Tracking and measuring drivers eyes

David Tock and Ian Craw
Department of Mathematical Sciences *
University of Aberdeen
Aberdeen AB9 2TY, Scotland

Abstract

We describe a computer vision system for tracking the eyes of a car driver in order to measure the eyelid separation. This measure is used as part of a larger system designed to detect when a car driver is becoming drowsy. The system runs unattended in a car on modest hardware, does not interfere with the driver's normal driving actions, and requires no co-operation from the driver.

1 Introduction

Our system consists of a colour CCD camera mounted behind the steering wheel of a car, connected to a Sun SparcStation 2 with a Datacell S2200 24 bit real time colour frame grabber which is installed in the boot. The system is able to compute eyelid separation, as a proxy for blink rate, usually at frame rates, so that it can be recorded synchronously with the output of other in-vehicle sensors. We describe the completed system and give sample test results over some 30,000 frames. Our initial design was described in (Tock and Craw 1992), while a fairly full description of an earlier version the system, written at a stage before detailed results were available, is given in (Tock and Craw 1995).

Early experimentation suggested that many standard vision techniques were inapplicable. We wish to track eyes, yet we find feature tracking systems, such as (Harris and Stennett 1990) and (Vinther and Cipolla 1993), typically perform poorly on people, and are even less effective in a car (see (Tock and Craw 1992)). Hardware limitations mean we cannot use certain simple features which *can* be tracked on faces (Azerbayejani, Starner, Horowitz and Pentland 1993). A further problem arises from the rapid lighting changes, normal in images from a moving car, which requires very robust tracking; thus rapid but relatively simple systems designed for faces (Gee and Cipolla 1994) are not necessarily appropriate. And our desire to ultimately detect events (blinks) which are completed within 0.1 of a second imposes speed constraints.

But there are simplifications. Since we aim to detect drowsiness, no measurements are needed in situations where the driver is constantly moving. Although

*This research was conducted as part of a contract with Ford UK.

BMVC 1995 doi:10.5244/C.9.38

the system must adapt to different drivers in different driving positions, allowing movements of the seat and steering wheel, these remain constant for long periods. An immediate result is not needed, nor one for every frame captured. Finally, in simplification, we assume that the system need not operate in the dark, and that drivers will not be wearing tinted glasses.

2 Overview

Fig. 1 shows the overall structure of the system, which consists of seven functionally independent modules. Module 0 attempts to compensate for gross changes in lighting. Although the camera has an AGC function, the range of illumination levels over which this performs satisfactorily is limited. The frame grabber provides facilities to adjust to different levels of input; provided the camera is not being driven to saturation, the frame grabber can usually have *its* sensitivity and contrast adjusted to provide a reasonable quality picture.

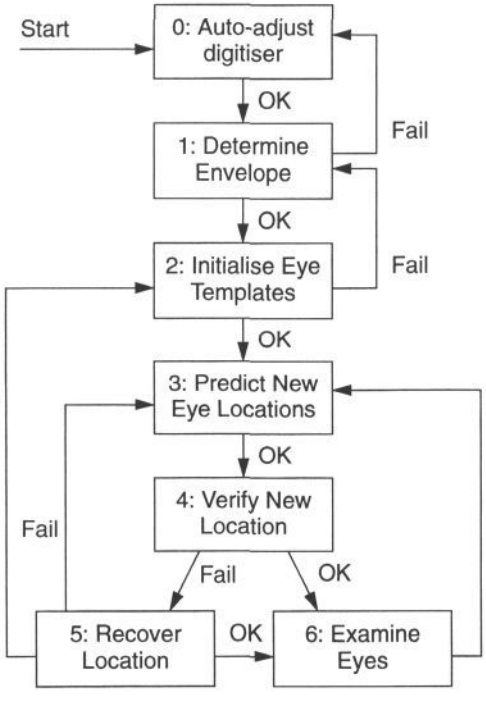

Figure 1: Block diagram showing structure of system.

