

Extracting Driver's Facial Features During Driving

J. M. Wang, H. P. Chou, C. F. Hsu, S. W. Chen, and C. S. Fuh

Abstract—In this paper, a vision system for monitoring driver's facial features is presented. To begin, the driver's face is first located in the input video sequence. It is then tracked over the subsequent images. The facial features of eyes, mouth and head are kept detecting in the course of face tracking. Feature detection and tracking are performed in parallel so that the precise can be improved. A number of video sequences with the drivers of different ages and genders under various illumination and road conditions were employed to demonstrate the performance of the proposed system. Future work is on how to extend the system to determine the level of vigilance of the driver.

I. INTRODUCTION

There are many drivers who would agree to prohibit drunk driving, yet they think it's endurable to drive in a low level of vigilance, such as tiredness, somnolence, diversion, and illness. However, weary, drowsy, distracted, and sickening driving can be equally as dangerous as drunk driving [4][5] in view that all can decline visual activity, perceptual sensitivity, situational awareness, and decision-making capability.

Many techniques have been proposed for monitoring driver vigilance. They can be divided into three categories: (1) physiological response, (2) driving behavior, and (3) facial expression approaches. In the physiological response approach [1][2], the signals of polysomnography have been utilized to assess human alertness. However, for measuring driver's physiological parameters, sensors have been either installed on the seat, the steering wheel or embedded in such equipments as elastic strap, wristband, helmet, and contact lenses. These equipments somehow hinder driving operations and make the driver uncomfortable [3].

In the driving behavior approach [8], assorted kinds of driving data measured from the vehicle and the road have been applied to interpret driver vigilance. In order to gather diverse kinds of driving data, a variety of sensors are introduced into the vehicle. However, the installation of multiple sensing devices not only increases the cost but also can be subject to the limitations of vehicle type, driver experience, and road condition [6].

The facial expression approach [6], [7], [10] is primarily motivated by the human visual system that can effortlessly identify the vigilance level of a person based on his/her facial

expressions. Facial expressions convey inward feelings, including both psychological and physiological reactions [11]. Facial expressions relating to these two classes of reactions are typically distinct. In this study, the physiological reactions of fatigue and drowsiness are of concern. The associated facial expressions include eye blinking, gaze fixation, yawn, and head nodding.

It has been known that color information provides important cues for object detection, tracking and recognition [9]. There are many face detection researches based on chromatic evidences [12], [13]. However, very few works [10], [12], considered the facial feature detection of drivers from their color videos inside vehicles. Potential challenges to such a work include video instability due to moving vehicles, rapid illumination variation resulting from ambient lights, abrupt lighting change, and partial occlusion.

In this study, we develop an in-vehicle vision system for monitoring driver facial features using a video camera without supplementary light. The rest of this chapter is organized as follows. Section II outlines the proposed system, including its configuration and workflow. Section III details the implementations of face locating. The face model and facial features locating are addressed in Section IV. Section V demonstrates the feasibility of the proposed system. Concluding remarks and future work are given in Section VI.

II. ARCHITECTURE

In this section, the configuration and workflow of the proposed system are addressed.

A. System Configuration

The system is composed of three major components: a video camera, a host computer, and a warning device. Of these three components, the camera plays an important role in determining the technique and workflow of the system. A proper installation of the camera could reduce the degrees of difficulties. In this study, we place a video camera on the top of the dashboard right behind the steering wheel. The camera has a tilt angle of about 30 degrees to the driver's face. Fig. 3(a) shows an image of a driver with different head orientations taken by the video camera.

B. System Workflow

Fig. 1 shows a block diagram for the proposed facial feature extraction system. There are four major blocks involved in the diagram, denoted by face location, facial feature location, face model updating, and reasoning (future work). Each block corresponds to an essential step of the system.

Manuscript received April 8, 2011.

