Design of a real-time eye tracking, blink feature and pupil meter system

2005, Dimitris Kouzis – Loukas

Introduction

This is the design of a system that can be used for accurate measurement of properties regarding eye movement and blink through a high speed video data stream. Frame rates of captured video may be up to 200 frames per second. This enforces the use of high-speed Digital Signal Processing techniques instead of complex techniques which are more often discussed in academic literature.

The Region Of Interest in this application is the eye. We assume that we have video with eye being captured from the frond. Extra pre-processing may be required in order to locate the ROI specially if the camera isn't mounted on the head of the subject e.g. on glasses. Registering, histogram equalization and other pre-processing techniques won't be covered here because they are very hardware dependant.

The system that we describe must be able to measure the quantities presented in Figure 1. Quantities are grouped by the Module that is responsible for measurement.

Module		Quantity
1	Lid Tracker	Blink interval before
		Blink duration
		Blink amplitude
		Lid closure speed
		Lid closure std speed
		Lid closure max speed
		Delay
		Lid opening duration
		Lid opening speed
		Lid opening std speed
		Lid opening max speed
2	Eye tracker	Eye movement interval before
		Eye movement duration
		Eye movement duration std
		Eye movement amplitude
		Eye movement speed
3	Pupil meter	Pupil diameter measurement

Figure 1. Required measurements

As we can see in Figure 1 the system uses three modules, Lid tracking system, Eye tracker and Pupil meter. These modules will be presented in detail in the following sections. The data flow between these three modules and some secondary utility modules are presented in Figure 2.

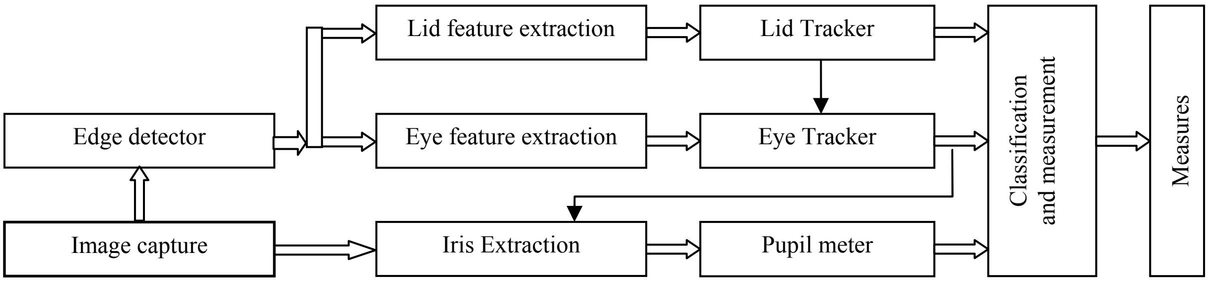

Figure 2. System data flow

This partitioning of the system reduces overall design risk by giving great flexibility where it's needed, in the classifiers and measurements module. We have one classifier module that receives and evaluates features that have been extracted from the image by other modules. This increases flexibility by making the system able to use features from different modules to achieve greatest accuracy and much faster than if we were using a single classifier for each feature extraction module.

We shouldn't ignore that by using 100 Frames Per Second (fps) data rate we have only 10ms per frame while with 200 fps there are only 5ms per frame. These hard constraints, should be taken into consideration from the very beginning of the design. The classic "make it work first, optimize then" technique can't be used in this system's design, because the goal of this project isn't to measure these quantities. It's to measure them in such high frame rate. As a result, optimization must be a part of the design and implementation process from the very early steps, in order to achieve this dual goal.

An early design decision that is being made because of the need for high performance, is to use the edge detected image as a source for our measurements. Of course this reduces the number of algorithms that can be used but all but a few very simple algorithms that process the grey levels of images are quite inefficient. Additionally grey levels can't be used because the system will be used in automotive applications. In the varying lighting conditions of the road, a grey-levels based system should re-configure it's threshold values continuously and this would dramatically affect the performance of the system. Edges give enough information for the eye and the lid. Additionally two-state values of an edge detected image can be used very effectively to increase performance with appropriate algorithm implementation. The only module that uses grey levels is the pupil meter but the area of iris and pupil is so limited that doesn't affect performance.