Module 1 determines whether there is a driver in the vehicle, and by monitoring for several hundred frames, identifies a region of interest within which the eyes can be expected to remain. These are then initially located by Module 2. Although speed is not essential at this stage, an accurate location is important. As tracking gets under way, and further locations are available, a Kalman filter is used in Module 3 to predicting subsequent location of the eyes.

Module 4 verifies and corrects the predicted position using template matching. This stage is the one where performance tuning is most important; a small search area leads to rapid execution and hence to more accurate predictions, which in turn allow a small search area. The output from this module is either the corrected and verified eye locations, or an indication of failure. Such a failure is considered in Module 5; some modes of failure indicate a possibly transient problem, permitting the program to continue in the main tracking loop, while other failures are more serious, and require the program to revert to Module 2 for initialisation.

Module 6 examines a small area of the image surrounding the verified eye locations, and measures the eyelid separation. The module assumes that the eyes have been correctly located, seeks characteristic (line) features that should be present and makes the appropriate measurement.

3 Detailed Description

3.1 Automatic Lighting Adjustment

The S2200 frame grabber provides three inter-dependent controls (*black, white* and *clamp*) with which to adjust the characteristics of the analogue to digital (A/D) converter. In order to avoid saturation, the camera is usually adjusted to produce a normal or slightly darker than normal image. A sample image is digitised and an intensity histogram produced for each of the three colour channels. If we consider a dark image, we find the histogram skewed to the left, with few pixels occupying the higher intensity level. In this case, the *black* level must be decreased, allocating more of the available range to the darker portion of the image. Another image is captured and processed, with small adjustments being made each time. As the adjustments affect the A/D conversion, a new image must be processed to check the effect of any change. This is repeated until either no more than 0.5% of pixels have values of zero, or until no more adjustment is available. An analogous process occurs if the image is too bright.

3.2 Determine Driver Envelope

Examination of recordings taken over long periods, with a range of drivers operating in natural conditions, reveals that each driver spends almost all of their time with their head in a relatively limited area of the field of view of the fixed camera. In order to reduce the processing requirements of subsequent modules, this *region of interest*, or *driver envelope*, is first determined, defining a region within which the eyes can be expected to remain most of the time and which remains relatively constant for a given driver.

The interior of most vehicles, as seen by the dash mounted camera, is quite plain. The head lining, door pillars, seats, and occasionally the steering wheel usually have little natural colour; this becomes clear when looking at the H and S components of an HSV image. Fig. 2 gives the distribution of pixels in HS space of an image without a driver, while Fig. 3 corresponds to the same image with a driver present, and shows how a range of skin tones form a cluster in HS space. Simple thresholding of the H and S components of the image allow us to identify

potential skin regions within an image (Akamatsu, Sasaki, Fukamachi, Masui and Suenaga 1992).

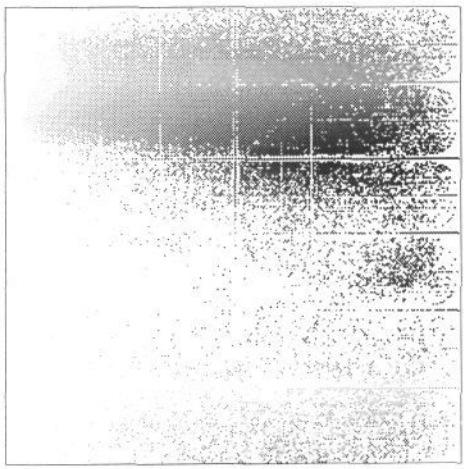

Figure 2: *Colour content in HS space of image of empty car.*

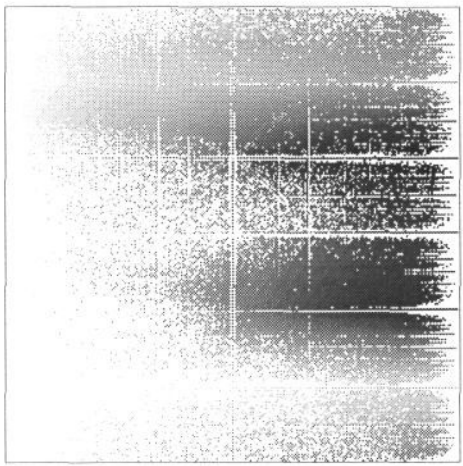

Figure 3: *Colour content in HS space of image with driver in car.*

The frame grabber operates in RGB mode. We convert to HSV based on the 6 most significant bits of each colour channel using a precomputed lookup table of a little under 8Mbytes. This rounding, combined with the rounding involved in the RGB to HSV conversion accounts for the patterns visible in Figs. 2 and 3.

