Automatic Calibration Method for Driver's Head Orientation in Natural Driving Environment

Xianping Fu, Xiao Guan, Eli Peli, Hongbo Liu, and Gang Luo

***Abstract*—Gaze tracking is crucial for studying driver's attention, detecting fatigue, and improving driver assistance systems, but it is difficult in natural driving environments due to nonuniform and highly variable illumination and large head movements. Traditional calibrations that require subjects to follow calibrators are very cumbersome to be implemented in daily driving situations. A new automatic calibration method, based on a single camera for determining the head orientation and which utilizes the side mirrors, the rear-view mirror, the instrument board, and different zones in the windshield as calibration points, is presented in this paper. Supported by a self-learning algorithm, the system tracks the head and categorizes the head pose in 12 gaze zones based on facial features. The particle filter is used to estimate the head pose to obtain an accurate gaze zone by updating the calibration parameters. Experimental results show that, after several hours of driving, the automatic calibration method without driver's corporation can achieve the same accuracy as a manual calibration method. The mean error of estimated eye gazes was less than 5° in day and night driving.**

***Index Terms*—Calibration, gaze tracking, head orientation.**

I. Introduction

GAZE tracking is crucial for studying driver's attention, detecting fatigue, and improving driver assistance systems. Video-based methods are commonly used in gaze tracking but are vulnerable to the illumination changes between day and night. Eye-gaze tracking methods using corneal reflection with infrared illumination have been primarily used indoor [1]–[5] but are highly affected by sunlight. Recently, video-based eye-gaze tracking methods have been used in natural driving environments [6], [7]. This paper presents an automatic tracking system for head-pose and eye-gaze estimations in natural driving conditions. To achieve this goal, we have developed a novel learning algorithm combined with a particle filter. This framework differs from previous methods, to a great extent, in its ability to estimate a driver's gaze zone automatically, which minimizes the need for driver compliance.

The two main contributions of this paper are in the configuration of hardware and designs of algorithms. The first contribution is the new learning algorithm that allows for the self-classifications of the different head poses and eye gazes. This new algorithm is motivated in part by the work of Toyama and Blake [8]. We define a set of head poses as a metric space and assign those head poses into corresponding gaze zones. Unlike other methods, it does not need to be trained before system deployment because the classification process can be completed and tuned during driving. The self-learning method is possible based on two reasonable assumptions: 1) when a driver is seated, the driver's head position relative to the side rear mirrors, the rear-view mirror, the windshield, etc., does not vary greatly and, 2) most drivers have habitual and consistent ways of moving their head and eyes when looking in a specific direction. The learning algorithm is crucial for the system's ability to calibrate automatically [9].

Our second contribution is the method of combining face detection, a learning algorithm, and particle filtering in a cycling structure that enables the tracking system to run automatically. These algorithms are put in a proper logical order so that they can call each other without manual intervention.

Satisfactory accuracy in head-pose and eye location estimations has been achieved in constrained settings in previous studies [10], [11]. However, in the absence of frontal faces that is common in driving, eye locators cannot adequately locate the pupil. While precise gaze direction provides useful information, coarse gaze direction is often sufficient, for instance, for determining whether a driver's attention is off the road ahead since most natural eye movements are less than 15° and a person usually starts moving the head to a comfortable position before orienting the eye [12]. Points of interest are grossly delineated by the head pose. In a meeting situation, for instance, the head pose was shown to contribute about 70% of the gaze direction [13]. Therefore, the head pose is important for the estimation of coarse gaze direction, which is called the gaze zone in this paper. The gaze zone was estimated based on facial features, face location, and face size using a self-learning method and particle filtering. The system was shown to operate in day and night conditions and is robust to facial image variation caused by eyeglasses as it does not solely rely on facial feature points,

Manuscript received March 8, 2012; revised June 5, 2012; accepted July 21, 2012. This work was supported in part by the National Natural Science Foundation of China under Grant 61272368, Grant 60873054, Grant 61073056, and Grant 61173035, by the Fundamental Research Funds for the Central Universities under Grant 2011QN031, by the Program for New Century Excellent Talents in University, by the Liaoning Education Department Research Fund under Grant L2010 061, by the Dalian Science and Technology Fund under Grant 2010J21DW006, and by the Prevent Blindness International Research Scholar Award 2010. The work of G. Luo was supported by the National Institutes of Health (NIH) under Grant AG041974. The work of E. Peli was supported by the NIH under Grant EY05957. The Associate Editor for this paper was S. S. Nedevschi.

X. Fu was with the Schepens Eye Research Institute, Harvard Medical School, Boston, MA 02114 USA. He is currently with the Information Science and Technology College, Dalian Maritime University, Dalian 116026, China (e-mail: fxp@dlmu.edu.cn).

X. Guan is with Tulane University, New Orleans, LA 70118 USA.

E. Peli and G. Luo are with the Schepens Eye Research Institute, Massachusetts Eye and Ear, Harvard Medical School, Boston, MA 02114 USA.

H. Liu is with the Information Science and Technology College, Dalian Maritime University, Dalian 116026, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2217377

such as eyes and lip corners, for gaze estimation. Because of its low computation, it can work in real time on a laptop computer.

II. Related Work

There are driver gaze tracking methods that consider only head orientations [7]. The size, shape, and distance of facial features and the distance between these features, such as the distance between the left and right pupils, are used to estimate a driver's head orientation [14]. The head-pose estimation often requires multiple cameras or complex face models that require accurate and lengthy initialization [11]. Although several head-pose or eye location methods have shown success in gaze estimation, the underlying assumption of being able to estimate gazes based on eye location or the head pose is only valid in a limited number of scenarios [15]–[18].

Smith *et al.* analyzed color and intensity statistics to find both eyes, lip corners, and the bounding box of the face [19]. By using these facial features, they estimated continuous head orientation and gaze direction. However, this method cannot always find facial features when the driver wears eyeglasses or makes conversation. Kaminski *et al.* analyzed the intensity, shape, and size properties to detect the pupils, nose bottom, and pupil glints to estimate continuous head orientations and gaze direction [20]. By using the foregoing methods considering both eye and head orientations, detailed and local gaze direction can be estimated. However, the accuracy of the eye location significantly drops in the presence of large head movements. This is because, in some cases, the eye structures are not symmetric; thus, the algorithm delivers poorer performance with respect to the distance from the frontal pose.

Because errors in facial feature detection greatly affect gaze estimation [12], many researchers measured coarse gaze direction by using only the head orientation with an assumption that the coarse gaze direction can be approximated by the head orientation [7]. The methods that only consider head orientation can be categorized into methods based on shape features with the eye position, methods based on shape features without the eye position, methods based on texture features, and methods based on hybrid (shape and texture) features.