In depth analysis of the modules

1. Lid Tracker

Fitting, elastic tracking, snakes and other complex algorithms can't be used due to the real-time requirement. A fast and efficient solution must be found in order to recognize lid movement during blinking. The solution proposed is based on cross-correlation, a well known technique in image processing and pattern recognition. We expect the cross correlation function of the image with a lid prototype to have a global maximum at the place where the two images look similar.

In order to make the cross correlation function give accurate results, we have to choose very carefully the prototypes of the lid. These are expected to vary between subjects specially from different ethnicities. This topic will be discussed latter on this report.

For the demonstration of the cross correlation technique, we used these three images picked from the same video that is being processed. These are presented in Figure 3.

Figure 3. Lid patterns

In the following image, we can see the result of the cross correlation between the original image and the prototypes. In the lower row from left to right, there are the cross correlation functions of the original image and the open lid prototype, the mid-open lid prototype and the closed lid prototype. In the video accompanying this document, you can see the cross correlation of the full video time series. It's important to note the peak value of the cross-correlation that indicates the point of best match. It's value is being displayed on the title of cross-correlation graphs for each frame.

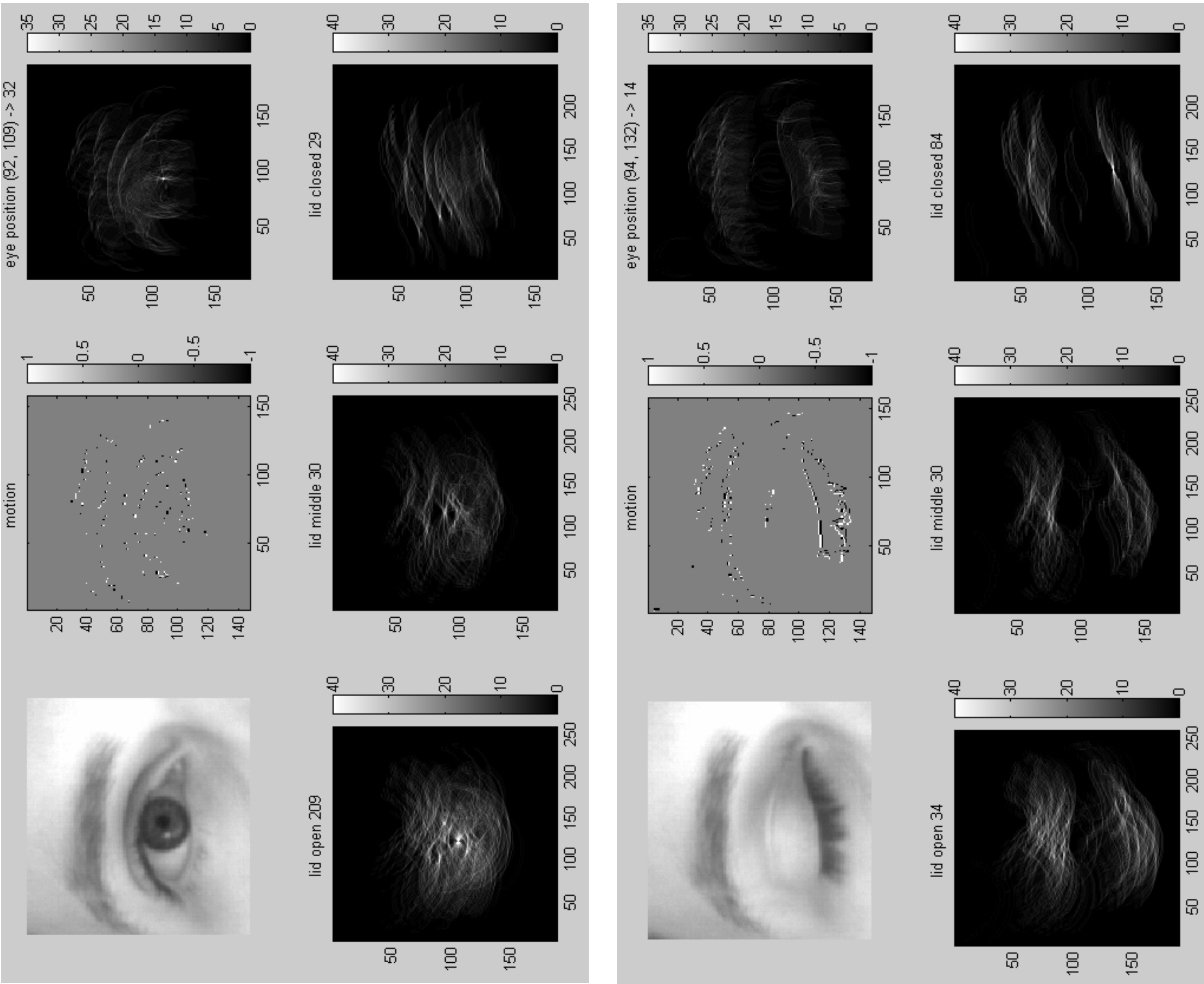

Figure 4. Algorithm's results

We summarize the results of the peak values for each cross correlation of the demo video in the following diagram.

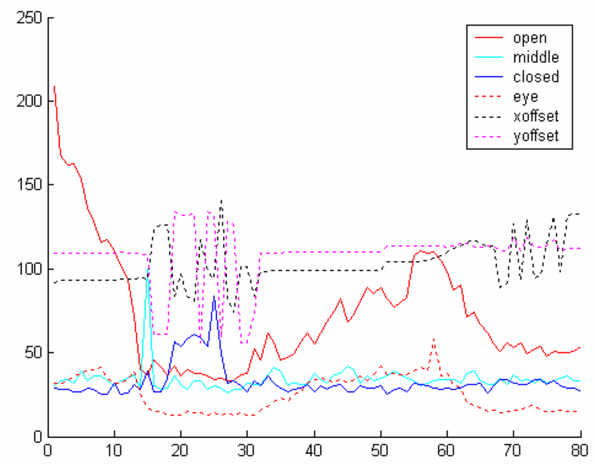

Figure 5. Results of effective feature extraction at video stream, 80 frames / 800ms at 100fps

As one can easily see, the likehood of ‘open’ and ‘closed’ states can be used to easily detect a blink. By using these peaks, we can also measure additional parameters like blink amplitude without adding extra computational effort. The middle-closed lid prototype doesn’t provide useful information apart from a full match peak during lid close.

In order to estimate if the lid matches a specific pattern we can use some pre-defined threshold value but this won’t work well in most of cases because peak’s value is highly related to the average cross-correlation value. In order to overcome this defect we can use the Peak-to-Correlation Energy (PCE) [1] metric.