By using the lookup tables, conversion and thresholding is done at close to frame rates. We can thus easily average over several hundred frames, and so eliminate short duration movements. Colour from the outside of the vehicle remains a problem, so horizontal and vertical projections of the accumulated face pixels are taken as shown in Fig. 4. The largest peak on each projection is taken to be the center of the face; the region is then extended until the corresponding histogram has dropped to 1/10th of its peak value. Finally the bottom half of the region is discarded. Over a period of time the position of the driver can change slightly, causing the region of interest to become invalid. In such cases, subsequent modules fail, and the region is re-evaluated.

If the resulting area of the region of interest is too small, it is abandoned, and the process repeated. Thus, in an empty vehicle, this module alternates indefinitely with the automatic exposure module.

3.3 Initialise Eye Templates

Tests with existing eye detectors (e.g. (Nixon 1985, Hallinan 1991, Bennett and Craw 1991)) operating on images from the vehicle demonstrated shortcomings: the most common being a trade off favouring accuracy over speed, which is inappropriate to our needs. The method adopted here, described in greater detail in Tock and Craw (1995), combines an eye proposer (Robertson and Sharman 1992), with subsequent filtering and template matching for localisation. This latter is

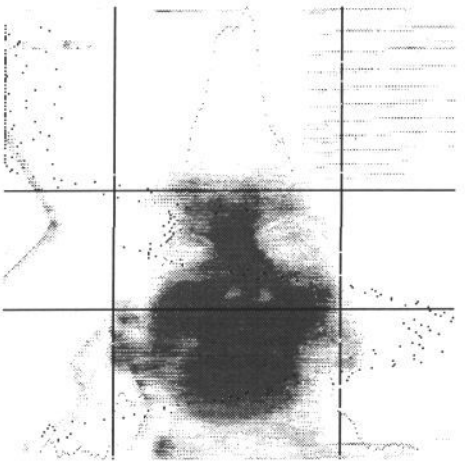

Figure 4: Projection onto axis of an intensity image from the envelope determination stage. The straight lines delimit the selected region of interest.

done against a set of ten pairs of stored eye templates, and in conjunction with a threshold, prevents the selection of non eye regions if no eyes are present. Initially this template set contains no image of the eye of the driver, but the region surrounding the selected eye locations are copied and used as one of the pairs for subsequent invocations of this module. The results obtained are sufficiently accurate for our needs, they are robust, and are produced rapidly.

3.4 Predict New Eye Location

In order to minimise the area to be examined during the next stage, we attempt to predict where the eye will be in each successive image. A simple Kalman filter predicts new eye locations: a constant velocity model is used for the position of the mid point of the eyes, and a constant distance model for the horizontal and vertical separation. The latter in fact varies considerably as the driver looks around or reacts to vehicle movement, although the drivers head may remain relatively still for considerable periods.

Even in the main tracking loop, the verification and recovery stages do not take a fixed time. The interval between processed images is determined from frame numbers and the (sometimes irregular) time intervals obtained between consecutive successful locations accommodated by the filter. The Kalman prediction yields a result in which the eye is within the required distance of the predicted location for significantly more of the time than the naive approach of assuming the position remains constant. In addition the Kalman filter proved helpful in imposing constraints required to ensure the predicted locations remain viable.

3.5 Verification of Prediction

Despite running at full frame rate (25 frames per second) the speed of movement of the driver can result in sudden large changes between successive frames. In

any case the predicted position is used as a starting point to initiate a template-matching search, and obtain a corrected eye position. The advantage of our careful initial eye location is that the template used is an image of the eye of the actual driver.