Methods based on shape features with the eye position analyze the geometric configuration of facial features to estimate the head orientation [14]. These methods rely on precise localization of facial features, which is prone to error when illumination varies and eyeglasses are used in practice. The methods based on texture features find the driver's face in the image and analyze the intensity pattern of the driver's facial image to estimate the head orientation. Learning techniques such as principal component analysis (PCA), kernel PCA (KPCA), linear discriminant analysis (LDA), and kernel discriminate analysis have been used to extract texture features, and these features are then classified to obtain the discrete head orientation [21]. Ma *et al.* analyzed the asymmetry of the facial image by using a Fourier transform to estimate the driver's continuous yaw [22]. The methods based on texture features are relatively reliable because specific facial features do not need to be localized. However, accuracy can degrade when the face detection module cannot give a consistent result [23]. Wu *et al.* detected the driver's discrete yaw and pitch by using a coarse-to-fine strategy [24]. This method is based on hybrid features combining shape and texture features to estimate the head orientation. Murphy-Chutorian *et al.* estimated the initial head orientation by using a local-gradient-orientation-based (LGO) head orientation method [25], and detailed head orientation was computed by using a 3-D face model for fitting and tracking [26], [27]. This method showed excellent performance but required accurate initialization. A general drawback of the hybrid methods is the relatively high computational complexity caused by combining two feature extraction methods [24].

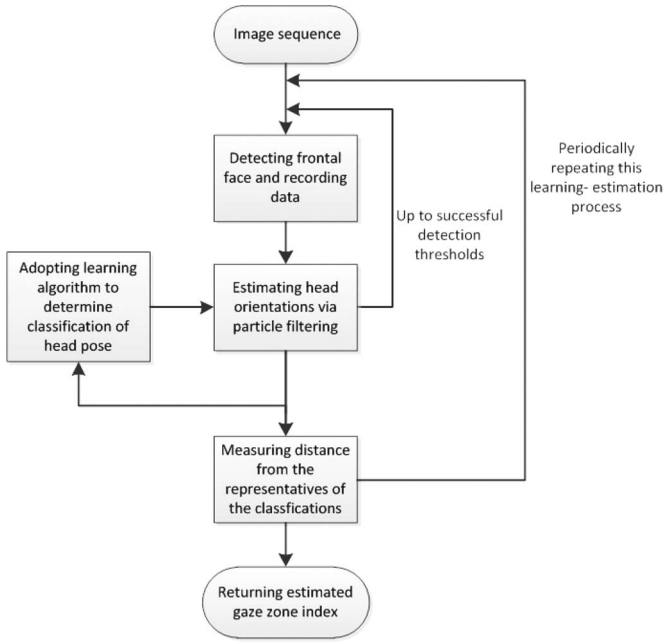


Fig. 1. Framework of the proposed method.

III. Proposed Gaze-Zone Estimation Method

A. Overview of Proposed Method

This paper presents a combination of face detection [28], self-learning, and particle filtering to build a system for driver gaze-zone estimation. The three modules are used for detecting driver's frontal face, determining the classification of driver's head orientation, and estimating the current driver's head pose and moving velocity, respectively. Once face detection is completed, the frontal face of the driver is buffered, and the particle filtering module calculates the current head pose based on the previous state and feature selected. Simultaneously, the head-pose data are fed into the learning module for updating the classification of head orientations relative to gaze zones. The returned results of the particle filtering is utilized to estimate current driver's gaze zone based on the distance from the representatives of the gaze-zone index set and the selected gaze-zone index. The driver's head orientation and position information are used to determine the gaze zone. The logical flowchart of the three algorithms is depicted in Fig. 1.

In the face detection step, the face region is located within the driver's entire head image to remove the unnecessary background, and the region of the head is used in the following

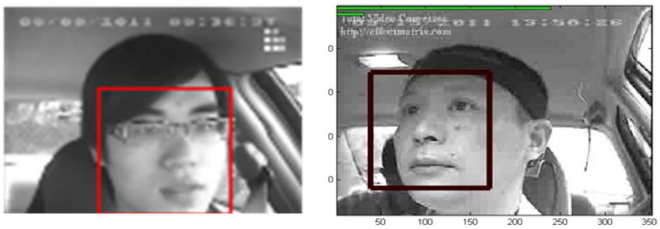


Fig. 2. Face detection with a bounding box.

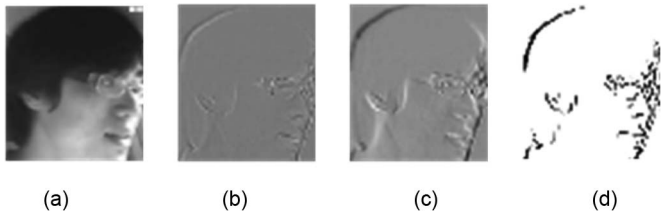


Fig. 3. Transformed image-depth image. (a) Mean face when looking at the left side mirror. (b) First eigenface of (a). (c) Second eigenface of (a). (d) Binary edge image of (c).

steps. In addition, the confidence probability is used to represent the accuracy of the head position estimation.

To calculate the extremely large head rotation angle, the head shape and facial features are combined to offer more raw information for the head position. Facial features are used when the frontal face is available. When no frontal face is detected, an appearance-based method is used to track the head position. The head pose provides a coarse indication of gaze that can be estimated in situations when eyes are not visible (such as low-resolution imagery, large head rotation angle, and in the presence of eye-occluding objects such as sunglasses).

B. Frontal Face Detection

It is necessary to detect the driver's frontal face first since the learning algorithm cannot perform correctly if the center coordinates of the driver's frontal face are not known. We used the frontal face detection method introduced in [28]. The main advantages of this method are threefold. The first advantage is its integral image representation, which is the sum of the pixels values above and to the left of a specific location in a 2-D plane. This is useful for calculating Haar-like features rapidly at any scale. The second advantage is a classifier built upon the AdaBoost learning algorithm by selecting a few critical features from a huge number of features extracted with the help of the integral image. During learning rounds, the examples are reweighted to put emphasis on those incorrectly classified by the previous weak classifiers that depend on a single feature. Finally, it uses a method where the more complex classifiers are arranged in a cascade structure that drastically reduces the detecting time by removing unpromising regions. To represent the detecting area, a bounding box is set to enclose the whole face of the driver, as shown in Fig. 2. After the face region is found, the background within the box is removed, and the region of the frontal face is configured for facial feature detection that is later used as the sample points in the particle filtering. Initially, the face detection is executed repeatedly within a short period of time. Afterward, the center of the bounding boxes of the face is used to guide the detection by increasing the weight of those locations. The face detection runs at a regular interval and modifies the mean center coordinates periodically.

As indicated in Fig. 2, simple facial features, such as the center of the driver's face and the left and right borders within the bounding box, can be extracted using the method of Ohue *et al.* [29]. Once these features are determined, a proper transformation is implemented on the specific area of the raw images in terms of image patches. Generally, one Gaussian and several rotation-invariant Gabor wavelets are applied on the

Fig. 4. Successful facial feature detection [28] used to enhance the accuracy of gaze estimation is shown as circles marking the lower lip, the nostrils, and the irises.

training patch images extracted from head images. A simple Gabor wavelet is defined by

$$\psi_{\omega_0,\sigma,\alpha}(x,y) = \exp\left[-\frac{1}{2\sigma^2}(\tilde{x}^2+\tilde{y}^2)\right]\cos\left(2\pi\omega_0\tilde{x}\right) \quad (1)$$

where $\tilde{x} = x\cos(\alpha)y\sin(\alpha)$ and $\tilde{y} = x\sin(\alpha)+y\cos(\alpha)$. ω_0 denotes the angular frequency, σ is the scale parameter, and α is the orientation of the wavelet. Thus, we obtain the rotation-invariant wavelet by integrating a simple wavelet over the rotation field of α. The distance function on the space $\mathcal{A}$ is the L^2 distance function.