J. M. Wang is with National Taiwan University, Taipei, Taiwan. Phone: +886-9-22266003; e-mail: d97922030@ntu.edu.tw.

H. P. Chou is with Chung-Hua University, Hsin-Chu, Taiwan.

C. F. Hsu is with National Taiwan Normal University, Taipei, Taiwan.

S. W. Chen is with National Taiwan Normal University, Taipei, Taiwan.

C. S. Fuh is with National Taiwan University, Taipei, Taiwan.

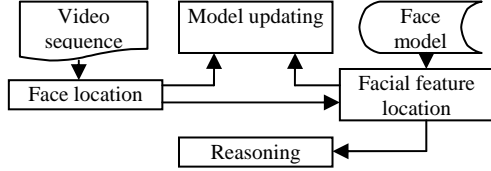


Fig. 1. Block diagram for the proposed driver drowsiness monitoring system.

Considering an input video sequence, the driver's face is first located in a video image. Facial features, including eyes and mouth, are next located within the located face with the aid of a predefined face model. The initial face model shown in Fig. 6(a) simply describes the relative relationships between facial features. Three detectors have been trained by adaboost to detect face, eyes and mouth in images. The system adapts the face model according to the detected facial features so as to preserve their latest structure for later use.

III. FACE LOCATION

Fig. 2 shows the flowchart for face location. In the beginning, we need processing for locating the driver's face without previous information. The processing can be divided into two steps: motion detection and face detection. Motion detection is designed to reduce the processing range, while face detection is designed to locate the driver's face in the rage. After have the face location, we can apply tracking technique to instead of the motion detection in the following frames.

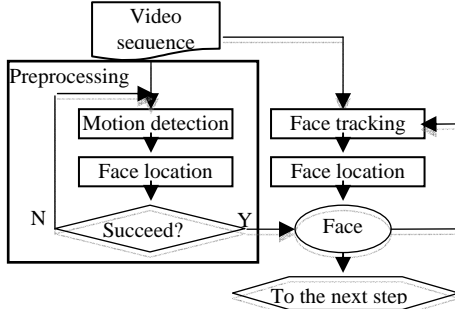


Fig. 2. Flowchart for face location.

In the video sequence, we can just compare the two successive frames to obtain the differences. These differences often are caused by the object motion and illumination change. We can suppose that the illumination cannot have a significant change between two successive frames, so they often shows parts of the moving objects instead of the static objects. In this application, the moving objects that we are interested would be the driver's face. This process spends a very short time but can help us to locate the region of interest efficiently. Fig. 3(b) shows the example of the difference image by comparing previous and current frames.

After accumulating the series of the difference images, we can obtain the motion image (Fig. 3(d)) in the monitoring range. The motion image obtained here is treated as a mask image for to detect face using the Viola-Jones like face detector [18] announced for OpenCV [19], which is widely used in those application for face detection. That will reduce the processing time and limit the searching result in the

motion parts to avoid false alarm. Fig 7(a) shows the face detecting result.



Fig. 3. (a) previous frame; (b) current frame; (c) difference image; (d) after accumulating the series of difference images.

Some problems may occur in this works. At first, the driver and passengers' faces may be extracted at the same time (Fig 7(b)). Because our driver's face should be the largest one at the middle of the monitoring range, we can locate the largest one as our driver's face. The second problem is the errors of face detection (Fig. 7(c)(d)). This error will cause others errors in the following works, so we have to improve the detection result in the initialization. Here, we confirm the detection result as the driver's face after we have n ($=2$) same detection results in the sequential frames. The probability of the detection result being correct would be $(1-p^n)$, where p is error detection rate. As the test shown in [16], p is about 15% and the correction rate will be about 97.8% if n is set as 2 (ex. Fig. 4(d)).

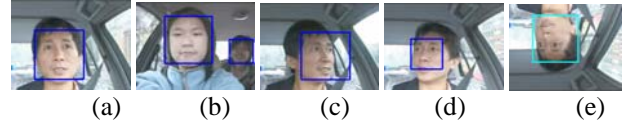


Fig. 4. (a) Face detection result, (b) multiple face detection result, (c) face detection with wrong size, (d) error detection result, and (e) after double check of (d).