$$PCE = \frac{[peak - mean]^2}{\mathrm{var}\,iance}$$

If the value of PCE is above a threshold value we can conclude that the image matches the specific pattern.

2. Eye tracker

The eye tracker must locate the iris in the eye. Of course this requires the lid to be open. Eye tracking must be fast as well. We used the cross-correlation technique with a circular iris pattern presented in Figure 6.

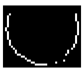

Figure 6. Iris pattern

The results are co-presented in Figure 4 and Figure 5. In Figure 4 in the upper right diagram we can see the cross correlation and above the diagram the title shows the peak position and the peak value. As we can see, in the frame where the lid is open, cross correlation has a value of 32 in contrast with closed lid where its value is 14. For a frame by frame analysis we can examine Figure 5. As we can see eye position is roughly constant (92, 109) before and after blink. During blink, of course, the position is not detected correctly because the lid is closed. This can be detected by low value of cross correlation peaks and large variance of the position signal. This is another way to detect a blink and can be used as a supplementary criterion for a classifier for blink detection.

[1] Biometric Recognition Using Correlation Filters in the Wavelet Domain, Jason Thornton, http://www.ece.cmu.edu/~jthornto/research.htm

During eye movement (frames 50 to 68) we can notice the increase of the offset on x axis which is the expected result. Y axis value remains almost constant as can be seen in the video, fact that keeps up with subject's eye movement. In Figure 5 the dotted red line is the peak value of the cross correlation of the eye pattern. The diagram is as expected. We can notice another defect of this methodology. When iris lies in the right corner of the eye, it isn't being successfully detected and measurements are less accurate. This is something expected from a careful observer. When Iris is in the right corner, its shape isn't circular anymore. Furthermore its effective radius seems to shrink. Extra care should be taken for detecting the two corner positions of the eye with an extra pattern. These positions are frequently observed in drivers and ignoring them will result to wrong classification.

Eye tracking results in a x-y diagram can be seen in Figure 7. Values with closed lid and with iris in the corner have been removed by thresholding the peaks of iris pattern cross correlation with value 70. As we can see the results are quite accurate.

