

Determining Driver Visual Attention with One Camera

Paul Smith, *Student Member, IEEE*, Mubarak Shah, *Fellow, IEEE*, and Niels da Vitoria Lobo, *Member, IEEE*
{rsmith,shah,niels}@cs.ucf.edu

Abstract—This paper presents a system for analyzing human driver visual attention. The system relies on estimation of global motion and color statistics to robustly track a person's head and facial features. The system is fully automatic, it can initialize automatically, and reinitialize when necessary. The system classifies rotation in all viewing directions, detects eye/mouth occlusion, detects eye blinking and eye closure, and recovers the 3D gaze of the eyes. In addition, the system is able to track both through occlusion due to eye blinking, and eye closure, large mouth movement, and also through occlusion due to rotation. Even when the face is fully occluded due to rotation, the system does not break down. Further the system is able to track through yawning, which is a large local mouth motion. Finally results are presented, and future work on how this system can be used for more advanced driver visual attention monitoring is discussed.

Index Terms—Automatic vision surveillance, driver visual attention monitoring, driver activity tracking, in-car camera systems

I. INTRODUCTION

According to the U.S. National Highway Traffic Safety Administration, approximately 4,700 fatalities occurred in motor vehicles in the year 2000 in the US alone due to driver inattention, driver fatigue, and lack of sleep [1]. Of these about 3,900 were from inattention and about 1,700 were from drowsiness, fatigue, illness, or blackout. Automatically detecting the visual attention level of drivers early enough to warn them about their lack of adequate visual attention due to fatigue or other factors may save US tax payers and businesses a significant amount of money and personal suffering. Therefore, it is important to explore the use of innovative technologies for solving the driver visual attention monitoring problem. A system for classifying head movements and eye movements would be useful in warning drivers when they fell asleep. It could be used to both gather statistics about a driver's gaze and monitor driver visual attention.

This paper describes a framework for analyzing video sequences of a driver and determining the visual attention of the driver. The system does not try to determine if the driver is daydreaming and thus not paying adequate attention to the road, which is an example of cognitive underloading. In this case the driver is looking straight ahead and appears to be fully alert. Other methods will need to be developed to detect these kinds of situations. The proposed system deals with the strictly measurable quantifiable cues like eye blink rate, and head rotation rate. The system collects data with a single camera placed on the car dashboard. The system focuses on rotation of the head and eye blinking, two important cues for determining driver visual attention, to make determinations of the driver's visual attention level. Head tracking consists

of tracking the lip corners, eye centers, and sides of the face. Automatic initialization of all features is achieved using color predicates[2] and the connected components algorithm. Occlusion of the eyes and mouth often occurs when the head rotates or the eyes close, and the system tracks through such occlusion and can automatically reinitialize when it mis-tracks. Also, the system performs blink detection, eye closure detection and determines 3-D direction of gaze. 3-D gaze information can be used for hands free control of devices like the radio or cd player.

The proposed system initializes automatically, tracks, and determines visual attention parameters like orientation of face. A moving vehicle presents new challenges like variable lighting and changing backgrounds. The system was tested in both stationary and moving vehicles with negligible differences in the accuracy of the system. With moving vehicles the system did not encounter any difficulty with the changing background. Further the system has performed well under a wide variety of daytime illumination levels, from strong daylight, to heavy cloud cover, to a half of the driver's face in sunlight and the other in shadow.

The results also show that the system can track through local lip motion like yawning. The work in [3] presented a robust tracking method of the face, and in particular, the lips, but this paper shows that the work in [3] can be extended to track during yawning or opening of the mouth.

The organization of the paper is as follows. First, the paper will discuss previous work and then describe the tracking system in detail. Next, occlusion work is presented. This entails occlusion of the eyes due to the rotation of the face. Next, 3D gaze reconstruction work is presented. After this, the details of the automated driver visual attention classification system are given, then the paper gives quantitative results, discusses future aspects of driver visual attention monitoring, and concludes.

A. Input Data

The video sequences were acquired using a video camera placed on the car dashboard. The system runs on an UltraSparc using 320x240 size images with 30 fps video, but it can be easily implemented on a PC laptop. Eight drivers were tested under different daylight conditions ranging from broad daylight to parking garages. Some of the sequences were taken with a lot of cloud cover so that the lighting was lower than other daytime conditions. Some of the sequences have partially illuminated faces (half of the face in sunlight and half in shade). The system is not yet tested on night driving

conditions. In parking garages and other low illumination levels the system works. The system has strategies for dealing with nighttime driving, but this work has not yet been fully tested. Some video sequences were recorded in moving vehicles and others in stationary vehicles. All sequences were recorded in real vehicles. Overall the method has been tested on about 7100 frames of video data, about four minutes, with eight different drivers. Nineteen sequences were collected. The drivers were selected from the graduate and undergraduate students working in our university computer science research labs. No particular screening was done. Various drivers were selected with many different skin tones and facial shapes and sizes. Drivers were not picked with regard given to whether the system would track well or not.

II. PREVIOUS WORK

Much terminology has been introduced in the driver vigilance and attention monitoring fields. In particular [4] lays a terminology groundwork, and there are others who use similar terminology. In our paper similar terminology is used. Visual attention refers to whether the driver is visually looking forward and alertness/drowsiness refers to whether the driver is fatigued and possibly fighting against sleeping bouts, microsleep, or other similar conditions.

Work on driver alertness [5] [6] [7] [8] [9] [10], has yielded many systems. However, it is becoming clear that more than alertness needs to be monitored [11]. With new technologies becoming more a part of every day life, drivers need to be careful to make sure they are paying adequate visual attention to the road. Therefore methods must be developed which monitor both drowsiness and visual attention. In the case of a decrease in visual attention, the driver may be fully awake, yet still not paying adequate visual attention to the road. Relying solely on determining if the eyes were open would not be enough in the event the driver was not drowsy, but was instead simply looking off center for a while. In this case the eyes would be open, yet the driver could possibly have a low visual attention level. More than eye closure metrics must be used in this case. Detecting rotation can play an important part in detecting a decrease in visual attention. Various classes of systems have emerged to determine driver drowsiness and attention levels. Some systems [12] [13] rely on external car behavior like the distance to roadway lines. Others [14] are trying to use infrared beam sensors above the eyes which detect when the eyelids interrupt the beam, and the system will measure the time that the beam is blocked, thus providing eye closure information. Another class that has emerged is the one in which data is acquired from visual sensors [11] [15] [16]. An important aspect of these systems is that unlike infrared beams and the necessary hardware the user must wear, these are simple to install and are non invasive.

To monitor driver visual attention or alertness a head tracking method must be developed. Several researchers have worked on head tracking [17] [18], and the various methods each have their pros and cons. Among more recent methods to track facial features Huang and Marianu [19] present a method to detect the face and eyes of a person's head. They

first use multiscale filters like an elongated second derivative Gaussian filter to get the pre-attentive features of objects. Then these features are supplied to three different models to further analyze the image. The first is a structural model that partitions the features into facial candidates. After they obtain a geometric structure that fits their constraints they use affine transformations to fit the real world face. Next their system uses a texture model that measures color similarity of a candidate with the face model, which includes variation between facial regions, symmetry of the face, and color similarity between regions of the face. The texture comparison relies on the cheek regions. Finally they use a feature model to obtain the location of the eyes. Their method uses eigen-eyes and image feature analysis. Then they zoom in on the eye region and perform more detailed analysis. Their analysis includes Hough transforms to find circles and reciprocal operations using contour correlation.

Shih, Wu, and Liu [20] propose a system using 3D vision techniques to estimate and track the 3D line of sight of a person using multiple cameras. Their approach uses multiple cameras and multiple point light sources to estimate the line of sight without using user-dependent parameters, thus avoiding cumbersome calibration processes. The method uses a simplified eye model, and it first uses the Purkinje images of an infrared light source to determine eye location. When light hits a medium part is reflected and part is refracted. The first Purkinje image is the light reflected by the exterior cornea [20]. Then they use linear constraints to determine the line of sight, based on their estimation of the cornea center.

