

Monitoring Driver Fatigue Using Facial Analysis Techniques

Sarbjit Singh
sasingh@cs.umn.edu

Nikolaos P. Papanikolopoulos¹
npapas@cs.umn.edu

Artificial Intelligence, Robotics, and Vision Laboratory
Department of Computer Science and Engineering,
University of Minnesota
Minneapolis, MN 55455 U.S.A.

Abstract

In this paper, we describe a non-intrusive vision-based system for the detection of driver fatigue. The system uses a color video camera that points directly towards the driver's face and monitors the driver's eyes in order to detect micro-sleeps (short periods of sleep). The system deals with skin-color information in order to search for the face in the input space. After segmenting the pixels with skin like color, we perform blob processing in order to determine the exact position of the face. We reduce the search space by analyzing the horizontal gradient map of the face, taking into account the knowledge that eye regions in the face present a great change in the horizontal intensity gradient. In order to find and track the location of the pupil, we use gray scale model matching. We also use the same pattern recognition technique to determine whether the eye is open or closed. If the eyes remain closed for an abnormal period of time (5-6 sec), the system draws the conclusion that the person is falling asleep and issues a warning signal.

1. Introduction

A large number of automobile accidents are caused due to driver fatigue. The U.S. National Highway Traffic Safety Administration has indicated that driver fatigue is responsible for as many as 240,000 motor vehicle accidents a year in the U.S.A. alone. Sleep deprivation and sleep disorder are becoming common problems for car drivers in a society in which people seem not to have enough time to perform all the activities they need to carry out on a daily basis. By monitoring the driver's symptoms, we can determine driver fatigue early enough as to take preventive course to avoid an accident due to lack of awareness.

There are many indicators of oncoming fatigue, some of which are possible to detect with the use of a camera. Two well-known symptoms that we consider feasible to be detected are micro-sleeps (short periods, 2-3 seconds, in

which the driver loses consciousness) and the forward or backward bouncing movement of the driver's head.

At the moment, we focus on the detection of micro-sleeps. The input to the system is a continuous sequence of images fed from a video camera. From this sequence, the system can analyze the eyes in each image, as well as compare the eyes between frames.

A previous version of the system [1] used computer vision techniques to extract and track eye locations throughout the entire video sequence. However, the techniques described in this paper have significant forward steps towards improved accuracy and robustness in the process of detecting driver fatigue.

Localizing the eyes at the first frame is the most computationally expensive phase of the tracking system. In this phase, the system has no previous information about the eye locations in the image. The system has to find the area of the image that will be used in subsequent frames in order to track the eyes. During the tracking phase, the search space is reduced as the system has an approximate knowledge of the eye's position from the previous frame. This tracking can be done at a relatively low computational cost. In order to detect failure in tracking, general constraints such as the distance between the eyes and the horizontal alignment of the two eyes are used. To make sure that the correct feature is being tracked, the eyes should be relocated periodically, even if no failure has been detected. By determining whether the eyes are open or closed during eye tracking, we can determine if there are any micro-sleep symptoms that can help us determine driver fatigue.

The next section describes some related work in the area of facial feature extraction and tracking and in the detection of driver fatigue. The eye tracking and driver fatigue detection steps are described in subsequent sections. Finally, we present our experimental results.

2. Related work

There have been several studies about motor vehicle driver fatigue [3]. There are even some commercial systems

¹ Author to whom all correspondence should be addressed.

in the market that detect and signal driver unawareness. Most of these systems monitor driver's steering patterns. When these patterns are unusual, a warning signal is issued. Facial feature extraction is an active research area in computer vision. Yoo [4] proposed an approach that takes into account facial symmetry in order to find the placement of the face in an image. Sophisticated methods such as eigen-space matching [5], contour tracking using snakes [6], deformable template [7], and ellipse fitting techniques are computationally very expensive. Skin color based detection [8,9] is a widely used technique to extract face region from background images. Eye tracking is also being widely studied due to its potential applications in multi-modal user interfaces [10,11].

This paper describes advances made to the vision-based system for the detection of driver fatigue described originally in [1,2]. The essentials of micro-sleep detection are present in this new version of the system, however a significantly different detection technique has been used in order to increase the accuracy and robustness.