3.6 Recovery from failed Verification

When verification fails, the recovery stage checks for symptoms characteristic of specific modes of failure; the aim is to reset only as much of the system as is necessary. For example, if the steering wheel is obscuring the camera, correlation scores are usually close to 0. In this case, the missed image is ignored, and execution continues. However, should more than a predetermined number of consecutive images be missed (currently about 2 seconds worth), prediction is unreliable and the accurate eye locator is used to reinitialise the tracking system.

A second type of failure is due to insufficient illumination, perhaps caused by driving under a bridge. Although the camera and the frame grabber can compensate automatically for a wide range of lighting levels, this range is far exceeded by the variations encountered during normal driving. Detection and response are essentially the same as above, but the entire image becomes very dark.

Another failure occurs when the driver appears to move out of the region of interest. If the correlation is still high, the eye region is examined anyway; the driver may return to the region of interest. A sufficient number of frames with low correlation scores (again, about 2 seconds worth) will first cause the eye templates to be checked, and failing that, re-evaluation of the region of interest. If the main loop is going to fail completely should the next image fail, the prediction is reset to the position where the eyes were originally located. This naive strategy is remarkably successful; many of the reasons that cause tracking to fail are of short duration, and the driver returns to a normal driving position. Having a single frame look back often reestablishes tracking without reverting to the initialisation stages.

3.7 Examine Eyes

The primary output from the system is a continuous measurement of eyelid separation. Two vertical strips of the image, each 21 pixels by 9 pixels, one centered on each eye are examined. Working out in both vertical directions from the center, the maximum dark to light edge is found. These correspond to the transition from iris to eyelid. The distance between the two edges is calculated, and the mean for the two eyes is transmitted via an RS232 serial link to the data acquisition unit for recording. This forms the only output from the system when running in turnkey mode in the vehicle.

3.8 Frame Grabber Control

In order to detect the required changes, the system is required to run quickly, processing a frame in $\leq$ 40ms if every frame was to be examined. Such performance is not unusual with dedicated and customised hardware, but we work on a Sparc 2,

with a single frame grabber. Nevertheless, the final system can usually process 25 frames per second, and transmit the results to the recording unit.

A single frame buffer can cause delay; it is not possible to start processing an image until that frame is complete, while the need to immediately capture of the next frame means the frame buffer must be logically partitioned. This is possible on the S2200 board at our favoured resolution of 256×256. However, a request to the frame grabber to capture an image is normally delayed until the frame is complete, and a subsequent request to capture into the alternate buffer space will then wait until the end of the next half frame before returning, with the request to capture the next frame waiting another half frame. This 40ms delay is simply to arrange for the capture of the next image, with a corresponding reduction in frame rate to 12fps before any processing. To avoid this, it was necessary to modify the device driver. Although the control registers should only be changed during a vertical blanking period (which occurs between each half frame, i.e. 50 times per second), there is no reason why *requests* to change the register values should not be placed, and the program be allowed to perform some other calculations while waiting for the retrace. The only restriction is that the programmer must now be aware that the device driver and application program are running asynchronously.

By implementing this change, the scenario becomes quite different. When the capture of the first frame is complete, the device driver makes the necessary changes to the registers and starts capturing the next frame, while the program proceeds with processing the image just captured. Provided that processing takes less than 40ms, the program can request the capture of the next frame before the one currently being digitised is complete, and so the loop continues. If processing takes more than 40ms, there is always a new image waiting to be processed, without the program having to wait at all. This means that if processing takes only slightly longer than the 40ms available, consecutive frames will be processed, without any idle time, until the accumulated excess causes a frame to be missed. The diagram and time line in Fig. 5 attempt to illustrate this pictorially.

4 Operation and Performance

The complete tracking system is written in C++, and runs on a Sun SparcStation 2 with a Datacell S2200 24 bit real time colour frame grabber and display. In practice, a resolution of 256×256 pixels, gave an appropriate compromise between clarity in the image, and reduced execution time consistent with rapid tracking; where execution times are given below, they are for images of this size.

The first stage, adjusting the parameters of the digitiser, is an iterative process. If the original settings are acceptable, this involves only a single image, but may involve many images, especially if the lighting is changing rapidly. Even in the worst cases the whole process takes only a few seconds.