Terrillon *et al.* [21] use Support Vector Machines (SVMs) to solve the pattern recognition problem. SVMs are relatively old, but applications involving real pattern recognition problems are recent. First they do skin color-based segmentation based on a single Gaussian chrominance model and a Gaussian mixture density model. Feature extraction is performed using orthogonal Fourier-Mellin moments. Then they show how, for all chrominance spaces, the SVMs applied to the Mellin moments perform better than a 3-layer perceptron Neural Network.

In [22] a lip color based approach is used to find the lip colors. They also use dynamic thresholds and a voting system to robustly find the lips. Then the 3D mouth height is computed, which allows the system to determine if the mouth is open or not. The method is stereo based, and relies on images being well lit in a controlled environment. In [23] the above feature point extraction method is evaluated for accuracy. This differs from the approach proposed in our paper because they rely on a well lit image, which makes lip identification much easier than with our unconstrained daytime driving illumination conditions.

In [24] a method is presented which tracks the head and estimates pose. It relies on 2D template searching, and then 3D stereo matching. A 3D model is then fit and minimized using virtual springs, which is simpler than the least squares fit approach. Manual initialization is required to build the facial feature model, which can be a cumbersome burden.

In [25] the method presented is a stereo based system that matches specific features from left and right images to

determine the 3D position of each feature. A least squares optimization is done to determine the exact pose of the head. Eye processing locates the iris and then combines eye-gaze vectors for each eye with the head pose to determine eye-gaze direction. The system relies on manual initialization of feature points. The system appears to be robust but the manual initialization is a limitation, and it makes trivial the whole problem of tracking and pose estimation. Their system requires roughly five minutes of manual initialization and calibration, evidence that manual initialization methods create many burdens and limitations. Our method requires no manual initialization, which is an attractive feature of any system.

The method presented in our paper has no manual initialization after the offline building of the color predicate and hence has no limitations presented by manual initialization. Further while stereo helps, it adds additional calibration and hardware constraints which a single camera system, such as the one presented in this paper, does not have. Our paper does not try to supplant stereo vision systems as they have some advantages, but the aim here is to advance the state of the art of single camera systems.

In [6] a method is presented to determine driver drowsiness in which each eye's degree of openness is measured. The method first detects the eyeball using preprocessing to isolate the eyes and other dark regions. Then a labeling operation is done to find the eyes. Subsequent tracking is performed and the relative size of the eyes are recorded and learned by the system. Then the system detects eye closures based on the degree of eye openness. The system will reinitialize if the eyes are not being tracked properly. In their work, visual attention is not monitored. They only monitor drowsiness. Further it is not clear how foreshortening of the eyes is taken into account, which will affect the degree of eye openness calculations. Partial occlusion of the facial features during rotation will affect the labeling operation, so the system seems to work for images in which the driver is looking relatively straight ahead. Highly accurate and robust identification is necessary to adequately cope with real world driving conditions.

In [5] a system is proposed using multiple cameras, one with a view of the whole face, and one with a view of the eyes only. Their idea is to move the eye camera on the fly to get the best image of the eyes, since their technique uses eye information like blink frequency. They use LEDs to minimize problems with lighting conditions. Their face detection algorithms are simple because they assume a simple background and limited face movement. Because of the LED illumination the method can easily find the eyes and from there the system finds the rest of the facial features. The next step uses ellipses to model the head and searches for the head in the next frame. They investigate model based approaches for facial pose estimation and discuss using the distance between the pupil and brightest part of the eye to determine facial orientation. To get a more accurate estimation they propose to analytically estimate the local gaze direction based on pupil location. LED illumination means additionally complicated hardware, and this is acceptable for some situations like night time driving, but in daytime driving conditions other possibilities exist which do not need to rely on LED illumination. Also, it is reported in [11] that

LED illumination methods have many problems in daytime driving largely because of ambient sunlight, which makes these methods much less usable in daytime conditions.

In [26] [27] and [28], a multiple camera method is presented that uses infrared LED illumination to find the eyes. A simple subtraction process is done, to find the bright spots in the image. Then, a segregation routine clusters the bright pixel regions together to determine if the cluster is an eye or noise. Then eye extraction is performed using size constraints. PERCLOS, a validated drowsiness metric, is computed by taking the measurement of the eye with the largest degree of openness. This system currently works under low light conditions. Many of the PERCLOS based systems using infrared illumination work under low light conditions only, as noted in [11].

A more detailed analysis of the strengths and weaknesses of the PERCLOS measurement can be found in [11], [15], and [29]. The method presented in our paper tries to address other driver metrics that PERCLOS does not measure, such as decreased visual attention from looking away from straight ahead for some length of time. The attempt in our paper is to address decreased vigilance in the following way: no claim is made to supplant the PERCLOS measurements as a valid physiological metric, but rather to show that the data collected can be used to measure driver drowsiness in a way similar to [6] but without the eyeball size estimation. As well, the methods presented to detect eye closure are sound and contribute to the general problem of detecting eye closures in moving vehicles. Our paper also shows and provides a detailed framework for acquiring metrics and measuring the driver's visual attention level. This was done because more accidents were reportedly caused by inattention than decreased vigilance [1].

III. THE OVERALL ALGORITHM

An overview of the algorithm is given below. In the following sections each step is discussed in detail.

- 1) Automatically initialize lips and eyes using color predicates and connected components
- 2) Track lip corners using dark line between lips and color predicate even through large mouth movement like yawning
- 3) Track eyes using affine motion and color predicates
- 4) Construct a bounding box of the head
- 5) Determine rotation using distances between eye and lip feature points and sides of the face
- 6) Determine eye blinking and eye closing using the number and intensity of pixels in the eye region
- 7) Reconstruct 3D gaze
- 8) Determine driver visual attention level using all acquired information

IV. INITIALIZING LIP AND EYE FEATURE POINTS

Initialization employs color predicates [2]. For the lips, a specialized lip color predicate is used that identifies lip color in images. The system looks for the largest lip colored region using the connected component algorithm. Figure 1 shows these lip color predicates. The other training images are similar. For the eye initialization, a similar idea with a

different color predicate is used as follows. First, the skin color predicate is built, which segments the skin from non-skin regions. Since eyes are not skin, they always show up as holes. Hence, connected components of non-skin pixels are possible eye holes. The system finds the two holes that are above the previously found lip region, and that satisfy the following size criteria for eyes. Since the dashboard camera is at a fixed distance from the face, the relative size of eyes is estimated to be between .02 and 1 percent of the area of the image. For all images tested (several thousand), these criteria were reliable. For our experiments each driver had one lip and one skin color predicate. Drivers appearing in multiple data sets did not have separate predicates for each sequence.



Fig. 1. Lip color predicate training. The first image is the original image, the second shows the manually selected skin color regions, and the third shows automatically detected lip regions.

Figure 2 shows an input image, manually selected skin region, and the output of the color predicate program on the input image. Figure 3 shows results of automatic eye and lip initialization from various data sets.



Fig. 2. Skin color predicate training. The first image is the input image, the second shows the manually selected skin regions, and the third shows automatically detected skin regions.



Fig. 3. Results from automatic eye and lip initialization

V. FEATURE TRACKING

A. Hierarchical Lip Tracking

The lip tracking system is designed as a multi-strategy lip tracker. The first strategy is the most accurate but most unstable and is used if it gives a correct estimate for the lip corners. The second strategy is not as accurate but more stable and is used if the first strategy gives a bad estimate.

The third strategy is coarse but very stable and is used if the other strategies fail. In parallel, a fourth strategy, the affine transformation of small windows around the lip corners is computed and the old lip corner is warped to get the new lip corner using [30]. This step produces a more stable value for the lip corners and is used, but is periodically updated by the first three strategies' estimate.