Training patches are extracted from head images by locating a tight bounding box enclosing the head. These patch images are resized to 64×64 resolution and preprocessed by histogram equalization to reduce the effect of lighting variations. The eigenface and eigenhead images are calculated using the PCA method. Thereafter, Gabor wavelets are applied on these eigenimages, as defined by (1). The rotation-invariant Gabor wavelets that we used are defined by the scales $\sigma = 1, 2, 4$ and angular frequencies $\omega_0 = (1/2), (1/4), (1/8)$. The resulting images are sampled at 191 points of a regular grid located inside a reference disk. The transformations are shown in Fig. 3.

In addition, the face detection method (available in OpenCV) in [28] is also used as a supplement when the detection is successful, to enhance the accuracy of the eye-gaze estimation. An example of detected facial features using [28] is shown in Fig. 4. When no frontal face is detected, however, the appearance-based method can be used to track the head position even when eyes are not visible due to, for example, large head rotation and eye-occluding sunglasses (see Fig. 3).

The proposed method does not use any head or face model since gaze-zone position are trained base on the detected features. This method has two advantages over other shape-feature-based methods. There is no need to train a specific

model for each driver, and it is relatively robust to the wearing of eye glasses or when the head movement is large.

C. State Model

The classification algorithm is based on the true metric space. Given metric space X, the distance function d defined on this space satisfies the following three properties: 1) $d(x, y) \geq 0$ for all pairs x, $y \in X$, and $d(x, y) = 0$ if $a = b$; 2) $d(x, y) = d(y, x)$; and 3) $d(x, y) + d(y, z) \geq d(x, z)$

In this paper, the driver's head is modeled as a rigid object constrained to 4° of freedom in the video image plane. We define the state $X_t = (A_t, v_t)$, where $A_t = (T_x, T_y, \alpha_t, \beta_t)$ is a 4-D vector consisting of two translations and two rotations (i.e., yaw and pitch), and v_t represents the linear and angular velocities. When looking forward in driving, the driver's head motion is normally slow. Only when the driver moves his head can linear dynamics provide a good temporary approximation of the motion. Thus, we use the mixed state X_t^*, as given in [26], with $X_t = (1 - \tau_t)X_t^{(1)} + \tau_t X_t^{(2)}$ to represent this situation, and it is defined as follows:

$$X_t^* = (X_t, \tau_t) \tag{2}$$

where τ_t is a binary random variable with $\tau_t \in \{0, 1\}$, $X_t^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X_{t-1}^{(1)} + \binom{u_t^{(1)}}{0}$ denotes the state at time t without velocity, and $X_t^{(2)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} X_{t-1}^{(2)} + \binom{0}{u_t^{(2)}}$ denotes the constant velocity. The distribution of τ_t depends on the motion of the object. Roughly speaking, $\tau_t = 1$ while

$$d(Y_t, Y_{t-1}) > \varepsilon_1 \text{ or } d(A_t, A_{t-1}) > \varepsilon_2$$

where ε_1 and ε_2 are sufficiently small positive numbers; otherwise, $\tau_t = 0$. In addition, $Y_t = T_\epsilon X_t + V_t$ is a realization of X_t provided that T_ϵ is a transformation between the two image spaces and that V_t is the noise of time point t [8]. In addition, the transition density $f_t(X_t|X_{t-1})$ can be determined by this state model along with the observation density given by $g_t(Y_t|X_t) \propto (1/B)\exp(-\lambda d(Y_t, T_\epsilon X_t + V_t))$, where B and λ are normalizing constants. Utilizing the method of particle filters, the current state X_t, which is estimated by a mass of sample points assigned with weights, is given by $\widehat{X_t} = \sum_{n=1}^{N} w_t^{(n)} \tilde{x}_t^{(n)}$, where $\tilde{x}_t^{(n)}$ is the nth sample point of the current state.

D. Classification of Head Orientations

Our proposed method includes a set of pose-annotated key frames obtained by using a single camera and a learning algorithm similar to that in [8]. The information that each key frame maintained is described as follows:

$$M_j = \{I_j, Z_j, X_j\}. \tag{3}$$

Provided that X_j is the image set of frame j, and I_j and Z_j are the intensity and depth images with this image set, respectively. Adopting the learning algorithm, the head-pose key frames are defined by the following set:

$$\mathcal{M} = \{M_j | 1 \leq j \leq k\}$$

where k is the number of key frames and is determined by the number of configured gaze zones. We initialize representative states for each gaze zone by splitting the video image rectangular into numbers of sectors, which amount to the number of gaze zones and each of whose center corresponds to the gaze-zone index, based on coordinate's representations transmitted by the head sensor. To achieve the best result, each image element in the incoming frame should be processed, and these elements then are reorganized to its corresponding gaze-zone index set. Once each such set is well defined, the new frame (current) can be approximated by measuring the distance from each representative of the gaze-zone index set based on the conditional probability distribution $p(M_j|M_t)$, and then a basic frame is chosen to substitute the current frame based on this probability distribution so that we can directly determine which zone the driver is looking at. To make the computation more efficient, the cardinality of gaze-zone index set $\#M$ needs to be properly small.

The exemplar set needs to be determined for each driver. Here, $\mathcal{A}^* = \{A_i^*, i = 1, 2, \ldots k\}$ denotes the final learned result for each gaze zone as representations for basic frames. In this paper, $k = 12$ as we divide the field of view into 12 different gaze zones, and $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ is the image sequence. When the exemplars are determined from the learning algorithm, the learned image set is expressed as $\mathcal{Y}^* = \{Y_0^*, Y_1^*, \ldots, Y_k^*\}$. The learning process is applied at a regular time interval so that each gaze zone is associated with standard head poses. In addition, distance functions are defined on the image space and transformed image space.

We constructed a standard data of yaw and pitch based on an inertial head position sensor, which is expressed by $\mathcal{A}_0^* = \{A_{10}^*, A_{20}^*, \ldots, A_{k0}^*\}$. The average distance vector of yaw and pitch between adjacent gaze zones is computed by

$$(\alpha_{\text{adt}}, \beta_{\text{adt}}) = \frac{1}{m} \sum_{(i,j)(\text{ad})} \left| (T_{y,p}^* A_{i0}^* - T_{y,p}^* A_{j0}^*) \right|$$

provided that m counts the adjacent gaze-zone pairs, and $\sum_{(i,j)(\text{ad})} |(T_{y,p}^* A_{i0}^* - T_{y,p}^* A_{j0}^*)|$ means that only the absolute difference of adjacent states is counted $(\text{ad} = \text{adjacent})$. In addition, the notation $|\cdot|$ means that each component of vector $T_{y,p}^* A_{i0}^* - T_{y,p}^* A_{j0}^*$ is assigned by its absolute value so that the average vector is finally returned.

The driver's gaze zone is analyzed in terms of a probability derived from the distance function that measures the yaw and pitch between the given status and each element of a trained exemplar set (basic frame) $\mathcal{A}^*$ or $\mathcal{Y}^*$. In what follows, we exploit the mixture centers represented by the estimated yaw and pitch for each gaze zone.