Once we have the face region in the preprocessing step, we can apply a tracking technique with face detection at the same time. The tracking result helps us to reduce the search region in the following detection step. In two successive image frames, we can suppose that the illumination would not be changed significantly, so the face color would be similar between the faces in them. We also can suppose that the driver's face could not have a large moving in driving, so the face position in the current image frame would be close to the previous position. Under these properties, mean shift method [14] is applied here for tracking.

The tracking result is then further applying face detection on the neighborhood to confirm. Three cases may happen right here. At first, we can obtain the face, and then the detection result is considered as the new face location. In the second, we cannot detect the driver's face while he or she faces to the side, the tracking result is still considered as the face location. Finally, if we could not detect the face and the measurement was smaller than a threshold value, the driver may leave the driver seat and we need to reinitialize the detection processing.

IV. FACIAL FEATURE LOCATION

After the face location step, we will have the face position no matter the driver's facing. In this step, we determine the facial feature inner the face region. There are three features to be detected: eye position, mouth position, and facing

direction. System flowchart is shown in Fig. 5. The face model is defined beforehand. Using the transformation state, we can project the face model to the image plane to obtain the previous positions of the facial features. The particle filter is applied here for tracking the facial features. After we have the expected positions of features, eye and mouth detection is applied to confirm the tracking result. Detection results are then referred for updating the transformation state and even the face model.

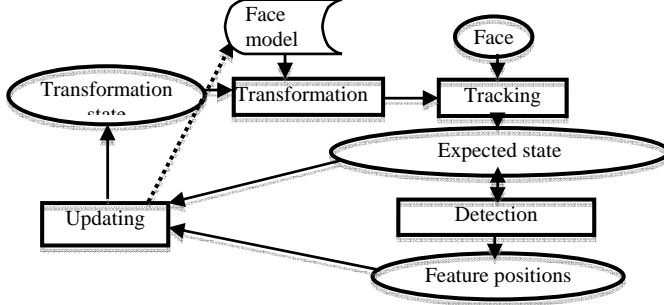


Fig. 5. System flowchart of facial feature detection.

A. Transformation

In a three-dimensional Euclidean space, let us describe the face model in a coordinate frame that is defined using three basis vectors \mathbf{i}, \mathbf{j} , and \mathbf{k} orthogonal to each other. Each point \mathbf{P} in this coordinate frame can be represented using a coordinate vector $\mathbf{P} = (x, y, z)^T$. Consider another coordinate frame with the same origin point and another three orthogonal unit vectors. The point \mathbf{P} can be represented using a new coordinate vector \mathbf{P}' in this new coordinate frame by combining a rotation to the following transformation:

$$\mathbf{P}' = \mathbf{R}\mathbf{P},$$

where \mathbf{R} is a rotation matrix and is a combination of the rotation matrices, \mathbf{R}_α , \mathbf{R}_β , and \mathbf{R}_γ , about the \mathbf{i}, \mathbf{j} , and \mathbf{k} axes of the coordinate frame F :

$$\mathbf{R} = \mathbf{R}_\alpha \mathbf{R}_\beta \mathbf{R}_\gamma,$$

where α, β , and γ are pan angle, tilt angle, and rotation angle of the face respectively.

In this step, we project the face model (Fig. 6(a)) on the image plane according to the transformation state. Suppose the image plane is parallel to the i - j plane, we can compute the transformation result is $\mathbf{P}' = f\mathbf{R}\mathbf{P} + \mathbf{O}'$, where f is the scaling value and \mathbf{O}' is the offset value. Finally, the transformation state consists of α, β, γ, f , and the \mathbf{O}' . Fig. 6(c) shows the transformation result according to a transformation state and (d) shows the projecting result.

The face model is defined as three points, two eyes and mouth positions. The original point $(0, 0, 0)$ is set as the middle point between two eyes. Their coordinates are measured from a driver's face in a monitoring image (Fig. 6(a) shows the coordinate values). For each point (feature), some feature values are stored for tracking: region size (width and height), histogram of the color distribution, and the histogram of the edge orientation.