For the first strategy, the dark line between the lips is automatically found as follows (shown in Figure 4 as a white line): The center of the lips will be $\frac{PC_1 + PC_2}{2}$, where PC_1 is the previous left lip corner and PC_2 is the previous right lip corner. For each column extending beyond both mouth corners, consider a vertical line (of height 15 pixels) and find the darkest pixel on this vertical line, by using $\min \frac{R+G+B}{3}$. The darkest pixel will generally be a pixel in the gap between the lips. To determine where the lip corners are the system obtains

$$f(x, y) = \frac{1}{D(x, y)} + \frac{1}{I(x, y)},$$

where $D(x, y)$ is distance of a pixel (x, y) from the closest corner of the mouth, and $I(x, y)$ is the intensity at (x, y) . This will give a pixel that is close to the previous lip corner, and that is not too bright. The function maximum is the lip corner.

If this estimate is too far from the previous lip corner, the second strategy for lip tracking, described next, is run. Abrupt lighting changes or large head motion will cause the intensity approach to fail. For the second strategy, each darkest pixel is checked to see if there is lip color above and below it. The search starts with the pixel closest to the previous lip corner and goes outward. Once there are no lip colored pixels above and below each point, this is considered the lip corner. If the second strategy fails then the third strategy is to reinitialize the system, as described above in section IV, running the connected component algorithm in the region around where the lips were last.



Fig. 4. Example of dark line between lips

B. Hierarchical Eye Tracking

Eye tracking is done in a multi-strategy way, similar to the multi-strategy lip tracker. First the system uses intensity information of the eye region to find eye pupils. If necessary the system employs skin color information to see where the eyes are. In parallel the third strategy uses an affine motion estimate to map small windows around the eyes in the current frame to the next frame.

The affine transformation computation in [30] is less likely to break down during heavy occlusion. The affine tracker is not as accurate as the eye black hole tracker, because of the interpolation step involved in affine tracking during warping, which is why it's not used exclusively.

For the first strategy, the system searches around the eye center in the previous frame and locates the center of mass of the eye region pixels. Then a small window is searched around the center of mass to look for the darkest pixel, which corresponds to the pupil. If this estimate produces a new eye center close to the previous eye center then this measurement is used. If this strategy fails, the second strategy is used, which searches a window around the eyes and analyzes the likelihood of each non-skin connected region being an eye. The search space is limited to windows of about five percent of the image size around the eyes. The slant of the line between the lip corners is found. The eye centers selected are the centroids that form a line having the closest slant to that of the lip corners. From the experiments run thus far, this method by itself can get lost after occlusion. For simplicity, these two strategies together are referred to as the eye-black-hole tracker.

The third strategy, the affine tracker [30], runs independently of the first two strategies. The system computes the affine transformation between the windowed subimages around both eyes and then, since it knows the eye center in the previous frame, it warps the subimage of the current frame to find the new eye center. Since the eye-black-hole tracker finds the darkest area, during eye occlusion instead of finding eye regions, it will get lost. When there is rotation or occlusion or when the eye-black-hole tracker produces an estimate that is too far away from the previous frame, the affine tracker is used solely. In all other cases take an average of the locations provided by the two trackers to be the eye center.

Figures 5 and 6 show some results of the eye and mouth tracker. The mistracks are infrequent, and the system always recovers. Mistracks happen no more than 2.5 percent of the time. When a mistrack occurs usually the system recovers within the next 2-3 frames. The system handles and is able to recover from heavy occlusion automatically.



Fig. 6. Feature tracker with eye blinking

Whenever the distance between the eyes gets to more than $\frac{1}{4}M$ (where M is horizontal image size), the eyes are reinitialized as this means the system encountered problems. This criteria was adopted because both the location of the camera in the car and the approximate size of the head are known. The eyes are reinitialized when the lips reappear after complete occlusion, which is determined to be when the number of lip pixels in the lip region drops below some threshold and then reappears.

This feature tracker is very robust; it tracks successfully through occlusion and blinking. Further, it is not affected by a moving background, and it has been verified to track

continuously on sequences of over 1000 frames. The system is not foolproof. Given a low-lighting picture, like those taken at night, the method of head tracking may break down. However, the system has been tested on nineteen sequences ranging from 30-1100 frames including yawning sequences and stationary and moving vehicle sequences and the system appears to be very robust and stable. A total of eight drivers were tested for a total of about four minutes of video data. Some of the input sequences had instances where part of the face was illuminated and part of the face was in shadow, and the system successfully tracked through such instances. It is able to track through so many varied conditions because of the multi-strategy trackers which work well in varying conditions. The first stage trackers work well in constant illumination. When illumination changes occur these methods will still usually give a reliable estimate as all the pixels in the region will decrease or increase uniformly. For those situations in which this doesn't happen and a bad estimate is given, the affine tracker will usually be able to successfully cope with partial illumination changes as it computes global motion of a region making it less susceptible to small changes in part of the region. Drivers with lots of facial hair were not tested in the current system. With the reliance on skin color, facial hair could distort some of the measurements. This would be a good direction for future research. However, with the system's multi level approach it was able to cope with some facial hair. Some of the results can be seen at <http://www.cs.ucf.edu/~rps43158/Project/HeadTracker/Tracking.html>

VI. BOUNDING BOX OF FACE

To find the box around the face the system starts at a fixed distance from the center of the face and looks inward until a high number of skin pixels are observed, which will indicate that a side of the face has been reached. These measurements will yield the sides, top and bottom of the face.

The center of the head region is computed using the average of the eye centers and lip corners as only a rough estimate is needed here. Then for each side of the face, the search starts at a constant distance from the center of the face and goes inward finding the first five consecutive pixels that are all skin, where the outermost pixel is recorded as the side of the face for this side and row. Using five pixels protects from selecting the first spurious skin pixel. For each row in the image the process is repeated recording the (x, y) locations for each side of the face. These x values are averaged to get a smooth line for each side of the face. This approach gives an acceptable face contour. Figure 7 shows the contour of the face that the system finds along with the bounding box of the face.

VII. OCCLUSION, ROTATION, AND BLINKING

Often the driver blinks or rotates the head, and so, occlusion of the eyes or lips occurs. The tracker is able to track through most occlusion, but does not recognize that occlusion (from rotation or blinking) has occurred. The occlusion model presented deals with rotation and blinking, two important factors for monitoring driver visual attention.



Fig. 5. Feature point tracker



Fig. 7. Face contour and face trace

Because of foreshortening, when rotation occurs, the distance between the feature points and sides of the face will increase or decrease. In each frame, compute the distance from the sides and top of the face to the eye centers, and also compute the distance from the side of the face to the mouth corners. In consecutive frames the system finds the difference in the distance for a particular D_i , the derivative, and the system looks at the sign of this difference. When more than half of the last 10 frames have the same sign of the derivative for a particular feature point then this feature point is assumed to be involved in head rotation.

Figure 8 shows the distances that are tracked on the actual image. Each distance that is tracked is labeled in the figure by $D1, D2, \dots, D8$. Figure 9 shows a sequence where the driver gradually rotates his head toward the right. Figure 10 shows graphs of the distance, derivative, and sign of the derivative of $D4$ as a function of the time(frame numbers). All three measures are displayed to progressively show how the rotation data is extracted. Since the signs of the derivatives of all frames except one outlier are positive, the rightward rotation is detected for this feature. In the experimentation on thousands of frames, the sign of the derivative was found to provide the most stable information. Next, the system analyzes these rotations of individual features to determine rotation of the face.