The acquisition of A_t for each image is based on the particle filtering algorithm, which will be described in the following. To improve the initial alignment, the first face image is chosen such that

$$Y_0 = Y_0^* = \arg\min_{Y \in \mathcal{Y}} \max_{Y' \in \mathcal{Y} - \{Y\}} d(Y, Y').$$

The corresponding state A_0 (first component of X_0) is determined, which may be different from the zero point of yaw and pitch.

The centers for gaze zones are formed in such a way in which the sequence of A_i, associated with image sequence Y_i, is selected to be approximately identical spaced in distance. The base lattice is generated as each point shares the average yaw deviation $\rho_y = \max d_y(A_i, A_j)/y$ and average pitch deviation $\rho_p = \max d_p(A_i, A_j)/p$ measured from its adjacent point, where d_y is the yaw distance, d_p is the pitch distance, and $A_i,\ A_j \in \mathcal{A}$. Integers y and p are chosen based on the distribution of gaze zones. Thereafter, the ratio coefficient vector is figured out by the formula $(c_y, c_p) = (\alpha_{\mathrm{adt}}/\rho_y, \beta_{\mathrm{adt}}/\rho_p)$ between standard data and samples.

Hence, for each j, the element of the training set is put into class j if and only if $\|T_{y\cdot p}A_i - T^*_{y,p}A^*_{j0}\|$ is the minimum after measuring with all elements in $\mathcal{A}^L$, where $\|\cdot\|$ is the norm on space $\{(T_{y\cdot p}A_i)\}$. A quotient set $\bar{\mathcal{A}} = \{\overline{A_J}, j = 1, 2, \ldots, k\}$ is then derived from the classification, and the new representative for each element in $\bar{\mathcal{A}}$ is determined by $A^*_i = \arg\min_{A\in\overline{A_\iota}} \max_{A'\in\overline{A_\iota}\setminus\{A\}} \|T_{y\cdot p}A - T_{y\cdot p}A')\|$. Thus, the final learning result for exemplar set $\mathcal{A}^* = \{A^*_i, i = 1, 2, \ldots, k\}$ is formed. The representing gaze zone of exemplars is denoted by the indicator of exemplar element.

The whole learning algorithm is summarized as follows:

ALGORITHM 1 LEARNING ALGORITHM

(1) Before Initialization

$$\mathcal{A}^*_0 = \{A^*_{10}, A^*_{20}, \ldots, A^*_{k0}\}$$

$$(\alpha_{adt}, \beta_{adt}) = \frac{1}{m}\sum_{i,j\ ad} \left|(T^*_{y,p}A^*_{i0} - T^*_{y,p}A^*_{j0})\right|$$

(2) Face Detection Procedure

Do

if succeed in face detection

return the center point (x_c, y_c) of the bounding box;

record the data in the subsequent place of the specific array;

while $V < sam\_fa$ (V is the number of whole data recorded and sam_fa the threshold)

Compute the weighted mean center of the detected bounding boxes

(3) Initialization

$$Y_0 = Y^*_0 = arg\min_{Y\in\mathcal{Y}} \max_{Y'\in\mathcal{Y}-\{Y\}} d(Y, Y')$$

$$\varphi(A_0) = a_0$$

(4) Computation for extended coefficient

$$\rho_y = \max d_y(A_i,A_j)/y$$

$$\rho_p = \max d_p(A_i,A_j)/p$$

$$(c_y, c_p) = (\alpha_{adt}/\rho_y,\ \beta_{adt}/\rho_p)$$

(5) Exemplar determination

for each $i, j;\ A_i \in \mathcal{A},\ j = 1,2,\ldots.k$

$$\varphi(A_i) = \arg\min_j \left\|T_{y,p}A_i - T^*_{y,p}A^*_{j0}\right\|$$

$$A_i \in \overline{A_{\varphi(A_i)}}$$

(6) Repeat step (4) for remaining training elements.

(7) Find the new representative for each $\overline{A_\iota}$ given that $\overline{A_\iota} = \{A_t: \varphi_t(A_t) = i\}$ for each cluster i

$$A^*_i = \arg\min_{A\in\overline{A_i}} \max_{A'\in\overline{A_i}\setminus\{A\}} \left\|T_{y,p}A - T_{y,p}A')\right\|$$

(8) Repeat steps (5) to (7) until convergence and save the final exemplars $\overline{A_\iota}$.

(9) Repeat step (2) if the norm of velocity $\|v_t\| < \varepsilon$

(10) Repeat steps (3) to (8) after the adoption of step (9)

Notation: $T^*_{y,p}$ here is a linear operator that returns the coordinate of yaw and pitch of state A and $(|\alpha|, |\beta|) = |T^*_{y,p}A|$. $T_{y,p}$ is also a linear operator which is defined by $(\tilde{\alpha}, \tilde{\beta}) = (c_y, c_p)\cdot T^*_{y,p}A$. The two linear operators are bounded and the general proof for this fact can be found in[30]

This learning algorithm runs routinely as all parameters are set up before the system is initialized. In addition, the repeated steps of frontal face detection are also included in the learning structure. Therefore, we can achieve the automatic control over the learning process embedded in the calibrations of head orientations without driver's collaboration toward specific head poses.

E. Particle Filtering for State Estimations

We usually consider only a probability density function denoted as $p(X)$. However, in some specific cases, we use notation $P(X)$ to stand for the associated probability distribution (measure). The mathematical expressions adopted in this algorithm are consistent with those in [31].

Particle filtering is an inference technique that gives an estimation result of an unknown state X_t based on a sequence of noise observations $y_{1:t} \overset{\mathrm{def}}{=} \{y_1, y_2, \ldots, y_t\}$, which is a realization of the sequence, arriving in an incremental time step. In this paper, we use the state model referred in section E of [31] with Markovian assumptions. The sample states are selected according to the transition density function and the observation density function.

The evaluation of the transition and observation densities is a crucial part of the particle filtering algorithm described later as it determines the motion of the object and puts a great emphasis in the simulation step. The adoption of Monte Carlo methods for nonlinear and non-Gaussian filtering dates back to the pioneering work of [32] and [33] with the aid of importance sampling [34]. Its basic idea comes from the fact that when the analytic form of the probability density function $p(X)$ or the probability distribution function $P(X)$ is not available, we must select the samples according to an instrumental distribution $Q(X)$ as a substitute for the true distribution. Finally, the true distribution was approximated by weighing these samples after the resampling step is executed, which is helpful for determining the best expectation values of specific functions with variables as the sample points. The details of this algorithm can be found in [31]. Fig. 5 shows the flowchart.