Since the face model is defined from an actual monitoring image, so that we have $f = 1$ as the initial scale

value. The same idea also is applied here for defining the offset value \mathbf{O}' , that is, we measure the offset of the facial feature to a face in a monitoring image as the initial offset value. Finally, we can set $\alpha = 0, \beta = 0$, and $\gamma = 0$ as the initial state values, because the face will be located at first time while the driver faces to the front direction after starting the system.

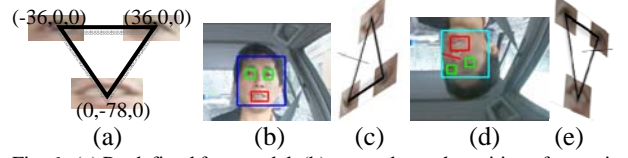


Fig. 6. (a) Predefined face model, (b) eye and mouth position after projecting the face model on the image plane, (c) the face model after transformation, and (d) the projecting result of (c). (e) is different facing direction, but they have similar relationship between facial features as (c).

B. Feature Tracking

In our application, we want to know the eye and mouth positions that are transformed from the face model (Figure. 9(a)) by the transformation state (hidden states) according to the monitoring image sequence (observations). Since the actual transformation state cannot be known, its distribution given the passing observations, $z_{0:t} = (z_0, \dots, z_t)$, is defined as $p(s_t | z_{0:t})$, where s_t is the transformation state at time t .

Using Bayesian rule, this prior probability can be updated to

$$p(s_t | z_{0:t}) = \frac{p(z_t | z_{0:t-1}, s_t) p(s_t | z_{0:t-1})}{p(z_t | z_{0:t-1})}, \quad (1)$$

where $p(z_t | z_{0:t-1})$ is the predictive distribution z_t given the past observation $z_{0:t-1}$, and it is a normalizing term in most case. Assume that $p(z_t | z_{0:t-1}, s_t)$ depends only on s_t through a predefined measurement model $p(z_t | s_t)$. Equation (1) can be rewritten as

$$p(s_t | z_{0:t}) = \alpha p(z_t | s_t) p(s_t | z_{0:t-1}), \quad (2)$$

where α is a constant.

Now, suppose s_t is Markovian, then its evolution can be described through a transition model, $p(s_t | s_{t-1})$. Based on this model, $p(s_t | z_{0:t-1})$ can be calculated using the Chapman-Kolmogorov equation:

$$p(s_t | z_{0:t-1}) = \int p(s_t | s_{t-1}) p(s_{t-1} | z_{0:t-1}) ds_{t-1}. \quad (3)$$

Equations (2) and (3) show that we can obtain $p(s_t | z_{0:t})$ recursively if we have the following requirements: the measurement model $p(z_t | s_t)$, the transition model $p(s_t | s_{t-1})$ and the initial distribution $p(s_0 | z_0)$.

Particle filter [15] is a Monte Carlo method that uses m particles, $s_t^{(i)}$, $i = 1 \dots m$, and their corresponding weights, $w_t^{(i)}$, to simulate the distribution $p(s_t | z_{0:t})$. The tracking result, s_t^* , can be calculated to the expected value of $p(s_t | z_{0:t})$:

$$s_t^* = E[p(s_t | z_{0:t})] = \sum_{i=1}^m w_t^{(i)} s_t^{(i)},$$

where the sum of $w_t^{(i)}$ is normalized to 1 for convenient. Here, we use mean shift [14] instead of computing expected value to locate the final result.

B.1 Measurement model

In the face model, we also store the region sizes and the hue histograms, H_{le} , H_{re} , and H_m , of the facial features. Given a transformation state s , the face model can be projected on the image plane and we can compute the hue histograms, H_{le}^s , H_{re}^s , and H_m^s , for each facial feature in the current image.