A voting system is constructed where each feature point determines the direction of rotation. When at least half of the feature points detect rotation in one direction, then the system declares rotation in this particular direction is happening. Each feature point can be involved in rotation along combinations of directions, but some cases are mutually exclusive(e.g., simultaneous left and right rotation). The system was experimentally verified to successfully detect rotation along multiple axis (e.g., up and left). In translation parallel to the 2-D image plane, there is no foreshortening, and thus none of the distances in Figure 8 decrease, which allows us

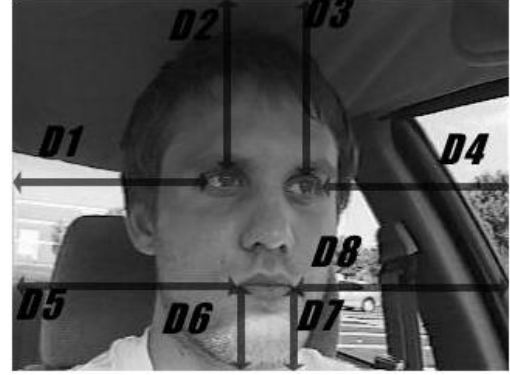


Fig. 8. Distances that the program monitors to classify rotation



Fig. 9. A few frames from the sequence corresponding to the data graphed in Figure 10. The looking right motion is gradual.

to differentiate rotation from translation of the head. Figure 11 shows the output of the feature tracker including rotation analysis messages, automatically displayed by the system.

For eye occlusion as long as the eye region contains eye white pixels then it is assumed that this eye is open. Otherwise this eye is assumed to be occluded. Each eye is checked independently. In the first frame of each sequence the system finds the brightest pixel in the eye region to determine what is considered eye-white color. This allows the blink detection method to adapt to various lighting conditions for different data sets. Some drivers wear eye glasses. Currently there are methods available which are able to remove eye glasses. Future work on our system could use these methods to work with eye glasses. For the above eye occlusion detection, each eye is independent of the other. This method gives good results. Figure 20 shows some of the results from blink detection for both short blinks and long eye closures.



Fig. 11. Feature tracker with rotation messages

A. Reconstructing 3D Gaze Direction

The problem of 3D reconstruction is a difficult one. Many current approaches use stereo to determine the 3D geometry. With uncalibrated cameras the problem becomes even more difficult. It may be expensive and impractical to have multiple cameras in a single car looking at one person to derive 3D data. Therefore, this paper provides a solution to the 3D gaze tracking problem using a single camera. This is possible as only the direction of the gaze is needed. It doesn't matter if the gaze stops at the windshield or another car thus the need to know the distance from the head to the camera is eliminated. Since head size is relatively constant between people then this distance can be fixed. The method used here is to have parallel eye gaze rays. The (x, y) locations of the eyes, the points directly behind the eyes lying on the back of the head, and the back of the head are all that is needed. Since the camera is facing the driver, when the driver is looking forward his/her face is roughly parallel with the (x, y) plane and the $+z$ axis is coming out of the face. The back of the head can be approximated well by doing the following: Consider the midpoint between the two eyes in the image. Extend a line perpendicular to the (x, y) plane going through the point between the two eyes. If the line in the $-z$ direction is extended it will pass through the head and come out eventually. The point where it comes out of the head is the point named the back of the head. Since we are using parallel eye gaze rays, the points directly behind the eyes on the back of the head are needed. With the driver facing forward initially, the initial (x, y) eye locations correspond to the (x, y) location of the points directly behind the eyes on the back of the head. Since the distances between the back of head and the points behind the eyes lying on the back of head are constant, these distances can be subtracted/added with the current back of head point to

acquire the new positions of the points behind the eyes. These points are needed to compute the parallel eye gaze direction. These sums are added to the initial location of the back of the head to find the new location of the points directly behind the eyes. This assumption is valid because when rotation occurs the average position of the two eyes moves in the opposite direction to the back of the head. Since the (x, y) location of the eyes, the back of the head and thus the points behind each eye are known, lines can be drawn in xyz space showing the direction of the gaze. This model will give a fairly accurate gaze estimate.

Figure 12 shows in detail how to compute the 3D gaze. Notice that the view is of the xz plane. Before rotation the head is looking forward, and the system records the middle point between the two eyes in xy space. If a line is extended from each eye center in the $-z$ direction, to a point directly behind each eye, each line will end at a point displaced from the point that in this paper is referred to as the back of the head. These two points and the initial back of head are the reference points for the gaze calculation. In the right half of the figure rotation has occurred to the right. Notice that the eyes have shifted and the middle point between the two eyes has moved with relation to the middle point between the two eyes in the initial frame. The head is symmetric so if the eyes rotate in one direction the back of the head rotates the same amount in the other direction. Thus, by taking the difference of the center point between the eyes in all consecutive frames and adding this difference to the back of the head in the opposite direction the system will acquire the new back of the head. From here, since the distances from each point behind the eyes to the center of the head were recorded, the new locations of the points behind the eyes can be found. The eye tracker already provides us the (x, y) eye locations. The location of the back

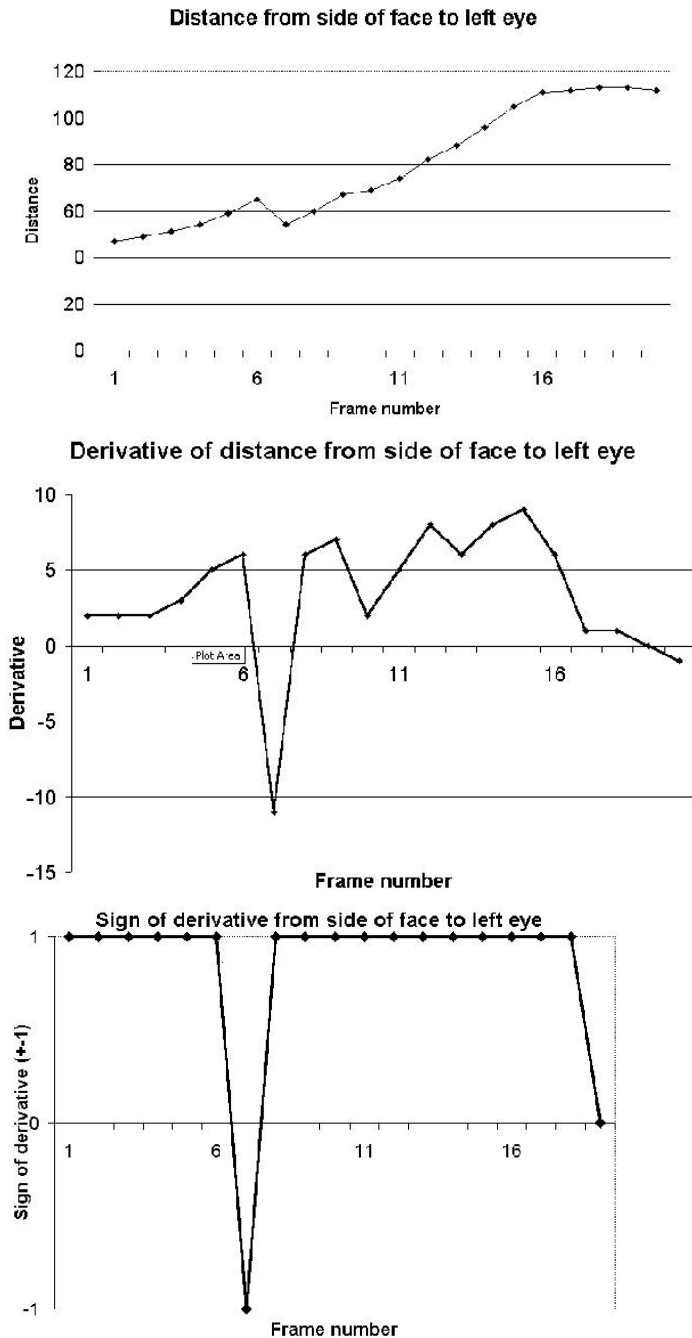


Fig. 10. Graphs showing the distance from the side of the face to the left eye, its derivative, and the sign of derivative as a function of eye frame number.

of head (x, y) is known. By projecting lines infinitely from a point behind the eyes through the eyes in the z direction the gaze will be projected in 3D space.