The basic idea of resampling is to select a number from the set $[N] = \{1, 2, \ldots, N\}$ with $P(X = i) = w^{(i)}_t$, and it is easily comprehended that if we choose N digits from the set $[N]$, the selecting times each digit assigned obey the following probability distribution:

$$P(X_1 = i_1, X_2 = i_2, X_N = i_N) = \binom{N}{i_1 i_2, \ldots, i_N} \left(w^{(1)}_t\right)^{i_1} \left(w^{(2)}_t\right)^{i_2} \ldots \left(w^{(N)}_t\right)^{i_N} \quad (4)$$

where $X_j = i_j$. It indicates that the digit j is selected by i_j times among the N selecting times and that $i_1 + i_2 + \cdots + i_N = N$. In addition, $\binom{N}{i_1 i_2, \ldots, i_N} = (N!/i_1!i_2! \ldots i_N!)$ denotes the multinomial coefficient. Then, a new particle index set is naturally constructed as a multiset $B = \{i_1 \cdot 1, i_2 \cdot 2, \ldots, i_N \cdot N\}$, with $i_1 + i_2 + \cdots + i_N = N$. There exists a one-to-one map σ from $[N]$ to B, and then we can resample the particles as $x^{(i)}_t = \tilde{x}^{(\sigma(i))}_t$ and reset $w^{(i)}_t = (1/N)$. As the unimodality of the multiplicative decomposition on the right side of (4) was proven in the combinatorial probability literature in advance, we can thus achieve the best resampling result by searching

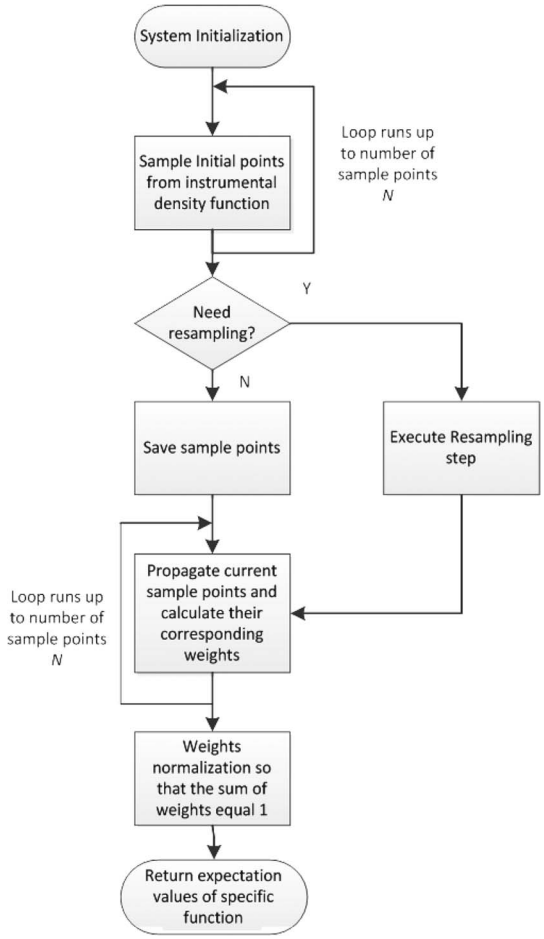


Fig. 5. Flowchart of particle filtering.

proper values of $i_1, i_2, \ldots, i_N$ that attain such peak. We need to pay attention to the fact that the particle $\tilde{w}_t^{(i)}$ may be repeatedly sampled in the resampling step. Therefore, instead of learning all possible calibrations in 2-D space, we propose to automatically re-target a set of known points on a target plane to simulate a recalibration each time the driver moves his head. Using this new set of known points and the known pose-normalized displacement vectors collected during the calibration phase, it is possible to automatically recalibrate and develop a new mapping.

As a result, the indicated gaze zone of the current head pose will be estimated more accurately. Sampling importance and resampling (SIR) (which is a part of the particle filtering method) has made great changes when compared with importance sampling. Our SIR includes the following aspects: 1) the overall algorithm can no longer be viewed as a simple extending of importance sampling because it embeds a repeated application of the importance sampling and resampling; and 2) the resampled paths $x_{0:t}^{(i)}$ are dependent on each other. We summarize this algorithm as follows.

Algorithm 2 Particle Filter

Initialization:

for $i = 1 : N$

Sample $\tilde{x}_0^{(i)} \sim q_0(X_0|Y_0 = y_0)$;

Assign initial importance weights

$$\tilde{w}_0^{(i)} = \frac{g_0\left(y_0|\tilde{x}_0^{(i)}\right) p_0\left(\tilde{x}_0^{(i)}\right)}{q_0\left(\tilde{x}_0^{(i)}|y_0\right)}$$

end

for $t = 1 : T$

if Resampling (condition)

Select N particle indices $k_i \in \{1, 2, \ldots, N\}$ according to the multinomial distribution:

$$\left(N, w_{t-1}^{(1)}, \ldots, w_{t-1}^{(N)}\right)$$

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(k_i)}$, and $w_{t-1}^{(i)} = (1/N)$, $i = 1, \ldots, N$

else

Set $x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(i)}$, $i = 1, \ldots, N$.

end

for $j = 1 : N$

Propagate

$$\tilde{x}_t^{(j)} \sim q_t\left(X_t|x_{t-1}^{(j)}, y_t\right);$$

Calculate weight

$$\tilde{w}_t^{(j)} = \tilde{w}_{t-1}^{(j)} \frac{g_t\left(y_t|\tilde{x}_t^{(j)}\right) f_t\left(\tilde{x}_t^{(j)}|x_{t-1}^{(j)}\right)}{q_t\left(\tilde{x}_t^{(j)}|x_{t-1}^{(j)}, y_t\right)}$$

end

Weight normalization

$$w_t^{(i)} = \tilde{w}_t^{(i)} / \sum_{m=1}^{N} \tilde{w}_t^{(m)}, \quad m = 1, 2, \ldots, N$$

Return $\hat{\mathbb{E}}h_t(X_t) = \sum_{n=1}^{N} w_t^{(n)} h_t(\tilde{x}_t^{(n)})$;

end

Thus, we obtain the estimated $\widehat{A_t}$ at time point t, and its corresponding gaze zone is determined by

$$k = \arg_j \min_{A_j^* \in \mathcal{A}^*} d\left(\widehat{A_t}, A_j^*\right) \tag{5}$$

where $d(\widehat{A_t}, A_j^*)$ is an image-based distance metric. The exemplar set is updated at a regular time cycle interval.

IV. Experimental Result

We use two methods to verify the calculated gaze direction. First, an inertial motion sensor (Colibri made by Trivisio, Germany) was attached on a driver's head to record the 3-D head movement. Second, we recorded testing videos (image resolution: 320 by 240 pixels) in which a driver was instructed to look at given targets. Videos were collected in four driving conditions: daytime, nighttime, wearing glasses, and wearing sunglasses. Video frames from 1 to 500 are used to detect frontal face and determine the center point; frames 501 to 5000 are employed to learn the exemplar element of each gaze zone; the remaining frames are utilized to test the robustness of the system.

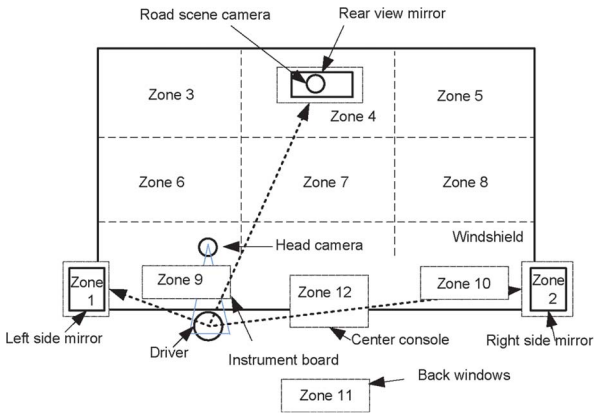


Fig. 6. Gaze zone for calibration and verification in the car.

Fig. 7. Gaze zones denoted by the head pose and the eye gaze of images placed at each zone. The lowest face image represents looking at the back window.