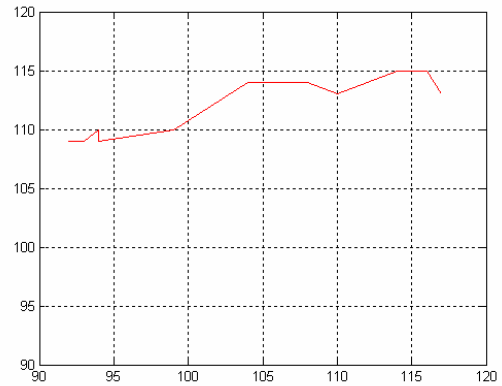

Figure 7. Effective eye tracking results

3. Pupil Meter

When we know iris' position and radius, it's easy to determine pupil's size. A high-performance solution is presented in the paper "Pupil-meter and tracking system based in a fast image processing algorithm"[2]

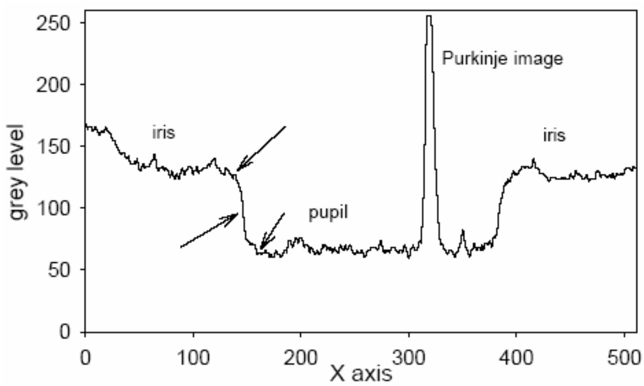

Figure 8. Radial intensity profile of pupil[2]

This uses the intensity diagram taken over an iris' radius. We can see that pupil can be easily discriminated from the iris. In some cases, black eyes may need some histogram equalization till the difference between iris and pupil becomes detectable.

---

[2] Pupil-meter and tracking system based in a fast image processing algorithm, Ignacio Miró, Norberto López-Gil and Pablo Artal, 1999

In order to prevent errors due to purkinje image we can track intensity diagram via many radius' as shown in Figure 9 and then except those that are far by the average. By averaging their values we can improve accuracy.

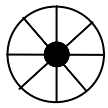

Figure 9. Profiles from multiple angles will improve robustness of the method

Real time aspects

This system is being based almost completely in the cross correlation technique. Cross correlation is a time consuming process with high computational complexity. The only way to implement cross correlation effectively is by performing it in the Fourier space. Instead of performing the inefficient analytical implementation of cross correlation, we FFT both the image and the pattern (zero padding may be needed in order to have the same size) and then we multiply them. With inverse FFT of the result, we take the cross correlation of the two images.

It must be noted that FFT version, specially if implemented in an integer arithmetic, may be less accurate[3]. Extensive testing must be done in order to verify that rounding errors don't affect overall accuracy. Because our images are binary, the direct cross correlation approach with a semi-custom algorithm may be competitively efficient and more accurate. It must be noted that FFTs of the patterns can be pre-calculated. Fast and efficient implementations of cross correlation give us the ability to cross correlate the image with many pattern images at the same time and still remain real time.

Video classifiers for efficient blink detection

We have already shown the results of feature extraction on Figure 5. The aim of the classifier is to use these signals to detect blinks.

The best solution would be to have a real 3D cross correlation engine that will cross correlate videos of blinks taken from the calibration procedure and detect completely blinks. The problem of this technique is that video stream shouldn't be correlated only with the pattern video but with several variations specially in the time domain. This means that a full set of varying speed blinks should be captured during calibration in a video database and in real time cross correlate them both in image and time domain. Of course this solution is not feasible because of it's computational complexity. Other solutions that are proposed in the literature[4] are more appealing but still can't operate well in such a high-frame video stream.

In order to have almost the same results with limited resources, we can use techniques that are being used in high energy physics for particle image classifications. We introduce the term trigger. A trigger is going to be assigned to each pattern (e.g. close lid and open lid) and is going to be fired when the PCE metric of the cross correlation is above some threshold as described in previous section. Each trigger has some properties e.g. peak location and likehood which correspond to the PCE. If we have two patterns, then we are going to have two triggers and this results to limited accuracy. We can use more (e.g. 6) triggers for patterns captured at the calibration procedure.