Figure 13 shows some results from acquiring 3D gaze information. The first two pictures are the input pictures. The last two figures are the graphical 3D representation of the scene from the (x, y) plane (with the z axis coming out of the page). Figure 14 shows similar results. From the figures one can see that the eye gaze direction is independent of head motion. It is thus possible to generate statistics of the driver's gaze, and it is possible to determine where the driver is looking

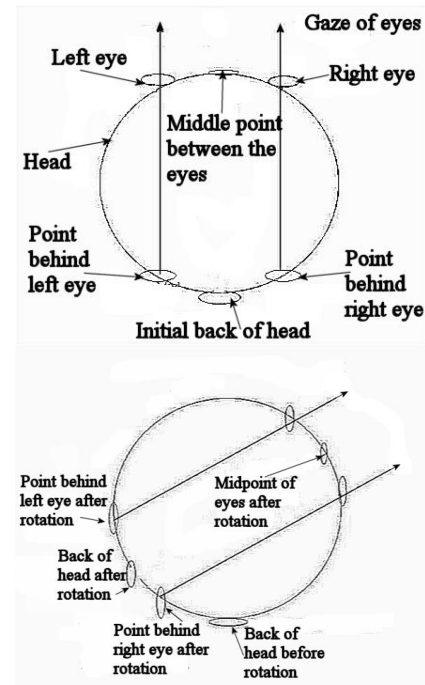


Fig. 12. Pictorial representation of 3D gaze reconstruction. The picture on the left is the picture of head from above (top view) for the initial frame of sequence. The picture on the right is the picture after rotation to the right (top view); note the shifted eye positions and back of the head, represented by a box

in 3D space, using only a single camera. In the experiments the resolution of the face was relatively low and certain kinds of head/eye configurations would give inaccurate measurements. When the head rotated and the eyes did not rotate in their sockets, the above measurement worked fine. When the head did not rotate and the eyes rotated in their sockets the above measurement worked fine. When the head rotated and the eyes rotated in their sockets the measurements could be distorted. For instance if the head and eyes rotated in opposite directions the results would be distorted, though this problem can be avoided by having higher resolution images of the eyes to determine more precisely their location relative to the head.

VIII. DRIVER VISUAL ATTENTION

This section describes the method to determine the driver's visual attention level. This method shows merely that a system could be constructed fairly quickly using simple assumptions of head rotation. More of a rigorous analysis on the physiology of visual attention is necessary before accurately determining when a driver's visual attention has decreased.

Many of the previous computer vision methods focused solely on PERCLOS measurements, essentially eye closure measurements. In the present work the system classifies visual attention with eye closure metrics, but also shows that it can be used to acquire other very important statistics. For instance if the driver rotates his/her head, then this is a decrease in the visual attention, but it may or may not result in a decrease in an eye closure metric, depending on how much occlusion of the eyes occurs. By explicitly detecting rotation, the method gives a framework for dealing with other kinds of decreases in



Fig. 14. Acquiring 3D information. First row: input images. Second row: here the rotation is occurring left and right, there is no need to display a side view since there is no rotation up or down.



Fig. 13. Acquiring 3D information. The cylinder represents the head, the lines coming out of the cylinder represent the eye gaze, and the plane represents the dashboard. First row shows the input images, the second row shows the side view, and the third row shows the top view (birds eye view). It is clear from the side view that there is up and down movement of the head. Also, it is clear from the top view there is no left or right rotation.

visual attention. Neither eye closure metrics alone nor a strict drowsiness model could detect these other kinds of events.

Driver visual attention is modeled with three independent Finite State Automata(FSM). To get the global driver visual attention level the system takes the union of all the machines; The FSM's are shown in Figures 15 and 16. The third FSM is similar to the second except it models left and right rotation. Some transitions are not defined; for instance the low visual attention states do not have a transition for no rotation. This would correspond to the driver keeping his/her head in the rotated position. The FSM ignores these frames, and the appropriate interpretation is that the FSM would not change states. It would stay in the low visual attention state, which is the expected output, because even if the driver stops rotating his/her head, it will still be in a rotated position, looking away from the road. States are represented by circles, and descriptions of states and descriptions of transitions are represented by boxes, which provide a short explanation of the state.

The first FSM monitors eye closure metrics. If the eyes are closed for more than 40 out of the last 60 frames then the system warns that the driver has a low visual attention level. Specifically, the number of frames where the driver has his/her eyes closed are counted. If there are more than 40 frames where the driver's eyes are closed in the last 60 frames, the system reports decreased driver visual attention. There have been many studies done on the lengths of the times the eye must be closed in certain time intervals in order to determine if the driver's visual attention is decreasing. In particular [15] and [29] both reviewed some of the eye closure technologies. In [29] standards were given for comparing new drowsiness technologies. Further research was done in [11]. The ratio in this paper is not intended to supplant other more established ratios. Rather the ability of the system to collect these metrics shows that it could be easily modified to conform with accepted norms of eye closure metric information. Now