A. System Setup

The gaze-zone estimation system consisted of a monocular camera mounted at the center of the dashboard pointing to the driver. To be used in various vehicle environments, including day and night, the cameras were equipped with an additional infrared illumination source that was controlled by an automatic switch illumination sensor. In daylight, the infrared illuminator is turned off automatically, and visible light is used to track facial features and the head structure. When visible light is too weak to recognize the face, the infrared illumination turns on automatically.

Facial features are extracted after the face region is detected using the method in [28]. As the system does not solely rely on facial features, it can still estimate gaze direction even when the detection of some facial features fails. Because of the self-learning algorithm, the driver's cooperation for this experiment is not needed. Thus, a set of data is chosen randomly to test the estimation accuracy of the proposed method from about 200 000 frames of eight subjects.

B. Gaze-Zone Estimation

Similar to the method in [7], the field of view of a driver is partitioned into 12 different gaze zones corresponding to the two side mirrors, the rear-view mirror, the dashboard, the console, the back window, and six zones on the windshield (see Fig. 6). These gaze zones cover most of the possible gaze directions in real-world driving. The typical head-pose image for these zones is shown in Fig. 7.

TABLE I
GAZE-ZONE REPRESENTATIVES

Gaze Zone	Neighbor gaze zones	yaw	pitch
1	9, 6	30.132	0.229
2	10, 8	-16.026	2.699
3	4, 6, 7	24.482	-10.004
4	3, 5, 6, 7, 8	-5.764	-5.317
5	3, 4, 6, 7, 8	-12.152	-1.054
6	1, 3, 4, 7, 9, 12	22.013	1.444
7	3, 4, 5, 6, 8, 9, 10, 12	0.000	0.000
8	4, 5, 7, 10, 2	-11.012	-2.882
9	1, 6, 7, 12	23.944	-0.693
10	2, 7, 8, 12	-13.946	4.756
11	12	-1.748	8.864
12	6, 8, 9, 10, 11	-8.486	0.934

The gaze-zone estimation needs to be initialized under a condition in which the zero point of yaw and pitch represents straight-ahead gaze direction, i.e., gaze zone 7. In this paper, this initialization was performed using the first 500 frames, and the center point was determined by a weighted average of estimated gaze direction.

In our method, yaw and pitch can be accurately evaluated but not roll; however, they are sufficient for gaze-zone estimation. The calculation for yaw and pitch depends on the deviated coordinates of the center of the bounding box compared with the coordinate of the frontal face center after the face detection step by the following:

$$\alpha = 2\arcsin\frac{y_d}{2l}, \beta = \arctan\frac{x_d}{l}$$

where x_d and y_d represent the row and column deviation from the zero point, respectively. l represents the distance from the center point of the driver's face to the rotary axes; there are two rotary axes of the driver's head. The process of frontal face detection will periodically operate to modify the zero point of yaw and pitch to enhance the correct rates for head-pose classification that is crucial for the current gaze-zone estimation.

We collected a standard data set of head poses for each gaze zone using a head motion sensor (see Table I). For each particular driver, the video system gradually revises the 12 exemplar sets, and proper elements are assigned to distinct exemplar sets with the highest probability. Typically, the personalized yaw and pitch values for each gaze zone are nearly stabilized after about 5000 video frames. In the following tracking, the system seeks to estimate head poses and gaze zones by finding the minimum distance between the previous exemplars and the current sample using (5).

Fig. 8 shows the yaw and pitch in the four driving circumstances, i.e., daytime and nighttime, and wearing glasses and wearing sunglasses. The convergence of the curves indicates that the driver mostly looked forward after the first couple of minutes of starting to drive.

As the system runs further, the confidence probability increases, as shown in Fig. 9. This helps in confirming the

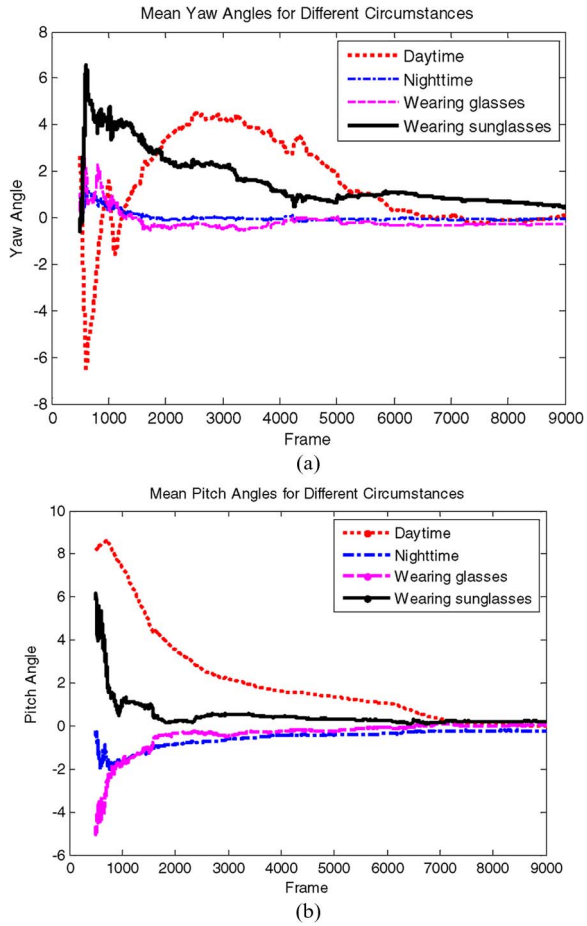


Fig. 8. Means of (a) yaw and (b) pitch from frames 501 to 9000 in each driving condition.

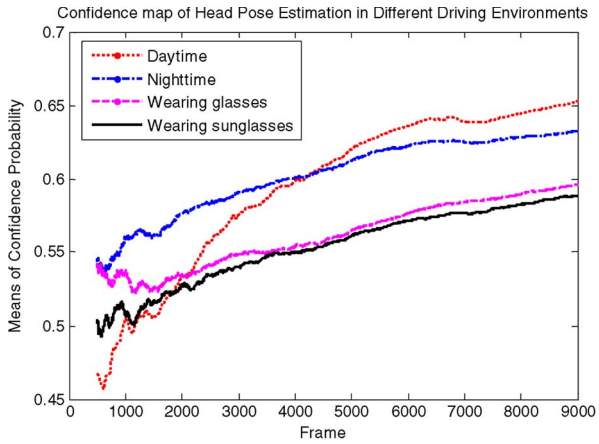


Fig. 9. Mean confidence probabilities of head-pose estimation increases along with the number of frames in different driving conditions.

tracking accuracy. The performance of the particle filtering greatly relies on the values of confidence probability, the ratio of matched features, and the total number of features of exemplars. The statistical values for confidence probability from frames 501 to 9000 in four driving conditions are shown in Table II. Although the confidence probabilities for the four driving conditions are somewhat different, which could be due to the particle filtering algorithm not fully converging, they are sufficient for our application.