The next stage, determining the region of interest, is allowed to run for a period of approximately a minute, during which time we typically examine 100 images. The speed of this section is not critical; indeed we are trying to find the most probable position of the driver over a long periods of time. The selection of the region of interest from the averaged image yields a satisfactory result in the majority of cases, with the most other cases yielding a region that is too generous.

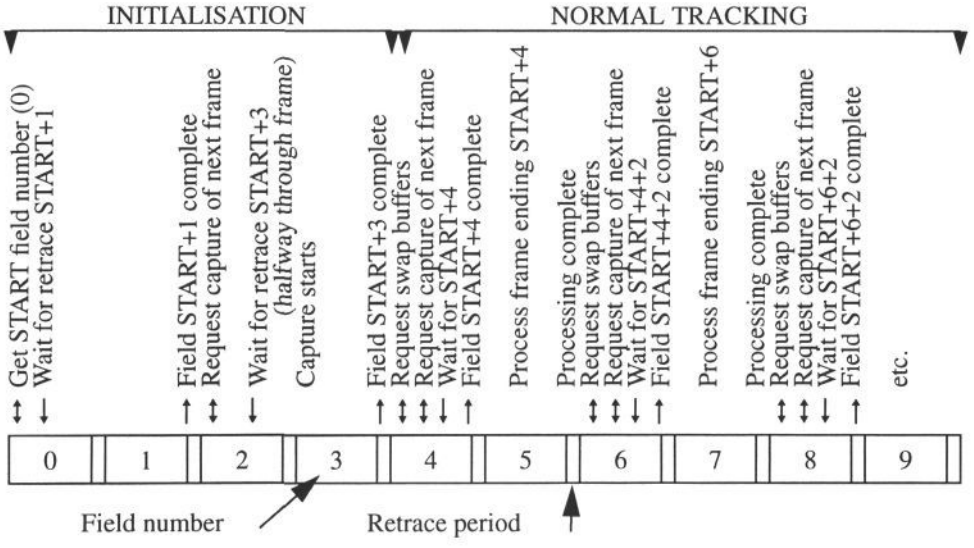

Figure 5: Time line showing frame grabber operations and timings.

The success of the main loop is largely determined by the initialisation stages. A good initialisation may be sufficient for a particular driving session, while a poor initialisation will rapidly lead to failure, and a re-initialisation. The main loop takes approximately 40ms per cycle, allowing 25 images per second to be processed. The times are approximate, as they vary to a small extent due to the variation in processing requirements that occur from frame to frame, however this cycle time has been obtained both by code optimisation, and by trimming the region used for correlation as necessary. In fact a final performance boost, replacing the standard processor with a Weitek (80 MHz) Sparc power μP chip, was needed before an adequately large region could be scanned.

The cycle time does not include time to write intermediate results to the frame grabber display; indeed doing so reduces by a factor of five the frame rate that can be maintained. If results *are* being displayed, one of the outputs is the redisplay of the verified eye regions in fixed positions on the display. The effect, when the tracking is performing correctly, is to display a pair of eyes on the screen that do not move, even though the driver may be moving. This allows a quick visual verification that the correct region is being tracked.

This interference between rapid operation and display means performance measures need to be considered carefully. One such is the ratio of time spent tracking the wrong features to time spent performing correctly. A similar measure of efficiency is the ratio of time spent initialising to the time spent tracking and generating results. Another measure of interest is how long the system can run within the main loop, without reverting to the eye initialiser, or the region of interest detector. This measure allows for the system to *fail* at the prediction, verification or recovery stages, but recover automatically within a few images, and continue without significant loss. This philosophy is similar to the measurement of MTBF used to describe hardware reliability: a recoverable error is not classed as a failure.

The three measures just described can all be calculated by observing the output from the system while it is running, and noting whether the system is initialising, tracking correctly, or tracking incorrectly. This analysis was done on a number of different drivers under different conditions for a total of 30000 frames, and the time spent in each of the three states noted; the relevant ratios are presented in Table 1. Examining why the system moved between states can provide an insight into how the program could be made more robust; in the sequence analysed, the most common cause was simply the driver moving, usually to look in a mirror, closely followed by rapid changes in lighting, typically shadows on the drivers face.