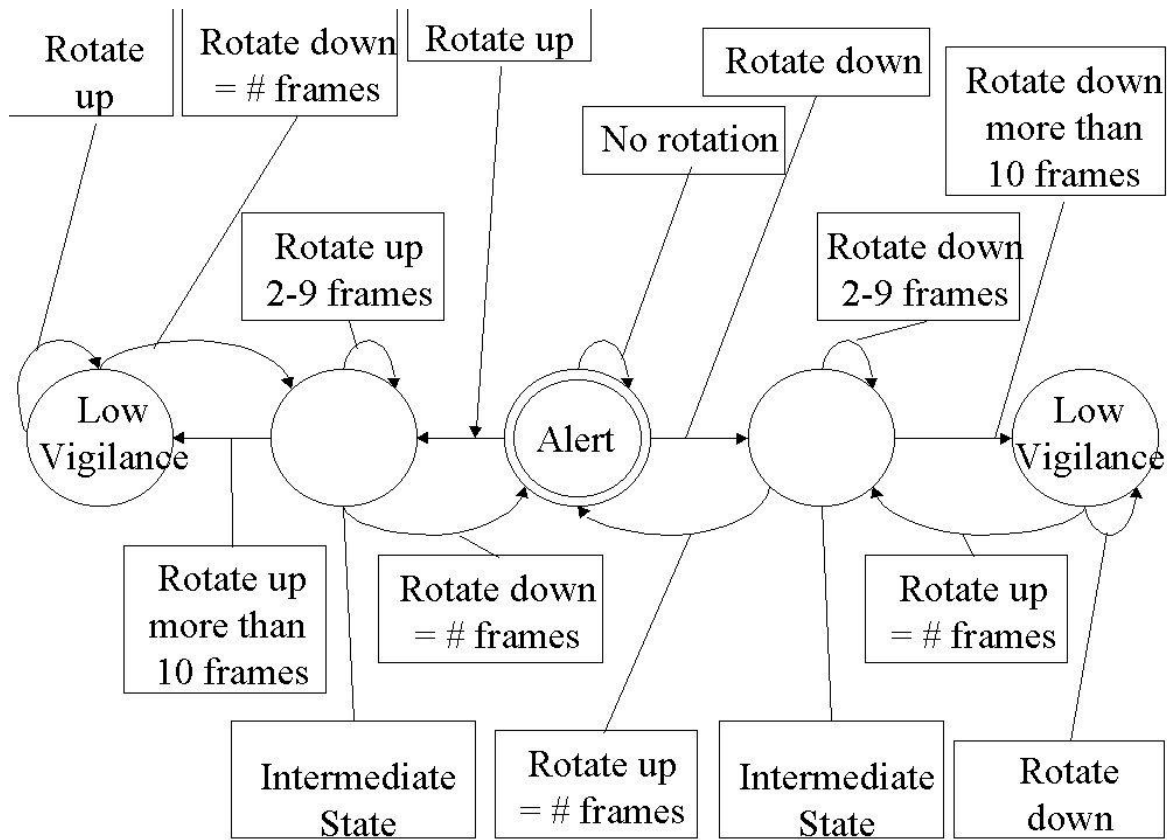


Fig. 16. Finite State Automata for driver looking up and down

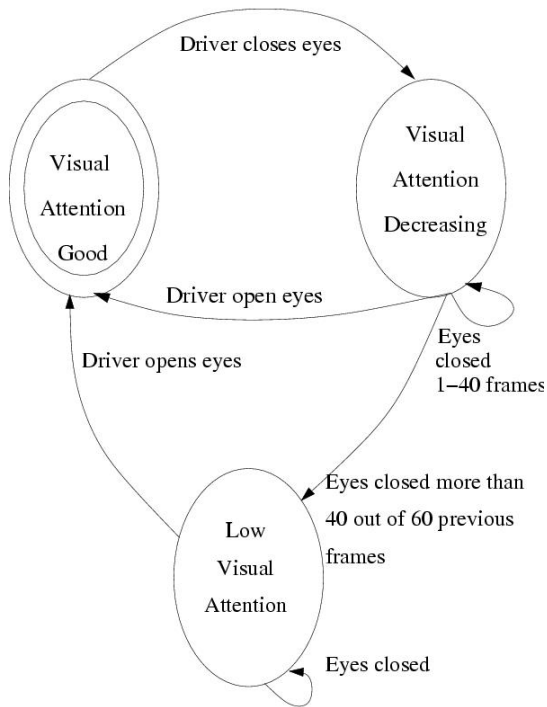


Fig. 15. Finite State Automata for driver closing eyes/blinking

depending on further studies into sleep physiology this ratio could be modified. Eye blinking is short enough so that it will not hinder the system in determining that the driver has become inattentive. This demonstrates that the system can monitor eye closure rates and can be used to determine visual inattention resulting from eye closure. In the tests on blinking data the system never made the mistake of classifying blinking as low visual attention. The blinking tests were done on about 1100 frames of video data, and were tested in various conditions from a bright sunny day, to a dimly lit parking garage. When the driver blinks his/her eyes, the system knows that the driver's eyes are closed, however, during blinking the driver reopens his/her eyes quickly, before the FSM reaches the low visual attention state.

Prolonged rotation of the head could reduce the driver's effective visual attention as well. To determine if the driver is nodding off to sleep or to determine if the driver is not paying adequate attention to the road because of rotation, the duration that the driver has been rotating his/her head is recorded. If rotation in a single direction occurs for more than 10 out of the last 20 frames then the method assumes that the driver is not paying adequate attention to the road. It was determined that 10/20 frames of rotation in a particular direction gave a fairly reliable basic criterion of indicating whether the driver was paying adequate attention to the road or not. The system records the number of frames rotated in a rotation sequence. For the driver to increase his/her visual attention, he/she must rotate his/her head the same number of frames in the opposite

direction, which will put his/her head looking straight ahead. Figures 17 to 19 show the results of characterizing driver visual attention based on the rotational data.

Again, this paper does not delve into the physiology of driver visual attention, rather it merely demonstrates that with the proposed system, it is possible to collect driver information data and make inferences as to whether the driver is attentive or not. One could more carefully define the system with a more extensive finite state machine to get more accurate classifications.

As already stated, more than alertness must be monitored. Information such as if the driver's head is pointing straight ahead is useful. These characteristics can be referred to as visual attention characteristics. The system successfully measures visual attention characteristics, and it shows that this information can be used to monitor driver visual attention. Both visual attention and drowsiness must be monitored and the proposed system is able to do both. No attempt here is made to delve into the physiology of visual attention but merely to show that attributes such as rotation must be considered, and that the system gathers these statistics and is able to infer the visual attention level of the driver. The system could be adapted to conform to norms of visual attention parameters, such as, how far the head could rotate before the driver is visually inattentive.

IX. QUANTITATIVE RESULTS

Throughout the paper various methods to compute feature points, rotation, blinking, and driver visual attention have been presented. In each section examples of how the method performs in a variety of environments with different drivers have been provided. Since the results have been shown one frame at a time it can be hard to see how well the method works. In this section all the results presented are collected together in tabular form. Performance was measured by comparing system performance to the observed motions, eye closures, rotations, gaze directions etc... that were in the recorded video sequences. For instance, for ground truth on tracking, if the system marked the feature points in the eyes and on the mouth corners, the system was deemed to have a successful track for that particular frame. All other ground truths were developed in a similar way by observing the input videos. The tracking results are presented in Table I. The results and ground truth comparison was done for a subset of all our data. Other sequences performed similarly. In Table II quantitative results of the method to reconstruct the gaze are shown.

The results from the eye closure detection algorithm are presented in Table III. Whether the driver closed his/her eyes for one frame or twenty frames, it counted as one eye closure. The ratio of number of eye closures the program detected for the whole sequence over the total number of eye closures counted in ground truth is shown in column three. The other columns are self explanatory.

Sequence 7 had erroneous results. There were many false positives detected. The system can adjust to varying lighting conditions based on the initial frame of the sequence. This sequence, was particularly dark, but in the first frame, the

Tracking Results			
Sequence #	Total # Frames in Sequence	# Frames Successfully Tracked	Percentage Correct
Sequence 1	48	48	100%
Sequence 2	28	25	89%
Sequence 3	28	28	100%
Sequence 4	28	28	100%
Sequence 5	28	24	86%
Sequence 6	400	350	88%
Sequence 7	400	390	98%
Sequence 8	200	199	100%

TABLE I

Gaze Reconstruction Results			
Sequence #	Total # Frames in Sequence	# Frames Gaze Successfully Reconstructed	Percentage Correct
Sequence 1	48	48	100%
Sequence 2	28	25	89%
Sequence 3	28	28	100%
Sequence 4	28	15	54%
Sequence 5	28	24	86%
Sequence 6	400	350	88%
Sequence 7	400	350	88%
Sequence 8	200	165	83%

TABLE II

system's automatic initialization found a high intensity value in the eye region because of some reflection in the eye white region. In subsequent frames there were lighting changes in the image and so the eye whites appeared darker as the reflection was gone, which caused the eyes to appear darker than usual. So eye closing was frequently detected when there was none because the intensity initialization threshold was a high intensity level, which did not reoccur frequently in the eye region because the lighting of the sequence, in general, was so low.

Table IV shows rotation detection results. The rotation detection bases itself on the previous 10 frames, so it took about 5 frames for the method to start detecting the rotation. This effect is factored into the results table, by counting only the rotation frames that the program could have detected. Each row in the table contains a direction. All the sequences which had rotation in that direction are contained in the respective row. As can be seen from looking at the table, the system gave false positives a small percentage of the time. For the row which starts with none, this indicated that there was no rotation, but the program actually detected rotation. This situation happened in only one sequence.

One comment is that sacrificing a little accuracy to gain robustness helped system performance. It was important to be confident that the system always gave eye estimates near the eye centers. The system needed to be able to recover from full occlusion and other extreme changes in the system. Without this kind of robust tracking a system would be of little value, since it would not be able to cope with normal and natural head movements. The system presented here is able to cope with these real world scenarios. Because of this the feature points are not always exactly correct. However,

