: Detecting driver stress. In *Proceedings of the Conference on Pattern Recognition*. 4218–4221.

Ueno Hiroshi, Masayuki Kaneda, and Masataka Tsukino. 1994. Development of drowsiness detection system. In *Proceedings of the Vehicle Navigation and Information Systems Conference*.

Berthold K. P. Horn and Brian G. Schunck. 1981. Determining optical flow. *Artificial Intelligence* 17, 1–3, 185–203.

Qiang Ji and Xiaojie Yang. 2002. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging* 8, 5, 357–377.

Farid Abedan Kondori, Shahrouz Youse, Haibo Li, Samuel Sonning, and Sabina Sonning. 2011. 3D head pose estimation using the Kinect. In *Proceedings of the Conference on Wireless Communications and Signal Processing*. 1–4.

Peilin Lan, Qiang Ji, and Carl G. Looney. 2002. Information fusion with Bayesian networks for monitoring human fatigue. In *Proceedings of the Conference on Information Fusion*. 535–542.

Fan Liu, Zhenmin Tang, and Jinhui Tang. 2013a. WLBP: Weber local binary pattern for local image description. *Neurocomputing* 120, 325–335.

Wu Liu, Yongdong Zhang, Sheng Tang, Jinhui Tang, Richang Hong, and Jintao Li. 2013b. Accurate estimation of human body orientation from RGB-D sensors. *IEEE Transactions on Cybernetics* 43, 5, 1442–1452.

Hieu Tat Nguyen and Arnold W. M. Smeulders. 2006. Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision* 69, 3, 277–293.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 7, 971–987.

Edgar Seemann, Kai Nickel, and Rainer Stiefelhagen. 2004. Head pose estimation using stereo vision for human-robot interaction. In *Proceedings of the Conference on Automatic Face and Gesture Recognition*. 626–631.

Rajinda Senaratne, David Hardy, Bill Vanderaa, and Saman K. Halgamuge. 2007. Driver fatigue detection by fusing multiple cues. In *Advances in Neural Networks—ISNN 2007*. Lecture Notes in Computer Science, Vol. 4492. Springer, 801–809.

Paul Smith, Mubarak Shah, and Niels da Vitoria Lobo. 2003. Determining driver visual attention with one camera. *IEEE Transactions on Intelligent Transportation Systems* 4, 4, 205–218.

Sheng Tang, Yan-Tao Zheng, Yu Wang, and Tat-Seng Chua. 2012. Sparse ensemble learning for concept detection. *IEEE Transactions on Multimedia* 14, 1, 43–54.

Yingli Tian, Takeo Kanade, and Jeffrey F. Cohn. 2000. Eye-state action unit detection by Gabor wavelets. In *Advances in Multimodal Interfaces—ICMI 2000*. Lecture Notes in Computer Science, Vol. 1948. Springer, 143–150.

Angelos Tzotsos and Demetre Argialas. 2006. A support vector machine approach for object based image analysis. In *Proceedings of the Conference on Object-Based Image Analysis*.

Vladimir Vapnik. 1999. *The Nature of Statistical Learning Theory*. Springer.

Peng Wang and Qiang Ji. 2005. Learn discriminant features for multi-view face and eye detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition.* 373–379.

Qiong Wang and Jingyu Yang. 2006. Eye location and eye state detection in facial images with unconstrained background. *Journal of Information and Computing Science* 1, 5, 284–289.

Yu-Shan Wu, Ting-Wei Lee, Quen-Zong Wu, and Heng-Sung Liu. 2010. An eye state recognition method for drowsiness detection. In *Proceedings of the Vehicular Technology Conference (Spring)*. 1–5.

Cui Xu, Ying Zheng, and Zengfu Wang. 2008. Efficient eye states detection in real-time for drowsy driving monitoring system. In *Proceedings of the Conference on Information and Automation*. 170–174.

Liyan Zhang, Dmitri V. Kalashnikov, and Sharad Mehrotra. 2013a. A unified framework for context assisted face clustering. In *Proceedings of the Conference on Multimedia Retrieval.* 9–16.

Liyan Zhang, Dmitri V. Kalashnikov, Sharad Mehrotra, and Ronen Vaisenberg. 2013b. Context-based person identification framework for smart video surveillance. *Machine Vision and Applications* 25, 7, 1711–1725.

Tao Zhao and Ramakant Nevatia. 2004. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 9, 1208–1221.

Received June 2013; revised December 2013; accepted March 2014