When we have a valid blink these triggers are going to be fired in sequence and their properties will be deterministically related. We can now use the signals of these triggers as a source of a simple classifier in

[3] Oliver Pust , "PIV: Direct Cross-Correlation compared with FFT-based Cross-Correlation"
[4] V. Argyriou and T. Vlachos, "Using gradient correlation for sub-pixel motion estimation of video"

order to detect blinks. Additionally, triggers' properties can give direct measurements of some very important quantities that we want to measure like Lid closure speed and Blink amplitude and Blink Duration.

The main benefit of this technique is that we solve the problem of time stretching of blinks and we still make video pattern recognition. We use limited number of patterns instead of images and this reduces the complexity amazingly.

System calibration procedure

The difference between system calibration and normal operation is that because the system doesn't have to provide measurements, operations don't have to be real time and as so can be more complex. In fact we prefer to make as much computation e.g. FFTs of patterns at the begging in order to reduce the computation amount that is required in normal operation.

In calibration we need to detect when there is a movement in the video stream and when it stops. We can estimate when there is a movement in the ROI by using differences between consequent frames. If their average power (V^2) is above a threshold and at the same time their variation is low, then we have a major movement in our ROI.

Considering the techniques presented in previous sections, we can summarize traced patterns required for accurate classification. These are:

Lid Tracker	Open and close lid patterns, 1-4 mid closed lid patterns
Eye Tracker	Iris pattern, left/right/top/bottom looking Iris patterns

For the eye tracker module the calibration is easy and straightforward. The only thing that has to be done is to estimate the diameter of iris. This can be done easily by Hough transform or any another slow fitting method. Alternatively the subject could be told to look at two different points. While the iris is moving, there is going to be a clear shape in the difference frames as presented in the left image of Figure 10.

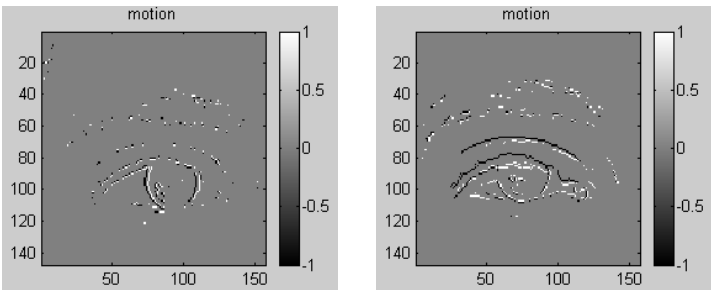

Figure 10. Differential motion images of edges

Four images that represent the left/right/top/bottom looking Iris patterns can be easily collected by telling the subject look at three specific points. When eye movement stops, edges that represent iris can be collected.

For the lid tracker module the procedure is more complex. There are two methodologies that can be used for the extraction of these patterns during the calibration procedure.

First method is based in a large image database that has pre-classified patterns from multiple ethnicities samples. By telling the subject to make some calibration blinks, which will be detected by start/stop of movement, we can see which patterns' triggers get fired and extract the pattern sequence that represents the blink.

The second methodology requires no image database but is based entirely on the extraction of traces while the eye is moving. As we can see in the right image of Figure 10, when the lid moves, this is indicated clearly in the difference between frames. By cross correlating negative and positive edges of two frames' differences, we can estimate very accurately which part has moved and how far and "lock" lid movement. This way, we can extract image patterns of the lid. In order this calibration technique to work correctly it should self-evaluate the quality of the patterns by telling the subject to make another blink and check if it's detected accurately. If not, the calibration procedure should continue.

All these patterns should be stored and re-used when the same subject is being detected (blink based access control). By this way the car could be aware of who the driver is but more importantly would enable the calibration procedure to take place only when that's needed, when a new subject drives the car.

Conclusion

In this report a complete design for a real-time eye tracking, blink feature and pupil meter system was described. New innovative ideas have been presented for the efficient performance of complex image and video recognition operations. These techniques are pure Digital Signal Processing techniques and can be implemented efficiently in software. In addition some feature extraction modules can be implemented in custom hardware (FPGAs or DSPs) in order to avoid the bottleneck of transfer of this high frame rate video stream on PC for processing. By having performance and optimization issues in mind from the very beginning of the design, we reduce application's development risk, reduce time-to-prototype and we achieve the dual goal of this project.