Bhattacharyya coefficient $BC(p, q) = \sum_x \sqrt{p(x)q(x)}$ is computed for measuring the similarity between histograms p and q , where x is the bin value. The measurement model $p(z_i/s_i)$ is then defined using the following equation:

$$p(z_i | s_i) = BC(H_{le}, H_{le}^{s_i}) \cdot BC(H_{re}, H_{re}^{s_i}) \cdot BC(H_m, H_m^{s_i}). \quad (4)$$

Under this computation, there will be many particles computed to small values and so is the $p(s_i/z_{0:t})$ values. To prevent the influence of the particles with small value, their weights are set to 0 if their weights are smaller than a threshold.

B.2 Transition model

Because only one face (state) will be located in our application, to reduce the processing time and memory storage, we set all $s_{t-1}^{(i)}$ as s_{t-1}^* to simplify the resample step. Under this modification, we will require fewer particles and fewer processing time.

In driving, the driver may face to the front in most of the time, which means that most of the parameters, α , β , and γ , tend to zero in our application. Assume the current α value is α_t , the transition model for α is defined as

$$p(\alpha_t | \alpha_{t-1}) = G(\frac{\alpha_{t-1}}{2}), \text{ where } G(x) \text{ is a Gaussian function with}$$

mean x . The same definition is also applied for β , and γ . The last parameter, f , value is defined in the similar way. Because f value tends to 1 instead of zero, it is defined as

$$p(f_t | f_{t-1}) = G(\frac{1+f_{t-1}}{2}). \text{ Finally, the driver's face may have}$$

unexpected movement in driving, so we define the offset value at time t , \mathbf{O}'_t , as $p(\mathbf{O}'_t | \mathbf{O}'_{t-1}) = G(\mathbf{O}'_{t-1})$.

Combining the above the transition model for each state element, we can define the transition model $p(s_t | s_{t-1}^{(i)})$.

The face model projected on the image plane with different pan angles may have the same relationship between the facial features (Fig. 6(e)). To solve that problem, we define the pan angle face to left if the offset value is located on the left half of the face region, or otherwise. After computing the new state values of the particles, the pan angle α in each particle will be checked and given a new value $-\alpha$ if its corresponding offset value does not match the above criteria.

C. Eye and mouth detection

After the expectation transformation state of the face model is obtained, we can transform the face model on the image plane to obtain the expected positions of eye and mouth. Near the corresponding expected positions, eye and mouth detector trained by M. Castrillón [17] are applied to locate their positions more precisely. The detection results will give more hints for the measurement model. If any facial

feature is located, all weight values $w^{(i)}$ are recalculated again to produce a better expected transformation state s_t^* .

In Equation (4), $BC(p, q)$ will be set to 1 if the facial feature has corresponding detection result. We can believe the detection result because of the following reasons: First, as the detector applied in face detection, such detectors also have the advantage of less false alarm rate. Second, because the face region has been confirmed, the influence outside the face region cannot be existence right now. Third, since we have reduced the detection region according the tracking result, most of the error detection because of the outside objects will be eliminated at the same time. Fig. 7(b) shows a better location result of (a) after the process of eye and mouth detection. Fig. 7(c) shows the tracking result when the eye and mouth are not detected.

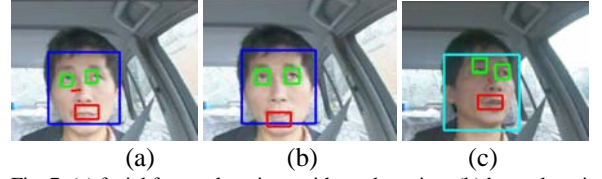


Fig. 7: (a) facial feature locations without detection; (b) better location result of the facial features because of eye and mouth detection; (c) We still have the expected facial positions even the eye and mouth are not detected.

D. Updating

The updating step is designed for two purposes: the first one is to update the transformation state, while the second one is to update the face model. Updating the transformation state is to replace the previous transformation state s_{t-1} with the current one s_t^* and play s_t^* as s_t in the next frame. Updating the face model is for keeping the face model as the driver along time.