State	Duration (S)	Duration (frames)	Duration (%)
Initialising	127	88 (not at frame rate)	10.2
Tracking	1031	25775	83.3
Failing	80	2000	6.5

Measure	Calculation	Ratio (Lower is better)
Reliability	$\frac{80}{1031}$	0.077
Efficiency	$\frac{127}{1031}$	0.12
MTBF	$\frac{1031}{52 \text{ re-initialisations}}$	19.8 sec.

Table 1: Reliability and efficiency measurements.

It is much harder to give a useful measure of the robustness of the program; it has run for periods of more than an hour while continuing to produce the type of result described above. However there are lighting conditions, typically when the sun is low, when performance degrades badly.

5 An Application

The blink detection and eye monitoring system we have described forms one of a number of inputs into a driver monitoring system. The goal was to be able to detect changes in driver status by monitoring existing vehicle sensors, or simple additional sensors. These include the brakes, throttle, steering, engine speed and load, inside and outside temperature and humidity, ambient light and time of day, all of which are already instrumented on many modern cars. Additional sensors include strain gauges in the drivers seat, and other engine parameters. These needed to be correlated against the drivers condition. For the initial study, the test vehicle was equipped with a number of additional sensors which would currently be impractical in a production vehicle. These include lane-tracking, electroencephalogram (EEG) and electrocardiogram (ECG). The system described in this paper is intended to

supplement or ideally replace the EEG and ECG equipment, which is impractical for long term testing planned for the future.

The system described aims to provide the necessary correlation during the development of the driver monitoring system—a final production version would use other vehicle sensors, as already described. This is partly due to anticipated reluctance by drivers to have a camera watching them, and partly for economic reasons. In other *driving* environments, such as trains, aircraft and HGVs there would be more potential for including the camera. Here the professionals tend to welcome the prospect of a system that could warn them of impending fatigue.

References

Akamatsu, S., Sasaki, T., Fukamachi, H., Masui, N. and Suenaga, Y.: 1992, An accurate and robust face identification scheme, *ICPR 92*.

Azerbayejani, A., Starner, T., Horowitz, B. and Pentland, A.: 1993, Visually controled graphics, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**, 602–605.

Bennett, A. D. and Craw, I.: 1991, Finding image features using deformable templates and detailed prior statistical knowledge, *in* P. Mowforth (ed.), *British Machine Vision Conference 1991*, Springer Verlag, London, pp. 233–239.

Gee, A. and Cipolla, R.: 1994, Determining the gaze of faces in images, *Image and Vision Computing* **12**, 639–647.

Hallinan, P. W.: 1991, Recognizing human eyes, *Society of Photo-Optical Instrument Engineers - Geometric Methods in Computer Vision.*

Harris, C. and Stennett, C.: 1990, RAPID - a video rate object tracker, *Proceedings of the British Machine Vision Conference*, pp. 73–77.

Nixon, M.: 1985, Eye spacing measurement for facial recognition, *Proceedings of the Society of Photo-Optical Instrument Engineers.*

Robertson, G. and Sharman, K. C.: 1992, Object location using proportions of the directions of intensity gradient - prodigy, *in* C. I. Processing and P. R. Society (eds), *Proceedings Vision Interface 92, Vancouver Canada*, pp. 189–195.

Tock, D. and Craw, I.: 1992, Blink rate monitoring for a driver awareness system, *in* D. Hogg (ed.), *Proceedings of BMVC-92*, Springer-Verlag, London, pp. 518–527.

Tock, D. and Craw, I.: 1995, Tracking and measuring driver's eyes, *in* C. Brown and D. Terzopolous (eds), *Real Time Computer Vision*, Cambridge University Press, pp. 71–89.

Vinther, S. and Cipolla, R.: 1993, Towards 3D object model acquisition and recognition using 3D affine invariants, *in* J. Illingworth (ed.), *Proceedings of the* 4^{th} *British Machine Vision Conference*, BMVA Press, University of Surrey, pp. 25–34.