VISION-BASED METHODS FOR DRIVER MONITORING

Eric Wahlstrom, Osama Masoud, and Nikos Papanikolopoulos, *Senior Member, IEEE*
Department of Computer Science and Engineering, University of Minnesota, USA

***Abstract*— This project involves the use of a dashboard-mounted camera to monitor the direction a driver is looking. This is accomplished by using the Framework for Processing Video (FPV), developed at the University of Minnesota by Osama Masoud. The monitoring software works by first finding the lips on the driver's face using color analysis. Then, the skin color of the driver can be sampled and a face region is generated. The largest holes in this region are the lips and eyes. Once the eyes are found, the software finds the darkest pixels in the eyes and marks these as the pupils. The software uses the relative positions of the eyes and pupils to make statements about the gaze direction. Also discussed is monitoring various other driver activities such as adjusting a car stereo.**

***Index Terms*—Monitoring Driver Activities, Computer Vision, Gaze Direction**

I. INTRODUCTION

Driver monitoring is a growing area of interest in computer vision. Researchers are using vehicle-mounted cameras to measure some aspects of the driving experience. Most of this research has focused on detecting signs of fatigue. Detecting driver fatigue is done primarily by tracking the driver's eyes and detecting blinks. Long blinks are signs of a driver experiencing fatigue. Several robust systems have been developed and are explained in [1], [2], [3], and [4].

Beyond detecting fatigue, an important role of driver monitoring is detecting driver distraction. Distraction qualifies as any behavior on the part of the driver that negatively affects driving performance. Activities like adjusting the radio and talking on the cell phone are obviously distractions. Less obvious distractions can be objects outside the vehicle like pedestrians and billboards. An interesting system for predicting driver distraction based on cognitive tasks is described in [5]. Research in this area indicates that many objects in a vehicle are more distracting than some might think. In [6], researchers describe how distracting a car stereo can be.

To aid detection of driver distraction, driver monitoring can encompass tasks such as detecting where the driver is looking and what the driver is doing. By watching the direction a driver is looking, one could potentially determine dangerous driver behavior like gaze fixation on the dashboard or changing lanes without checking mirrors. By watching the activities of the driver, one can determine if the driver is talking on a cellular phone or if he/she is adjusting the radio. These activities could constitute dangerous distractions for a driver. These types of driver monitoring have not been as heavily researched as detecting driver fatigue.

A critical piece of all driver monitoring systems is the method for finding the driver's eyes and tracking them. In [7], a system for finding eyes is described that uses infrared light bursts to find the eyes. This system is robust, but requires a special camera. In [8], researchers use color predicates to find a skin region and use the holes in that region to find the eyes. This method works well and requires no special equipment. Once the eyes and other facial features have been detected, there is some work to be done to determine the gaze direction. A mathematical model is described in [9] that allows a three-dimensional model of the driver's gaze to be determined. In [10], a good gaze direction system is described that does not depend on previous knowledge of the vehicle layout.

Beyond the field of computer vision, there are other interesting methods for driver monitoring. In [11], a method for measuring the workload a driver is dealing with is described. This system uses the motions of the steering wheel to make statements of how quickly a driver is responding to traffic. Other similar systems could be integrated into an automobile for fully monitoring the interactions between the driver and the vehicle.

It is the purpose of this work to develop systems for monitoring both driver gaze direction and driver activities using vehicle-mounted cameras. These systems could be used to study how drivers behave in real driving environments.

II. DESCRIPTION OF THE PROBLEM

The tasks undertaken for this project are two-fold: (i) tracking a driver's gaze to determine the current focus and (ii) monitoring the activities of the driver to find specific behaviors. Each of these tasks offers challenges of its own. In order to determine gaze direction, one needs to find the driver's eyes. There are various methods to do so and each offers its own benefits. Also, finding gaze direction can require knowledge of the positions of other parts of the driver's face and head.