Our tracking algorithm for facial features is designed on the information of the face model. If the face model more matched to the current driver's facial features, we will have a better tracking result. So, the model updating is designed for the following reasons: At first, the face model is designed using some initial parameter values. Updating the model will let the parameter values match the current driver's feature values. Second, even the face model match the current driver's face feature, the driver may be changed in time. Third, the illumination may be changed during time, so is the color information of the facial features. At last reason, the driver may have a slight pose change in driving, for example, shifting his or her seat.

The process for updating face model is activated under all of the following criteria being held:

1. Face region has been detected in the previous step.
2. Two eyes and one mouth have been detected.
3. The face orientations (α , β , and γ) are close to zero.
4. The relationship between facial features is similar to the original face model.

The first criterion would be held only when the driver face to the front, then we can have the correct relationship of facial features. The second situation means that we have the whole facial features for constructing the face model. The third

criterion is similar to the first one, but we enforce the face with no pan and tilt angle. Finally, to prevent the suddenness, we set the forth criterion.

V. EXPERIMENTAL RESULTS

The proposed driver's drowsiness detection system has been developed using the Xcode run on an Intel Core 2 Duo 2GHz PC running under Mac OS X. The input video sequence to the system is at a rate of 30 frames/second. The size of video images is 320 x 240 pixels. We divide our experiments into two parts. The first part examines the accuracies and efficiencies of the face location steps of the system process. The second part demonstrates the results of the facial feature location. Fig. 8 shows the 16 video sequences tested in our system.



Fig. 8. Video sequences for testing

A. Face Location

Here, we prove that combining detection and tracking can help to locate the face more precisely. Viola-Jones like face detector are applied here for detecting face. The average processing time for detecting the face is 0.02 second. Combining with the tracking algorithm (mean shift) and setting the region of the interest to reduce the searching area for face detector, we can have the face location in the whole time. The average processing times for tracking and detection are 0.01 and 0.005 seconds respectively, and the total processing time for each frame is about 0.015 seconds.

We say that the face location is correct only if the locating block covers all facial features (eyes and mouth) and more than 3/4 block region is face region. Fig. 9 shows some examples of the fail location results. Fig. 10 shows the location and detection rates for the video sequences. The location rate is defined as the frame number of face location (our method) over the total frame number, while the detection rate means the frame number of face detection (Viola-Jones like face detector) over the total frame number.



Fig. 9. Face location would be fail because of the (a) low illumination, (b) over high illumination, (c) turning the head, and (d) shadow on the face.

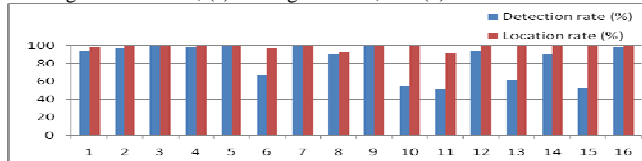


Fig. 10. Correct location rate (right) and detection rate (left) for the face in

the video sequences.

The experimental results show that we can locate the driver's face in most of the time. However, there are still some cases with lower location rate. For example, in video 6, this driver turns her head frequently. Since the measurement model is based on the hue value of the face, the location region falling on the neck when the driver turns her head as shown in Fig. 9(c).

In the video 8, the vehicle is driven in the tunnel with low illumination. The position of the face is hard to recognize in such environment. We may lose the face location in the dark. In the video 11, the camera is set to over exposure. Over exposure is the same problem as that in the tunnel. The edges in the image are not visible, and neither face detector nor face tracking can locate the face position precisely. In the video 1, this vehicle is driven in sunny day. The sunlight falls on the driver's face, so the other part of the face is darker than the bright part.

B. Facial Feature Location

Fig. 11 shows the positions of eyes and mouth. The vertical lines show the frame with the fail detection: missing eyes or missing mouth. The three horizontal lines represent the x -positions of the right eye, mouth, and left eye from top to bottom. In this experiment, we can see that all of the facial features cannot be obtained at the same time (ex. Fig. 11(a)). The mouth may be missed because of the following reasons: the driver is not facing to the front (Fig. 11(d)) and the mouth is occluded (Fig. 11(e)), while the eyes may be missed because of eye closing (Fig. 11(c)) besides of the above reasons.