TABLE II
STATISTICAL PROPERTIES FOR MEANS OF CONFIDENCE PROBABILITIES

	Daytime	Nighttime	Wearing Glasses	Wearing Sunglasses
mean	0.653	0.633	0.596	0.589
variance	0.013	0.074	0.107	0.111
standard deviation	0.112	0.271	0.327	0.334

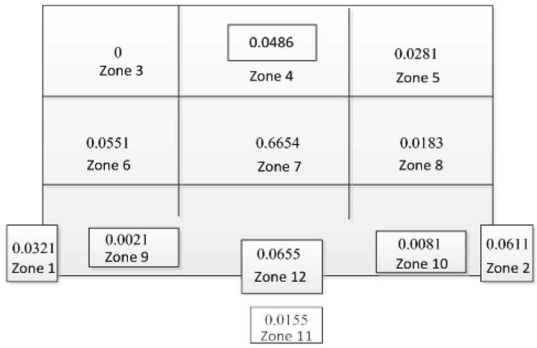


Fig. 10. Probability distribution of the patronized time of different zones.

Experiment results (see Fig. 11) shows that large errors of head motion velocities appeared only in a limited number of frames. In addition, the mean value of the velocity converges as the number of switch points becomes large, which indicates good stability of the driver's head movement. The value of 1.56 is assigned to the velocity threshold up to the current switch points for which condition. This threshold can be utilized in the other three conditions, provided that the mean velocity of head movement maintains high stability and consistency. The determination of threshold proceeds automatically while adopting the learning algorithm, and this value continues to be modified.

C. Gaze-Zone Statistics

Gaze zone 7 should be patronized in the majority of the time samples because drivers almost always look forward. Our experimental data showed that this is indeed the case. The probability distribution of patronized time for the gaze zone is depicted in Fig. 10. The result mainly relies on the classification in the learning algorithm. Thus, it reflects the performance of the learning algorithm.

Fig. 10 also shows that the times (fraction of frames) that the driver looked at zones 1, 2, 4, 6, and 12 are higher than the times the driver looked at other noncenter zones. In other words, the driver had a relatively higher chance to look at side mirrors, the review mirror, and the center console.

D. Estimation Result Compared With Ground-Truth Data

The motion-sensor ground-truth data were then used to compare with the head pose computed using our algorithms. Fig. 11 shows the tracking errors for randomly selected data segments as an example. It can be seen that the absolute mean errors at a steady state of yaw and pitch are about 2°. Although the error at the beginning was larger, the rapid convergence of these curves indicates the stability of the automatic gaze-zone

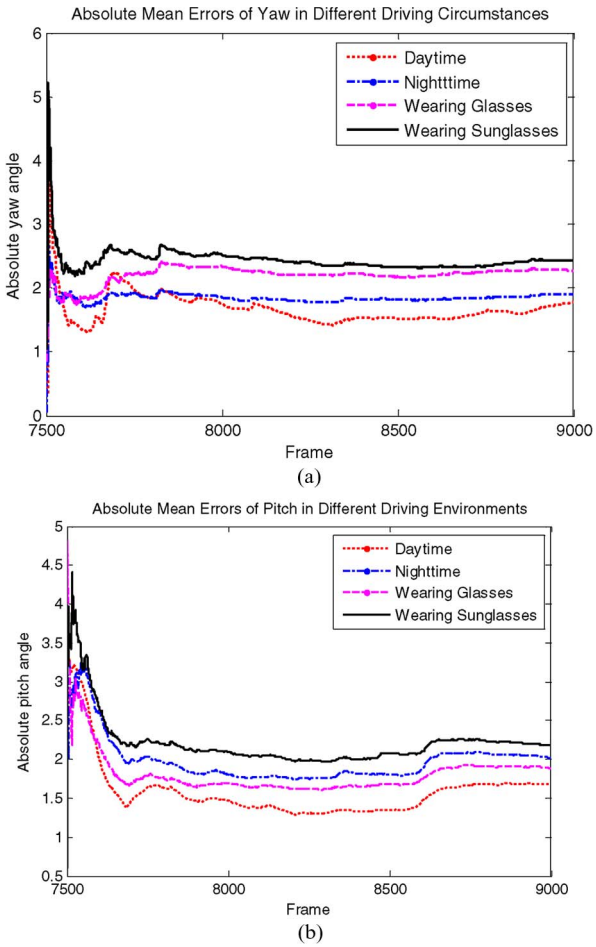


Fig. 11. (a) Absolute average error of yaw over four subjects from frames 7501 to 9000 in four scenes: daytime, nighttime, wearing glasses, and wearing sunglasses. (b) Absolute average error of pitch from frames 7501 to 9000 in four scenes: daytime, nighttime, wearing glasses, and wearing sunglasses.

estimation. It sometimes may take a longer time to reach a stable status while the final success detection rate (SDR) is still high. Slow convergence of errors can be resolved by modifying the center point and by running the learning algorithm (head-pose classification) more frequently.

We achieved high accuracy of head-pose and gaze-zone estimations. As shown in Table III, the absolute mean error is less than 5° (with sun glasses) and less than 2.3° in all other conditions. This result also serves as a separate evaluation of the learning algorithm since the performance of particle filtering has been estimated by the confidence map. The key to achieving a good result is that the zero coordinate of yaw and pitch are correctly determined, and the classification head poses are correct. Although it will generate a small error in selecting the coordinate system comparing with that created by standard data, such error is permitted in the overall estimation step. In Table III, we also show the SDR for yaw and pitch separately, each of which affects the correct gaze-zone estimation if the value of the error exceeds the corresponding threshold (9.5° and 4.5° for yaw and pitch, respectively). The success rate of gaze-zone estimation detection ranged from about 90% to nearly 100%. When the gaze-zone estimation was wrong, the incorrect selection was always to the adjacent zone.

TABLE III
STATISTICAL RESULTS FOR HEAD-POSE AND GAZE-ZONE ESTIMATION

		Yaw			Pitch		
Frames	Conditions	AME(°)	VAR	SDR (%)	AME(°)	VAR	SDR (%)
1500	Daytime	1.78	2.65	99.87	1.68	1.29	96.93
1500	Nighttime	1.90	2.54	99.73	2.02	3.86	91.00
1500	Wearing Glasses	2.28	3.76	99.33	1.89	3.17	93.13
1500	Wearing Sunglasses	2.44	4.39	99.13	4.73	4.34	89.87

Notation: AME: Absolute Mean Errors; VAR: Variance; SDR: Success Detection Rate of gaze zone.

E. Estimation Result Compared With Similar System

A system that is most similar to our system is described in [7]. It utilized the ellipsoidal face model to determine the driver's yaw angle. A support vector machine was adopted to train the exemplar set and each of the elements was assigned a specific gaze zone. The SDRs of our system are almost identical to their performance, as shown in [7, Tables V and VII]. An advantage of our technique is that our system can operate automatically, whereas the technique in [7] requires the driver's cooperation in the learning step.

V. CONCLUSION

We presented an approach to calibrating the eye gaze in a driving environment which achieved high accuracy of gaze-zone estimation while implementing an automatic learning algorithm and a particle filtering algorithm. This paper demonstrated robust estimates of the driver's gaze direction without need of the driver's cooperation in a calibration process. This advantage makes it suitable for wider application in driving research, particularly for monitoring driver inattention [35], [36].