The activity-monitoring component of this project is also difficult. Instead of using a head-only camera angle, this component used a side-view. A side-view of a human body presents many more inconsistencies than a head-only view. For example, if one wishes to track the driver's hand, one must determine some criteria for tracking the hand in all its possible configurations. Also, the behaviors to be detected must be well defined in order to be detected. For this project, it was decided to detect a sample activity and show the potential for detecting further activities.

III. Detecting Gaze Direction

A. Software Framework

This project was developed using Visual C++ and an application called Framework for Processing Video (FPV), developed at the University of Minnesota by Osama Masoud. This application is a movie viewer that allows frame-by-frame processing of images. It was decided to use this software because it allows easy access to movies without the need to learn a specific camera and hardware configuration. One limitation of the software is that it does not currently support live video processing.

B. Software Design

The first step taken to detect gaze direction was the detection of the driver's lips. This was accomplished by finding the largest region in the image that fits a color predicate. This predicate is a range of color values that correspond to human lips. Once the lip region is found, the corners of the mouth are stored to later find the gaze direction.

In order to find the eyes, the software uses the size and position of the mouth. It searches for the eyes in an area three mouth-widths wide and between one and two mouth-widths above the top of the mouth region. The eyes are found by searching this area for dark regions. If the two largest regions are symmetrically positioned along the x-axis, then the eyes have been found. Then, by using this rough guess, the pixel center is found by selecting the darkest pixel in the eye region. Now, a corner of each eye is found by searching between the two pupils for a dark pixel.

The above process is used for initially finding the mouth and eyes. However, once the first frame is processed, these features can be easily tracked. A simple tracking algorithm was used for tracking the mouth. Using the previous three frames, an average change of position is found for each feature point of the mouth. The software finds the mouth as normal, but then this is limited by the previous average change in position. The farther the new change is from average, the more it is weighed toward the average. This means that changes of twice the average or larger are instead made equal to the average. The result of this mouth tracking is a nearly constant mouth size.

Because of the mouth tracking, the eye positions can be found accurately enough that eye tracking proved minimally important. An attempt was made to track the eyes by detecting large changes in feature position and seeking the closest dark pixel to the previous feature position. This method shows no difference from normally finding the eyes based on mouth position.

Once the eye corners and pupil centers are known, the gaze direction can be calculated. At this stage in the project, it was decided that a 3D construction of the gaze direction is not necessary for estimating the target of the gaze. This is based on the fact that relatively few targets are important (dashboard, windshield, side mirrors, and rear mirror) and these gaze trajectories can be found without complicated calculations.

In order to determine the gaze of the subject, average distances between the pupil and eye corners are used. Each set of distances represents a gaze position. Whichever gaze position has the least variance from the detected positions is said to be the detected gaze direction. Then, a small amount of tracking is done to avoid noise in the results. In cases where the current gaze number differs from the previous direction, the previous will be used if its position distances are similar to those found.

IV. Detecting Driver Activities

For this project, it was determined to initially detect any activity that involved the car stereo. To do this, a region of interest is defined in the source images. This region is bounded around the car stereo in the image. Then, if this region is significantly disturbed, the activity is "detected." To determine if the region is disturbed, the software watches the average pixel values in the region. When the region is disturbed, the average pixel value rises. This is caused by color difference between the dark radio and the human skin. At each frame, if a disturbance is detected, the system signals that the radio is being adjusted.

In some cases, the driver may be performing some activity that results in the stereo region being disturbed, but he/she is not adjusting the stereo. If this is detected as a stereo activity, this would be a false positive. With the current software, false positives also occur when a sunspot drifts into the radio bounding box due to movement of the camera.

V. Issues and Challenges

A. Framework Limitations

One challenge that arose early in this project was expanding the FPV software to allow complicated manipulation and analysis of source images using common computer vision algorithms. The software does not provide these functions itself, so work was done early on to implement them. It was quickly determined that this would take far too long, so a new solution needed to be found. Instead, we used OpenCV and IPL, open source image tools developed by Intel Corporation, to process the images. These packages provided all the functions necessary to complete the project.