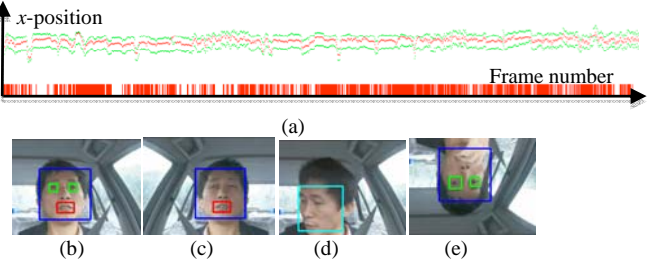


Fig. 11. (a) x -position of the eyes and mouth located in a video sequence. (b) Eyes and mouth are detected at the same time. The facial features may not be detected because of (c) eye closing, (d) the driver facing to the side, and (e) occlusion.

Fig. 12 shows the comparison between the location and detection rates. The location rate is defined as $N_l/N \times 100\%$, where N_l is the frame number of all facial features being located at the same time, and N is the total frame number. The detection rate is defined as $N_d/N \times 100\%$, where N_d is the frame number that all facial features are detected by the Viola-Jones like detector. The experiments show that the detection rate often is smaller than 50%, and our method can locate the facial features over 80% in most of the videos.

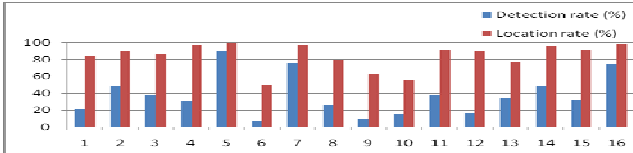


Fig. 12. Correct location rate (right) and detection rate (left) for the facial features in the video sequences.

Most of the error locations (Fig. 13) often are occurred under the unusual scenes. In video 6 and video 10, the drivers turn her/his head frequently and with large pan angle, so we cannot touch her/his face during most of the time. The video 8 is extracted in the case of driving in tunnel, which has low illumination so that we cannot have significant information (color or edge) for tracking or detection. That can be improved if IR camera could be set right here, which will be our future works.

The type of the driver's face would have slight influence in our system. The low location rate for video 9 is because of the driver's smiling during the detection time. Her eyes are so close that we cannot detect and track them, which is the reason why we have low detection rate (9.8%) and low location rate (63.3%). The location in video 13 also has the same problem. The driver put his hand on the face more frequently than the others, so the face is occluded and we cannot locate the facial features.

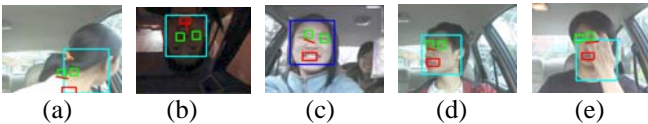


Fig. 13. Some error location cases extracted from (a) video 6, (b) video 8, (c) video 9, (d) video 10, and (e) video 13.

In our following applications, the facial features are located for estimate the level of the driver's vigilance. Most of the cases (except for video 8) in low location rate are under the situations that the drivers have frequent motion of the head. That means that they are in high vigilance that we don't need to worry about his/her drowsiness. When the driver has no motions of facial features, we will need to determine the vigilance and our system has high location rate in such cases. More experimental results are shown in our website: <http://www.csie.ntnu.edu.tw/~ipcv>

VI. CONCLUDING REMARKS AND FUTURE WORK

While there have already been many vision systems reported for detecting driver's drowsiness/fatigue, few systems utilized chromatic images as their input data. In this article, we presented a system for monitoring driver's features, which uses as the input data color images acquired by a video camera. Although colors provide rich information, they really suffer from low intensity and brightness variation. We introduced a combination of detection and tracking process in our system. The process can significantly improve the location rate of the facial features under many bad environments.