3. Overview of the system

Two well-differentiated functional phases have been defined in the system: the Eye Tracking and the Fatigue Detection. The eye tracker receives the first frame from the camera. At this point, it is assumed that there is no previous knowledge about the locations of the eyes. We use the initial frame to localize the eyes within the entire input image. It may be the case that due to unfavorable illumination conditions or head orientation in the initial image, the eye localization may fail. In this case, we have to grab a new initial frame and apply the localization algorithm in order to find the positions of the eyes. This process is repeated until we are confident about the positions of the eyes. After the portion of the image containing the driver's eyes is estimated, the system enters into its tracking mode. In this mode, the search space for eyes in subsequent input frames is reduced to a small area surrounding the eye regions. This is based on the assumption that the driver's head exhibits very small displacements in the time required to grab and process consecutive input frames. During tracking, error-detection is performed in order to recover from possible tracking failures. If an error occurs, the system gets out of tracking mode and recalculates the locations of the eyes from the current image frame.

The fatigue detection phase is closely related to the tracking phase. The information obtained from processing each consecutive input frame in the eye-tracking phase is fed directly to the fatigue detection phase if no tracking error has occurred. This phase is responsible of extracting the information from the image of the eye that will determine signs of micro-sleeps. At each frame, when the eyes are localized, the system determines whether the eyes are open or not. Thus, we are able to tell when the eyes have been closed for too long. In practice, we count the number of consecutive frames during which the eyes are closed. If this number gets too large, we issue a warning signal.

The fatigue detector system is to be mounted at the top of the dashboard inside a vehicle, pointing towards the driver's face. The system does not have the capability to control tilt or zoom factors in order to adjust for the driver's head movements. However, it is desirable to have this capability in order to get a better resolution inside the region of interest. For experimentation, we are using a JVC color video camera, and a Pentium Pro 200 MHz personal computer with a Matrox Genesis imaging board that uses the Texas Instruments TMS320C80 DSP processor.

4. The eye tracker

As described in [1], an eye tracking system can be divided into three functional units: localization of the eyes, tracking of eyes, and estimation of the tracking error.

4.1 Localization of the eyes

In order to reduce the search space and to avoid any kind of distracting features, the first step we followed was to extract the driver's face from the input image.

In [1], we had used a symmetry based approach similar to [4], in which it is assumed that there exists a horizontal symmetry on human faces and that we could reduce the search space for eyes by focusing our attention on a stripe of image around this symmetry line. This approach, although very fast, showed to be quite fragile because the system expected that the driver's face be in a totally frontal view every time the system entered into the face detection step. Moreover, the technique gave information about the possible left and right margins of the face but was incapable to provide any prediction of the upper and lower margins of the face.

We have adopted an approach that uses skin color information in order to predict and track the position of the driver's face. Skin color models have shown to be very successful to segment human faces in noisy images. We have used the approach suggested in [9]. The main idea behind most skin color models lays on the notion that skin pixels do not vary as much in color as they do in brightness. This is true even for individuals from different races.

According to [9], skin color pixels are clustered in chromatic color space and they can be represented using a Gaussian model $N(m, \Sigma^2)$, where $m = (\bar{r}, \bar{g})$, with

$$\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i,$$

$$\bar{g} = \frac{1}{N} \sum_{i=1}^N g_i,$$

and

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix}$$

where r_i and g_i represent red and green values of each pixel (the procedure for creating skin color model is fully explained in [8,9]).

Once the skin model was created, we used it in our system to filter the incoming video frames to allow only those pixels that had a high likelihood of being face pixels to survive. This allowed us to quickly detect the region of the image where the face is located. In order to reduce the computational cost, we resample the input image of resolution of 640x480 into a 160x120 frame. Figure 1(b) shows the result of applying this method over the incoming video frame in Figure 1(a). As one may see, only those pixels that resemble skin color have been admitted. In order to determine the exact face region from the image, we use simple blob operation (we chose the largest connected region as the face region). Figure 1(c) shows binarization of the skin pixels and Figure 1(d) shows the result of the entire face detection process after the blob operation is executed.

To determine the vertical position of the eyes on the face, we use the observation that eye-regions correspond to regions of high intensity gradients, as suggested in [13]. This method was implemented in [1,2] and it is preserved intact in this version of our system. We create a vertical gradient-map, $G(x,y)$, from the estimated region of the face. Any edge-detection method can be used. We chose to use a Sobel vertical edge kernel convolution. We then do a horizontal projection over $G(x,y)$ by summing the gray level of pixels along the horizontal rows:

$$H(y) = \sum_{i=1}^{xsize} G(i, y)$$

where $H(y)$ is the horizontal projection histogram value at pixel row y over the gradient image. Since both eyes are likely to be positioned at the same row, $H(y)$ will have a strong peak on that row. However, in order to reduce the risk of error, we consider the best three peaks in $H(y)$ for further search rather than the absolute maximum (see [2] for further explanation).