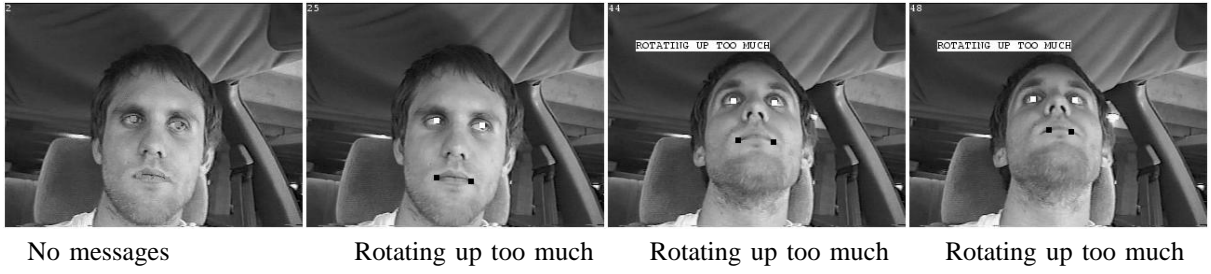


Fig. 17. Driver visual attention output

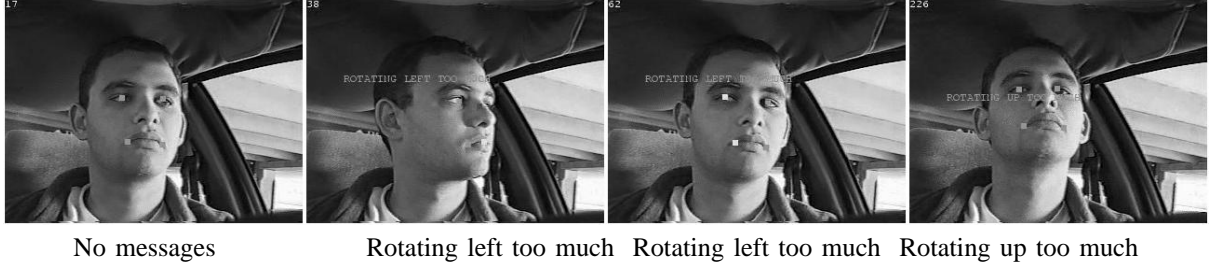


Fig. 18. Driver visual attention output



Fig. 19. Driver visual attention output

Eye Closure Detection Results							
Sequence #	Total # Frames in Sequence	# Eye Closures Detected /Total # Eye Closures	# Frames Eyes were Actually Closed	# Frames Program Detected Eyes Closed	#Missed	False Positives	Percentage Detected
Sequence 1	48	0/0	0	0	0	0	NA
Sequence 2	28	2/2	9	8	1	2	78%
Sequence 3	28	1/1	1	1	0	1	100%
Sequence 4	28	1/1	1	1	0	2	100%
Sequence 5	28	1/1	5	4	1	2	60%
Sequence 6	400	10/10	80	70	10	30	75%
Sequence 7	400	1/1	3	40	1	37	7%
Sequence 8	200	2/2	5	5	3(only left eye missed)	0	70%

TABLE III

rotation analysis methods that can cope with slightly degraded data were developed because it was important to have a system that could cope with realistic changes in the image. The best way to evaluate rotation and blink detection is to observe the image sequences and get ground truth, which was done here.

X. SUMMARY AND FUTURE DIRECTIONS

This paper presented a method to track the head, using color predicates to find the lips, eyes, and sides of the face. It was tested under varying daylight conditions with good success. The system computes eye blinking, occlusion information, and

rotation information to determine the driver's visual attention level.

Because of the inherent complexity in dealing with real world data sets, combined with moving backgrounds, variable lighting conditions, and heavy occlusion of eyes and lips, many image analysis techniques were introduced, which require selecting parameters for window sizes. One improvement will be reducing the need of empirical settings using dynamic window sizing depending on confidence of the tracking or learning approaches.

The system is not perfect and one problem that was noticed

Rotation Detection Results 8 Sequences, 1100 Frames				
Direction	# Frames Detected/# Actual (Ground Truth)	Frames Missed	False Positives	Percentage Detected
Up	115/148	33	0	78%
Down	29/45	16	0	64%
Left	87/115	28	0	76%
Right	89/110	21	0	81%
Right and Up	20/22	2	0	91%
Left and Down	6/12	6	0	50%
Right and Down	21/25	4	0	84%
None	0/0	0	12	NA

TABLE IV

is that when the eyes close, sometimes the system concludes that there is rotation because the remaining part of the eye is lower so the eye feature points move lower and lower. This problem could be addressed using knowledge that if the lip feature points don't move down then the eyes must be closing in contrast to the case of the lips moving down, which would indicate rotation.

There are many future directions for driver visual attention monitoring. For aircrafts and trains, the system could monitor head motions in general and track vehicle operator visual attention. The system could easily be extended to monitor patients in hospitals and in distance learning environments.

The presented method can recognize all gaze directions, and a next step would be to classify checking left/right blind spots, looking at rear view mirror, checking side mirrors, looking at the radio/speedometer controls, and looking ahead. Other improvements could be coping with hands occluding the face, drinking coffee, conversation, or eye wear.

REFERENCES

- [1] "Traffic safety facts 2000: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system," U.S. Department of Transportation Administration: National Highway Traffic Safety Report, DOT HS 809 337, December 2001, p 100.
- [2] R. Kjeldsen and J. Kender, "Finding skin in color images," in *Face and Gesture Recognition*, 1996, pp. 312–317.
- [3] P. Smith, M. Shah, and N. da Vitoria Lobo, "Monitoring head/eye motion for driver alertness with one camera," in *15th International Conference on Pattern Recognition*, vol. 4, 2000, pp. 636–642, barcelona.
- [4] L. Tijerina, "Heavy vehicle driver workload assessment," U.S. Department of Transportation: National Highway Traffic Safety Administration, DOT HS 808 466, October 1996.
- [5] Q. Ji and G. Bebis, "Visual cues extraction for monitoring driver's vigilance," in *Procs. Honda Symposium*, 1999, pp. 48–55.
- [6] M. Kaneda, H. Iizuka, H. Ueno, and *et al.*, "Development of a drowsiness warning system," in *11th International Conference on Enhanced Safety of Vehicles*, 1994, munich.
- [7] R. Onken, "Daisy, an adaptive knowledge-based driver monitoring and warning system," in *Procs. Vehicle Navigation and Information Systems Conference*, 1994, pp. 3–10.
- [8] H. Ueno, M. Kaneda, and M. Tsukino, "Development of drowsiness detection system," in *Procs. Vehicle Navigation and Information Systems Conference*, 1994, pp. 15–20.
- [9] W. Wierwille, "Overview of research on driver drowsiness definition and driver drowsiness detection," in *11th International Conference on Enhanced Safety of Vehicles*, 1994, munich.
- [10] K. Yamamoto and S. Higuchi, "Development of a drowsiness warning system," in *Journal of SAE Japan* 46(9), 1992.
- [11] L. Harley, T. Horberry, N. Mabbott, and G. Krueger, "Review of fatigue detection and prediction technologies," National Road Transport Commission, September 2000.
- [12] A. Suzuki, N. Yasui, N. Nakano, and M. Kaneko, "Lane recognition system for guiding of autonomous vehicle," in *Proceedings of the Intelligent Vehicles Symposium*, 1992, pp. 196–201.
- [13] D. J. King, G. P. Siegmund, and D. T. Montgomery, "Outfitting a freightliner tractor for measuring driver fatigue and vehicle kinematics during closed-track testing," *SAE Technical Paper Series: International Truck & Bus Meeting & Exposition*, no. 942326, November 1994.
- [14] T. Selker, A. Lockerd, J. Martinez, and W. Burleson, "Eye-r, a glasses-mounted eye motion detection interface," in *CHI 2001 Conference on Human Factors in Computing Systems*, 2001.
- [15] D. F. Dinges, M. M. Mallis, G. Maislin, and J. W. Powell, "Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management," U.S. Department of Transportation: National Highway Traffic Safety Administration, DOT HS 808 762, April 1998.
- [16] L. Tijerina, M. Gleckler, S. Duane, S. Johnston, M. Goodman, and W. Wierwille, "A preliminary assessment of algorithms for drowsy and inattentive driver detection on the road," U.S. Department of Transportation: National Highway Traffic Safety Administration, March 1999.
- [17] C. Morimoto, D. Koons, A. Amir, and M. Flickner, "Realtime detection of eyes and faces," in *Workshop on Perceptual User Interfaces*, 1998, pp. 117–120.
- [18] A. Gee and R. Cipolla, "Determining the gaze of faces in images," in *Image and Vision Computing*, vol. 30, 1994, pp. 639–647.
- [19] Huang and Marianu, "Face detection and precise eyes location," in *15th International Conference on Pattern Recognition*, vol. 4, 2000, barcelona.
- [20] W. Shih and Liu, "A calibration-free gaze tracking technique," in *15th International Conference on Pattern Recognition*, vol. 4, 2000, barcelona.
- [21] S. Terrillon, Sadek, and *et al.*, "Invariant face detection with support vector machines," in *15th International Conference on Pattern Recognition*, vol. 4, 2000.
- [22] R. Göcke, J. Millar, A. Zelinsky, and J. Robert-Bibes, "Automatic extraction of lip feature points," in *Proceedings of Australian Conference on Robotics and Automation*, August 2000.
- [23] R. Göcke, N. Quynh, J. Millar, A. Zelinsky, and J. Robert-Bibes, "Validation of an automatic lip tracking algorithm and design of a database for audio-video speech processing," in *Proceedings of the 8th Australian International Conference on Speech Science and Technology*, December 2000.
- [24] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Proceedings of IEEE Fourth International Conference on Face and Gesture Recognition*, March 2000, pp. 499–505.
- [25] T. Victor, O. Blomberg, and A. Zelinsky, "Automating driver visual behavior measurement," in *Vision in Vehicles* 9, 2001.
- [26] R. Grace, V. Byrne, D. Bierman, J. Legrand, D. Gricourt, B. Davis, J. Staszewski, and B. Carnahan, "A drowsy driver detection system for heavy vehicles," in *Proceedings of the 17th Digital Avionics Systems Conference*, vol. 2, October 1998.
- [27] R. Grace, "Drowsy driver monitor and warning system," in *International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, August 2001.
- [28] M. M. Mallis, G. Maislin, N. Konowal, D. M. Bierman, V. E. Byrne, R. K. Davis, R. Grace, and D. F. Dinges, "Biobehavioral responses to drowsy driving alarms and alerting stimuli," U.S. Department of Transportation: National Highway Traffic Safety Administration, DOT HS 809 202, March 2000.
- [29] M. Mallis, "Ocular measurement as an index of fatigue and as the basis for alertness management: Experiment on performance-based validation of technologies," in *Ocular Measurement as an Index of Fatigue and as the Basis for Alertness Management: Experiment on Performance-Based Validation of Technologies*, April 1999, herndon, VA.
- [30] J. R. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *ECCV*, 1992, pp. 237–252.