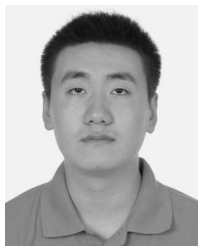
REFERENCES

[1] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1124–1133, Jun. 2006.
[2] V. Rantanen, T. Vanhala, O. Tuisku, P. Niemenlehto, J. Verho, V. Surakka, M. Juhola, and J. Lekkala, "A wearable, wireless gaze tracker with integrated selection command source for human–computer interaction," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 5, pp. 795–801, Sep. 2011.
[3] D. Model, E. D. Guestrin, and M. Eizenman, "An automatic calibration procedure for remote eye-gaze tracking systems," in *Proc. Annu. Int. Conf. IEEE EMBC*, 2009, pp. 4751–4754.
[4] A. Villanueva and R. Cabeza, "A novel gaze estimation system with one calibration point," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 1123–1138, Aug. 2008.
[5] C. Ahlstrom, T. Victor, C. Wege, and E. Steinmetz, "Processing of eye/head-tracking data in large-scale naturalistic driving data sets," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 553–564, Jun. 2012.
[6] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.
[7] S. J. Lee, J. Jo, H. G. Jung, K. R. Park, and J. Kim, "Real-time gaze estimator based on driver's head orientation for forward collision warning system," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 254–267, Mar. 2011.
[8] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proc. Int. Conf. Comput. Vis.*, 2001, pp. 50–59.
[9] A. Doshi and M. M. Trivedi, "Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 887–892.
[10] B. J. Lance and S. C. Marsella, "The expressive gaze model: Using gaze to express emotion," *IEEE Comput. Graph. Appl.*, vol. 30, no. 4, pp. 62–73, Jul./Aug. 2010.

[11] Z. Zhiwei and J. Qiang, "Novel eye gaze tracking techniques under natural head movement," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2246–2260, Dec. 2007.
[12] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.
[13] R. Stiefelhagen and J. Zhu, "Head orientation and gaze direction in meetings," in *Proc. Conf. Human Factors Comput. Syst.*, 2002, pp. 858–859.
[14] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 802–815, Feb. 2012.
[15] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 453–462, Sep. 2009.
[16] P. Jimenez, J. Nuevo, L. M. Bergasa, and M. A. Sotelo, "Face tracking and pose estimation with automatic three-dimensional model construction," *IET Comput. Vis.*, vol. 3, no. 2, pp. 93–102, Jun. 2009.
[17] K. P. Yao, W. H. Lin, C. Y. Fang, J. M. Wang, S. L. Chang, and S. W. Chen, "Real-time vision-based driver drowsiness/fatigue detection system," in *Proc. IEEE 71st VTC-Spring*, 2010, pp. 1–5.
[18] F. I. Kandil, A. Rotter, and M. Lappe, "Car drivers attend to different gaze targets when negotiating closed vs. open bends," *J. Vis.*, vol. 10, no. 4, pp. 1–11, Apr. 2010.
[19] P. Smith, M. Shah, and N. da Vitoria Lobo, "Determining driver visual attention with one camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 4, pp. 205–218, Dec. 2003.
[20] J. Y. Kaminski, K. D. Knaan, and A. Shavit, "Single image face orientation and gaze detection," *Mach. Vis. Appl.*, vol. 21, no. 1, pp. 85–98, Oct. 2009.
[21] P. Watta, S. Lakshmanan, and H. Yulin, "Nonparametric approaches for estimating driver pose," *IEEE Trans. Veh. Technol.*, vol. 56, no. 4, pp. 2028–2041, Jul. 2007.
[22] M. Bingpeng, S. Shiguang, C. Xilin, and G. Wen, "Head yaw estimation from asymmetry of facial appearance," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 6, pp. 1501–1512, Dec. 2008.
[23] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, Apr. 2009.
[24] J. Wu and M. M. Trivedi, "A two-stage head pose estimation framework and evaluation," *Pattern Recognit.*, vol. 41, no. 3, pp. 1138–1158, Mar. 2008.
[25] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2007, pp. 709–714.
[26] E. Murphy-Chutorian and M. M. Trivedi, "HyHOPE: Hybrid head orientation and position estimation for vision-based driver head tracking," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 512–517.
[27] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, Jun. 2010.
[28] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[29] K. Ohue, Y. Yamada, S. Uozumi, S. Tokoro, and A. Hattori, "Development of a new pre-crash safety system," presented at the SAE World Congr. Exhibition, Detroit, MI, 2006, SAE Tech. Paper 2006-01-1461.
[30] W. Rudin, *Functional Analysis*. New York: McGraw-Hill Science, 1991.
[31] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, May 2007.
[32] J. Handschin and D. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *Int. J. Control*, vol. 9, no. 5, pp. 547–559, 1969.
[33] J. Handschin, "Monte Carlo techniques for prediction and filtering of non-linear stochastic processes," *Automatica*, vol. 6, no. 4, pp. 555–563, Jul. 1970.
[34] N. Gordon, D. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussain Bayesian state estimation," *Proc. Inst. Elect. Eng.—Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
[35] D. Yanchao, H. Zhencheng, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 596–614, Jun. 2011.
[36] M. Wollmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, "Online driver distraction detection using long short-term memory," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 574–582, Jun. 2011.

Xianping Fu received the Ph.D. degree in communication and information system from Dalian Maritime University, Dalian, China, in 2005.

From 2008 to 2009, he was a Postdoctoral Fellow with Schepens Eye Research Institute, Harvard Medical School, Boston, MA. He is currently a Professor with Information Science and Technology College, Dalian Maritime University. His research interests include perception of natural scenes in engineering systems, including multimedia, image/video processing, and object recognition.



Xiao Guan received the B.S. degree in mathematics from Dalian Maritime University, Dalian, China, in 2012. He is currently working toward the Ph.D. degree in mathematics with Tulane University, New Orleans, LA.

Eli Peli received the B.S.E.E. and M.S.E.E. degrees from the Technion-Israel Institute of Technology, Haifa, Israel, and the O.D. degree from the New England College of Optometry, Boston, MA.

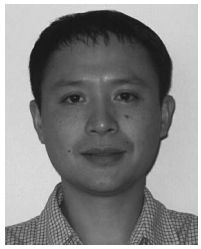
He is currently a Moakley Scholar in aging eye research, a Co-director of Research with Schepens Eye Research Institute, and a Professor of Ophthalmology with Harvard Medical School, Boston, MA. He is also a faculty member with the New England College of Optometry and Tufts University School of Medicine. His research interests include image processing in relation to visual function and clinical psychophysics in low-vision rehabilitation, image understanding, evaluation of display–vision interaction, and oculomotor control and binocular vision.

Dr. Peli is a Fellow of the American Academy of Optometry, the Optical Society of America, the International Society of Optical Engineering, and the Society for Information Display. He is currently an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Hongbo Liu received the Ph.D. degree in computer science from Dalian University of Technology, Dalian, China.

He is currently a Professor with the Information Science and Technology College, Dalian Maritime University, Dalian. In the last several years, he has been engaged in research and teachings on design and analysis of computer algorithms, agent systems, intelligent robotics, brain and cognition, etc. His research interests include system modeling and optimization involving soft computing, probabilistic modeling, cognitive computing, machine learning, data mining, etc.

Dr. Liu actively participates and organizes international conferences and workshops.



Gang Luo received the Ph.D. degree from Chongqing University, Chongqing, China, in 1997.

He is currently an Assistant Professor with Harvard Medical School, Boston, MA. He has been working in the fields of image processing and optics. His research interests include vision science.

Dr. Luo has been a Peer Reviewer for multiple journals across engineering and vision science.