In order to further separate the two eyes within the predicted vertical stripe, we define a symmetry point where this separation will take place. Vertical symmetry position is computed as in [1] over the reduced image. It should be clearly noted that even though the symmetry calculation technique is exactly the same as the one described in [1] to

estimate the position and the limits of the face, symmetry (in this system) is only used to separate the left eye from the right eye.

After estimating the position of the eyes in the image and using the original 640x480 input frame, we use gray scale correlation over the eye region in order to find the exact position of the iris. Gray scale correlation is a pattern matching technique that allows us to search for a pattern in an image. Models were created and stored in a database using rectangular areas of well-known sample eye images from different persons and at different face angles (see Figure 3). A search is performed by assigning a match score to each pixel in the target image, based on how closely the model matches the region around that pixel.

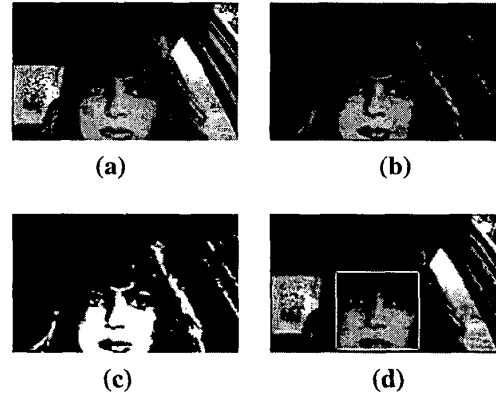


Figure 1. Detection of face using skin color information.

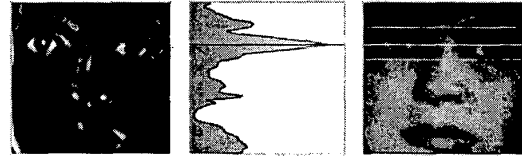


Figure 2. Vertical edge-map, edge projection, and estimated vertical position of the eyes.

The match F is computed as:

$$F = \frac{N \sum_{i=1}^N I_i M_i - \sum_{i=1}^N I_i \sum_{i=1}^N M_i}{\sqrt{\left[N \sum_{i=1}^N I_i^2 - \left(\sum_{i=1}^N I_i \right)^2 \right] \left[N \sum_{i=1}^N M_i^2 - \left(\sum_{i=1}^N M_i \right)^2 \right]}}$$

where N is the number of pixels in the model, M is the model, and I is the image against which the pattern is being compared.

We defined an acceptance level of match score above which a match is considered to be true. In other words, if the match score of a pixel is above the acceptance level, we consider that the iris of the eye is centered at that pixel point. If the match score is below the acceptance level, we conclude that the current pattern has no correspondence, and we use another model retrieved from the database. On the other hand, if a match is found, a new model is created from the live image and used in future pattern matching. We do this in order to avoid searching the database in each eye localization phase and to make the tracking process as smooth and accurate as possible.

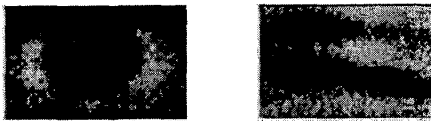


Figure 3. Sample model of open and closed left eye.

4.2. Tracking of the eyes

Tracking a feature in a sequence of images involves looking for that feature in a small neighborhood centered at the location of that feature in the previous frame. The main assumption is that the feature being tracked generally has small displacements between frames. In our case, we do not need to go through the entire process of localizing the eye in the following frames. If we assume that the eyes will not move very far between two consecutive frames, we can predict that the next eye match will happen almost certainly in an area surrounding the current iris position.

4.3. Finding tracking errors

In order to recover from tracking errors, we make sure that the distance between the eyes remains reasonably constant. We also restrict the eyes to be horizontally close to each other. If any of the geometrical constraints is violated, we re-localize the eyes in the next frame.



Figure 4. Eye tracking in progress.

5. The fatigue detector

As we had stated earlier in this paper, this system (as in [1,2]) detects micro-sleeps symptoms in order to diagnose driver fatigue. As the driver's fatigue increases, his/her eye blinks tend to last longer. We can determine his/her blink rate by counting the number of consecutive frames in which the eyelids remain closed. Our main problem was to differentiate among an open eye, a closed eye, or even the total absence of eyes in the given image. As the eye closes, we can be certain that the pattern-matching algorithm will fail to keep track of the eye. However, this notion should not always be interpreted as tracking error. It is very possible that a closed eye may have caused the recognition failure. Taking this into consideration, we have used a second model to keep track of the eye blinking. The model we have used is that of a closed eye, as shown in Figure 3, and the technique used is exactly the same as the one to find an open eye. If the tracking of the open eye fails, we try with the model of a closed eye. If none of these models seems to produce an acceptable match, we declare the tracking void and go back to the step relocating the eyes. Figure 5 shows how the system signals a "fatigue alert" when the eyes have been closed for several consecutive input frames (3-4 seconds).