B. Automatic Initialization

Another challenge that was found early was that of automatically initializing the software. Many computer vision tools rely on a hands-on initialization phase for determining feature points in the image. We wanted this software to work without such a step, so color was chosen as the basis for selection. We found a range of lip colors that allowed quick retrieval of the mouth region without finding too much noise in the image. After the mouth was found, the project originally tried to find the eyes by testing the pixels near the mouth for color. Then, the software created a face region by finding colors similar to those around the mouth. In [10], this allowed the researchers to find the eyes by looking at the holes in the region. In our project, the face region was not useful. Instead of a nicely connected region, the process resulted in a region that was full of holes wherever shadows or specular spots occurred. To fix this problem, we chose to guess at the eye locations based on the mouth width.

VI. Results

A. Gaze Direction

The software described above accurately finds a driver's gaze approximately 85% of the time. This is based on a single sample movie with a driver gazing in all the desired directions. The movie was taken outdoors under realistic lighting conditions. The main challenge of this movie is that the driver's face has shadows and specular white spots that the software must deal with.

Figure 1 shows the driver looking at the rear view mirror and the eyes and mouth are being tracked. "X"s can be seen on the image in the locations that are found.

While the application runs, a gaze history is displayed in a window above the video. This history shows the gaze direction that is detected plotted over time. Each colored segment on the history refers to a specific gaze direction. Figure 2 shows a close up of the gaze history.

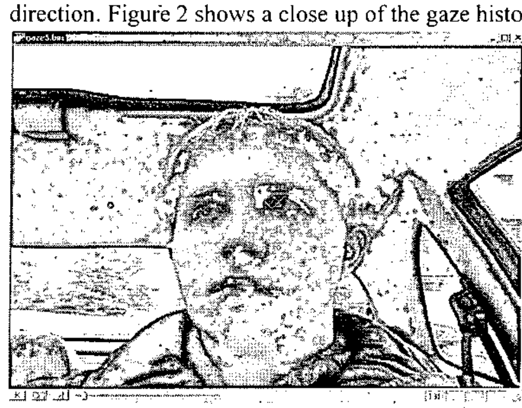

Fig. 1. Driver features being tracked.

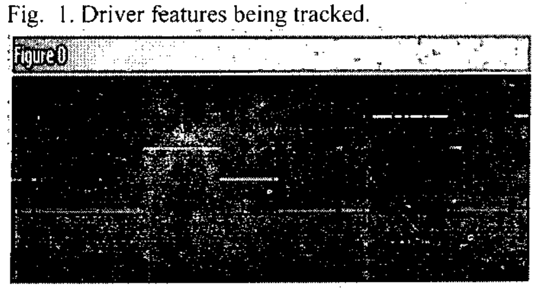

Fig. 2. Close up of gaze history.

Fig. 3. Failing to find gaze.

The 15% failure rate of this application was due to two types of errors. First, the application may fail to find the eyes at all. Second, the application may have detected the gaze direction incorrectly. Most of these errors occurred rarely and only for one or two frames in a row. The most significant error, which lasted for 39 frames and accounted for 6 of the 15%, occurred when the driver turned his head to look at the passenger side mirror. First, the software detected the gaze correctly, then due to tracking errors, it began to miscalculate the gaze. Finally, it could not track the driver's gaze at all. Figure 3 shows the driver gazing at the passenger side mirror and the software failing to find his gaze correctly.

B. Activity Monitoring

The activity monitoring software successfully detects when the driver manipulates the radio in every case when more than just a single finger enters the radio area of the image. Figure 4 shows the driver adjusting the radio and the software indicating this by drawing an "X" over the radio bounding box. In Figure 5, the driver is adjusting the radio but the software does not detect the event because the disturbance in the bounding area is too low. Over the course of the test video, only a single false positive detection occurs. This occurred when the radio in the image shifted and a bright space appeared in the bounding area.