Facial feature detectors trained by ADABOOST approach is incorporated in the facial feature detection step. Particle filters and mean shift methods are used to trace face over

video sequences. A number of video sequences of different drivers, genders, accessories, and illumination conditions have been employed in the experiment for examining the efficiency and effectiveness of the proposed system.

REFERENCES

- [1] R. Agarwal, T. Takeuchi, S. Laroche, and J. Gotman, "Detection of rapid-eye movement in sleep studies," *IEEE Trans. on Biomedical Engineering* vol. 52, no. 8, pp. 1390-1396, 2005.
- [2] B. C. Carmona, G. F. Capote, B. G. Botebol, L. P. García, LP, A. A. Sánchez, and G. J. Castillo, "Assessment of excessive day-time sleepiness in professional drivers with suspected obstructive sleep apnea syndrome," *Arch Bronconeumol*, vol. 36, no. 8, pp. 436-440, 2000.
- [3] I. G. Damousis and D. Tzovaras, 2008, "Fuzzy fusion of eyelid activity indicators for hypovigilance-related accident prediction," *IEEE Trans. on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 491-500, 2008.
- [4] P. H. Gander, N. S. Marshall, R. B. Harris, and P. Reid, "Sleep, sleepiness and motor vehicle accidents: a national survey," *Journal of Public Health*, vol. 29, no. 1, pp. 16-21, 2005.
- [5] H. Häkkinen and H. Summala, "Sleepiness at work among commercial truck drivers," *Sleep*, vol. 23, no. 1, pp. 49-57, 2000.
- [6] Q. Ji, Z. Zhu and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Trans. Vehicular Technology*, vol. 53, no. 4, pp. 1052-1068, 2004.
- [7] M. Lalonde, D. Byrns, L. Gagnon, N. Teasdale, and D. Laurendeau, "Real-time eye blinking detection with GPU-based SIFT tracking," *Proc. of 4th Canadian Conf. on Computer and Robot Vision*, pp.481-487, Montreal, 2007.
- [8] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 340-350, 2007.
- [9] R. Lukac and K. N. Plataniotis, *Color Image Processing, Methods and Applications*, CRC Press, Taylor & Francis Group, New York, 2007.
- [10] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse Bayesian learning," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 431-440, 2007.
- [11] Y. Mitsukura, H. Takimoto, M. Fukumi, and N. Akamatsu, "Face detection and emotional extraction system using double structure neural network," *Proc. of the Int'l Joint Conf. on Neural Networks* vol. 2, pp. 1253 -1257, 2003.
- [12] I. Takai, K. Yamamoto, K. Kato, K. Yamada, and M. Andoh, "Robust detection method of the driver's face and eye region for driving support system," *Proc. of 16th Int'l Conf. on Vision Interface*, Halifax, Canada, pp. 148 -153, 2003.
- [13] Z. Zhang, and J. Zhang, "A new real-time eye tracking for driver fatigue detection," *Proc. of 6th Int'l Conf. on ITS Telecommunications*, pp. 8-11, 2006.
- [14] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, 1995.
- [15] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc.-F Radar and Signal Processing*, vol. 140, no. 2, pp. 107-113, 1993.
- [16] R. Sznitman and B. Jedynek, Active testing for face detection and localization, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1914-1920, 2010.
- [17] M. Castrillon, O. Deniz, D. Hernandez, and J. Lorenzo, "A comparison of face and Facial feature detectors based on the Viola-Jones general object detection framework," *Machine Vision and Applications*, doi:10.1007/s00138-010-0250-7, 2007
- [18] P. Viola and M. Jones, Fast and robust classification using asymmetric adaboost and a detector cascade, *Neural Information Processing System*, vol. 14, pp. 1311-1318, 2002
- [19] A. F. Reimondo, (2008, September 2), *Haar Caasades*, [Online]. Available: <http://alereimondo.no-ip.org/OpenCV>