6. Results and future work

We have tested the system with drivers of different skin color, with facial hair, and of different gender. We have monitored the system's response with different degrees of head rotation and tilt. The system is able to complete the eye localization at 10 frames per second and tracking at 15 frames per second. For small head-movements, the system rarely loses track of the eyes. The system has a tolerance on head rotation of up to 45 degrees and on tilt of up to 30 degrees. Under these circumstances, the system was able to detect prolonged eye blinks in 95% of the times and it produced occasional false alarms.



Figure 5. Fatigue signal activated.

We are planning to add the capability of auto-zoom on the eyes, once they are localized. This would address the issue between having a wide field of view in order to locate the eyes and a narrow field of view in order to detect fatigue.

To extend the capabilities of our system, we are currently working on the extraction of other facial features such as nostrils. Nostrils' positions when combined with the positions of the eyes can give more information on head rotation and tilt. This information would not only help to tolerate greater head rotation in the actual eye tracking and micro-sleep detection system, but would also help to determine bouncing movements of the head, another common symptom of driver fatigue.

7. Conclusions

We have presented a non-intrusive real-time eye tracking system. The system is able to localize and track the pupil of a driver as soon as he/she sits in front of the camera. The system uses a skin color based approach to locate the face and a gray scale-based eye feature extraction. During tracking, the system is able to automatically detect any tracking error that might have occurred. In case of a tracking error, the system is able to recover and resume the proper tracking. The system is able to automatically diagnose fatigue by monitoring the eyes for micro-sleeps. This is achieved by counting the number of consecutive frames in which the eyes are found to be closed.

Acknowledgements

We would like to thank Steve Hay (CTS) and Ron Cassellius (MNDOT) for their support. This work has been supported financially by the ITS Institute, University of Minnesota and the Minnesota Department of Transportation.

References

- [1] Eriksson, M., and Papanikolopoulos, N. (1998). "A vision based system for the detection of driver fatigue," in Proceedings of the ITS America Eighth Annual Meeting, 1998.
- [2] Eriksson, M., and Papanikolopoulos, N. (1997). "Eye-tracking for detection of driver fatigue," in Proceedings of the IEEE International Conference on Intelligent Transportation Systems, 1997.
- [3] US. DOT. FHWA (1996). "Commercial motor vehicle driver fatigue and alertness study". FHWA report number: FHWA-MC-97-001, TC report number: TP 12876E, November 1996.
- [4] Yoo, T.W., and Oh, I.S. (1996). "Extraction of face region and features based on chromatic properties of human faces," Pacific Rim International Conference on Artificial Intelligence, pp. 637-645.
- [5] Pentland, A., Moghaddam, B., and Straner, T. (1994). "View-based and modular eigenspaces for face recognition," CVPR '94, pp. 84-91.
- [6] Sobottka, K., and Pitas, I. (1996). "Segmentation and tracking of faces in color images," Proceedings of the Second International Conference on Signals, Sys. And Comp. pp 236-241.
- [7] Yuille, A.L., Cohen, D.S., and Hallinan, P.W. (1989). "Feature extraction from faces using deformable templates," in proceedings of CVPR, pp.104-109.
- [8] Yang, J., Lu, W., and Waibel, A. (1998). "Skin-color modeling and adaptation" Proceedings of ACCV'98, vol. II, pp. 687-694 (Hong Kong).
- [9] Yang, J., Lu, W., and Waibel, A. (1996). "A real-time face tracker," Proceedings of WACV'96 (Sarasota, Florida, USA).
- [10] Pomerleau, D., and Baluja, S. (1993). "Non-intrusive gaze tracking using artificial neural networks", AAAI Fall Symposium on Machine Learning in Computer Vision, Raleigh, NC.
- [11] Brunelli, R., and Poggio, T. (1993). "Face recognition: features versus templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No 10, pp. 1042-1052.
- [12] Chow, G., and Li, X. (1993). "Towards a system for automatic feature detection," Pattern Recognition, Vol. 26, No. 12, pp. 1739-1755.
- [13] Stringa, L. (1993). "Eyes detection for face recognition," Applied Artificial Intelligence, No 7, pp. 365-382.
- [14] Cox, I.J., Ghosn, J., and Yianilos, P.N. (1995). "Feature-based recognition using mixture-distance," NEC Research Institute, Technical Report 95 - 09.
- [15] Craw, I., Ellis, H., and Lishman, J.R. (1987). "Automatic Extraction of Face-Features," *Pattern Recognition Letters*, 5, pp. 183-187.