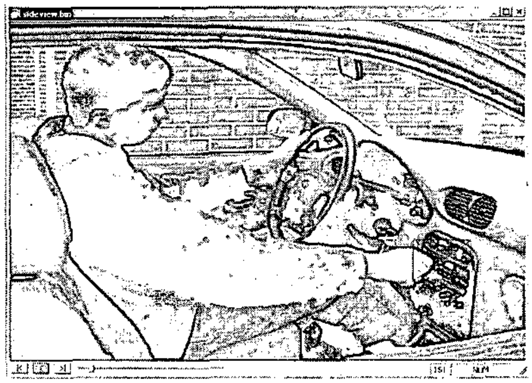

Fig. 4. Radio activity is detected.

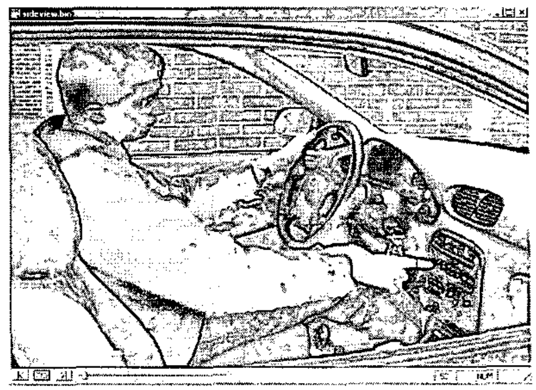

Fig. 5. Failing to find radio activity.

VII. Future Work

A. Storing Gaze Positions

A possibility for expanding this work is shown in [7], which describes a method for storing gaze positions in a histogram. The peaks in the histogram are assumed to be the desired gaze targets like the mirrors and the windshield. This process allows for differences in camera placement and driver height because the hard-coded gaze locations are no longer necessary. A drawback of this option would be the real-time usability of the system. If the software must wait until after the driver is done driving to report gaze history, it would be nearly useless. If instead the drivers had to perform some initialization steps each time they started the system, this would be more useful, but most drivers would be unlikely to use the system at all.

B. Robust Eye Tracking

Another possibility is to enhance the current eye tracking so that large turns of the head will not cause the system to fail to find the gaze direction. To aid this, a robust eye tracking module could be added. A robust eye tracker is described in [8] that uses a multi-phase design to track the eyes. Each phase tries to find the current position of the eyes based on the previous locations. If a phase fails, it passes the process along to the next slower but more robust phase. If the tracker fails entirely, the software simply uses the original eye finding routine. The current version of the software does not use any eye tracking because it was not needed 90% of the time.

C. Expanded Activity Monitoring

The next phase of this project will attempt to expand the activity monitoring software. Currently, one single activity is detected. In future releases of this software, it should be possible to categorize driver behaviors and track them over time similarly to the history that is displayed for gaze monitoring. Additionally, enhancements need to be made to the activity monitoring software. At the current time, the video used to detect activities is taken from a camera positioned outside the vehicle. This has led to a changing position of the radio in the image. The result is that the bounding area of the radio is much larger than the radio itself and the monitoring software is less sensitive than it could be. The first step to fix this issue is to take video of the activity with a firmly mounted camera. In the next generation of this software, an omni-directional camera will be mounted to the ceiling of the vehicle. This camera would then move with the vehicle and allow a variety of activities to be detected.

D. Detecting False Activity

A further enhancement that would be made possible by a firmly mounted camera is the identification of false positives in activity detection. To eliminate some cases of false positives, the software will check to see if too many pixels in the region of interest are affected. For example, if the driver reaches past the radio to open the glove box, every pixel in the region might be changed drastically. Because adjusting the stereo is unlikely to cause such a drastic change, the software notes the occurrence but states the strong possibility of a false positive. Cases of sunspots, like the one occurring in the test video, will be eliminated because the radio bounding region will be smaller and will not drift.