Fig. 20. Blink detection with feature tracker



Paul Smith received the B.Sc. M.Sc in computer science from the University of Central Florida, Orlando, Florida, in 2000 and 2002 respectively. He is currently working on his Ph.D from the University Central Florida. His research interests include machine vision and computer vision.



Dr. Mubarak Shah a professor of Computer Science, and the founding Director of the Computer Visions Lab at University of Central Florida, Orlando, is a researcher in computer vision, video computing and video surveillance and monitoring. He has supervised several Ph.D., MS, and BS students to completion, and is currently directing fifteen Ph.D. and several B.S. students. He has published close to one hundred articles in leading journals and conferences on topics including visual motion, tracking, video registration, edge and contour detection, shape

from shading and stereo, activity and gesture recognition and multi-sensor fusion. Dr. Shah is a fellow of IEEE, he was an IEEE Distinguished Visitor speaker for 1997-2000, and is often invited to present seminars, tutorials and invited talks all over the world. He received Harris Corporation Engineering Achievement Award in 1999, the TOKTEN awards from UNDP in 1995, 1997, and 2000; Teaching Incentive Program awards in 1995 and 2003, Research Incentive award in 2003, and IEEE Outstanding Engineering Educator Award in 1997. He is co-author of two books ("Motion-Based Recognition", Kluwer Academic Publishers, 1997; and "Video Registration", Kluwer Academic Publishers, 2003), an editor of international book series on "Video Computing" by Kluwer Academic Publishers, and an associate editor of the Pattern Recognition and Machine Vision and Applications journals. In addition he was an associate editor the IEEE Transactions on PAMI (1998-2002), and a guest editor of the special issue of International Journal of Computer Vision on Video Computing.



Niels da Vitoria Lobo completed the B.Sc. (Honors) degree at Dalhousie University, Canada, in 1982, and the M.Sc. and Ph.D. degrees at the University of Toronto in 1985 and 1993 respectively. He is currently an Associate Professor in the Department of Computer Science at the University of Central Florida.

His research interests center around the area of computer vision. His funding sources include the National Science Foundation, U.S. Department of Defense, Boeing Corporation, Harris Corporation and Lockheed-Martin Corporation. He has several patents, numerous publications, and currently supervises a number of graduate students. He is a member of the Computer Society of the Institute of Electrical and Electronic Engineers.

Figure 1: Lip color predicate training. The first row shows the original images, the second row shows the manually selected skin color regions, and the third row shows automatically detected lip regions.

Figure 2: Skin color predicate training. The first row shows the input images, the second row shows the manually selected skin regions, and the third row shows automatically detected skin regions.

Figure 3: Results from automatic eye and lip initialization

Figure 4: Example of dark line between lips

Figure 5: Feature point tracker

Figure 6: Feature tracker with eye blinking

Figure 7: Face contour and face trace

Figure 8: Distances that the program monitors to classify rotation

Figure 9: A few frames from the sequence corresponding to the data graphed in Figure 10. The looking right motion is gradual.

Figure 10: Graphs showing the distance from the side of the face to the left eye, its derivative, and the sign of derivative as a function of frame number.

Figure 11: Feature tracker with rotation messages

Figure 12: Pictorial representation of 3D gaze reconstruction. The picture on the left is the picture of head from above(top view) for the initial frame of sequence. The picture on the right is the picture after rotation to the right (top view); note the shifted eye positions and back of the head, represented by a box

Figure 13: Acquiring 3D information. The cylinder represents the head, the lines coming out of the cylinder represent the eye gaze, and the plane represents the dashboard. First row shows the input images, the second row shows the side view, and the third row shows the top view (birds eye view). It is clear from the side view that there is up and down movement of the head. Also, it is clear from the top view there is no left or right rotation.

Figure 14: Acquiring 3D information. First row: input images. Second row: here the rotation is occurring left and right, there is no need to display a side view since there is no rotation up or down.

Figure 15: Finite State Automata for driver closing eyes/blinking

Figure 16: Finite State Automata for driver looking up and down and right

Figure 17: Driver visual attention output

Figure 18: Driver visual attention output

Figure 19: Driver visual attention output

Figure 20: Blink detection with feature tracker

Table I.

Table II.

Table III.

Table IV.

LIST OF FIGURES

1	Lip color predicate training. The first image is the original image, the second shows the manually selected skin color regions, and the third shows automatically detected lip regions.	4
2	Skin color predicate training. The first image is the input image, the second shows the manually selected skin regions, and the third shows automatically detected skin regions.	4
3	Results from automatic eye and lip initialization	4
4	Example of dark line between lips	4
6	Feature tracker with eye blinking	5
5	Feature point tracker	6
7	Face contour and face trace	6
8	Distances that the program monitors to classify rotation	6
9	A few frames from the sequence corresponding to the data graphed in Figure 10. The looking right motion is gradual.	6
11	Feature tracker with rotation messages	7
10	Graphs showing the distance from the side of the face to the left eye, its derivative, and the sign of derivative as a function of frame number.	8
12	Pictorial representation of 3D gaze reconstruction. The picture on the left is the picture of head from above(top view) for the initial frame of sequence. The picture on the right is the picture after rotation to the right (top view); note the shifted eye positions and back of the head, represented by a box	8
14	Acquiring 3D information. First row: input images. Second row: here the rotation is occurring left and right, there is no need to display a side view since there is no rotation up or down.	9
13	Acquiring 3D information. The cylinder represents the head, the lines coming out of the cylinder represent the eye gaze, and the plane represents the dashboard. First row shows the input images, the second row shows the side view, and the third row shows the top view (birds eye view). It is clear from the side view that there is up and down movement of the head. Also, it is clear from the top view there is no left or right rotation.	9
16	Finite State Automata for driver looking up and down	10
15	Finite State Automata for driver closing eyes/blinking	10
17	Driver visual attention output	12
18	Driver visual attention output	12
19	Driver visual attention output	12
20	Blink detection with feature tracker	14

LIST OF TABLES

I	11
II	11
III	12
IV	13