VIII. Conclusion

The application developed for this project successfully finds the gaze direction of a driver 85% of the time. When errors did occur, the system never failed to find the gaze direction again and continue on. The software reports a history of the driver's gaze in real-time. This tool could easily be used for a driver warning system similar to those in most fatigue detection systems. Whenever a driver's gaze seems fixated on a position other than the road, the system could warn the driver to pay attention.

A. Limitations

The system does fail to detect the driver's gaze if the driver's head is turned to either side. This means that whenever a driver turns his/her head to view out either side windows, the software cannot find the driver's eyes. However, the software could still warn the driver in such a situation. Because the driver is not watching the road, any gaze fixation that occurs to either side is potentially dangerous. Since the system can find the drivers' gaze when their head is forward, a long period of non-detection is a strong sign of dangerous driving.

B. Key Successes

While this system does have limitations, two key successes were made. First, the software was built on an application that allows easy entry for developers who are not familiar with graphics hardware. The FPV software provides an easy-to-use, Windows-based shell that programmers can use to implement vision applications. This project will serve as an example for future researcher efforts wishing to work on the FPV platform.

Second, this application succeeds in finding gaze direction in realistic lighting conditions. Most computer vision applications are tested in strict lighting conditions that eliminate most shadows and provide no bright specular spots in the images used. This research not only addressed such conditions but also used shadows as visual cues to find the corners of the eyes.

Despite its limitations, this process shows great promise for detecting what goes on in a moving vehicle. Using an application like this, researchers can design vehicles and components that minimize the danger involved in driving.

IX. Acknowledgements

This work was supported in part by the Minnesota Department of Transportation, the ITS Institute at the University of Minnesota, and the National Science Foundation through grant #0219863.

REFERENCES

[1] M. Eriksson and N. Papanikolopoulos. "Driver fatigue: a vision-based approach to automatic diagnosis," Transportation Research Part C-Emerging Technologies, 9:399-413, 2001.

[2] R. Grace, V. Byrne, D. Bierman, J-M Legrand, D. Gricourt, B. Davis, J. Staszewski, and B. Carnahan. "A drowsy driver detection system for heavy vehicles," 17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference Proceedings. 2:1-8, 1998.

[3] N. Kojima, K. Kozuka, T. Nakano, and S. Yamamoto. "Detection of consciousness degradation and concentration of a driver for friendly information service," Proceedings of the IEEE International Vehicle Electronics Conference, 31-36, 2001.

[4] S. Singh and N. Papanikolopoulos. "Monitoring driver fatigue using facial analysis techniques," Proceedings 1999 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, 314-318, 1999.

[5] D. Salvucci. "Modeling driver distraction from cognitive task," Proceedings of the 24th Annual Conference of the Cognitive Science Society, 2002.

[6] H. Thimbleby, P. Duquenoy, and G. Marsden. "Ethics and consumer electronics," Proceedings of the 4th ETHICOMP International Conference on the Social and Ethical Impacts of Information and Communication Technologies, 1999.

[7] A. Haro, M. Flickner, and I. Essa. "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance," Georgia Institute of Technology, Atlanta, Georgia. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2000.

[8] P. Smith, M. Shah, and N. da Vitoria Lobo. "Monitoring head/eye motion for driver alertness with one camera," Proceedings of the International Conference on Pattern Recognition (ICPR'00), Session P4.3A, 2000.

[9] A. Gee and R. Cipolla. "Determining the gaze of faces in images," Image and Vision Computing, 30:639-647, 1994.

[10] R. Pappu and P. Beardsley. "A qualitative approach to classifying gaze direction," Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 160-5, 1998.

[11] O. Nakayama, T. Futami, T. Nakamura, and E. Boer. "Development of a steering entropy method for evaluating driver workload," SAE Technical Paper Series: #1999-01-0892: Presented at the International Congress and Exposition, 1999.