

# Eye Tracking

## SIGB Spring 2014

Marcus Gregersen  
[mabg@itu.dk](mailto:mabg@itu.dk)

Martin Faartoft  
[mlfa@itu.dk](mailto:mlfa@itu.dk)

Mads Westi  
[mwek@itu.dk](mailto:mwek@itu.dk)

TODO DATO

# 1 Introduction

## 2 Pupil Detection

In this section, we will investigate and compare different techniques for pupil detection.

### 2.1 Thresholding

An obvious first choice of technique, is using a simple threshold to find the pupil, then do connected component (blob) analysis, and finally fit an ellipse on the most promising blobs.

Fig 1 shows an example of an image from the 'eye1.avi' sequence and the binary image produced by, using a threshold that blacks out all pixels with intensities above 93. This manages to separate the pupil nicely from the iris.

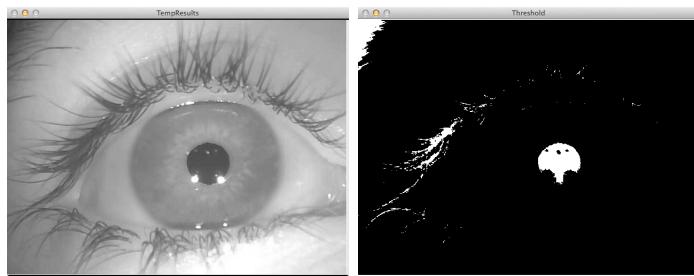


Figure 1: Thresholding eye1.avi

The next step, is to do connected component analysis, and fit an ellipsis through the blobs. As seen in fig 2, this successfully detects the pupil, but is extremely prone to false positives.

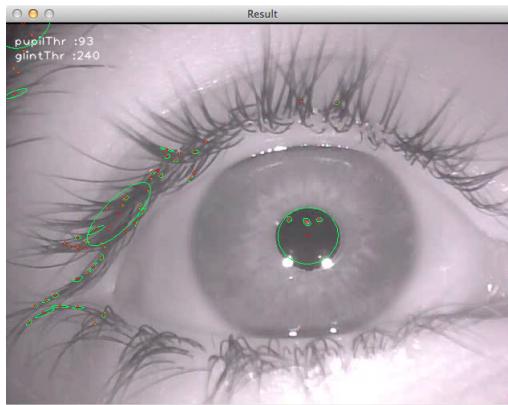


Figure 2: Fitting ellipses on blobs from eye1.avi (green figures are ellipses fitted through blobs, red dots are the centerpoint of each blob)

By experimenting, we find that requiring that the area of the blob lies in the interval [1000 : 10000], and the extent between [0.4 : 1.0], we eliminate most false positives on the entire eye1 sequence, while still keeping the true positive.

This approach has several problems, however. Note how the true positive on fig 2 fails to follow the bottom of pupil correctly. This is due to the glints obscuring part of the boundary between pupil and iris. It also makes some sweeping assumptions:

**The pupil has size at least size 1000** If the person on the sequence leans back slightly, the pupil will shrink and we will fail to detect it.

**A threshold of 93 will cleanly separate pupil from iris** This is true for eye1.avi, but generalizes extremely poorly to the other sequences. If this approach is to be used across multiple sequences recorded in different lighting conditions, the threshold will have to be adjusted by hand for each one.

This problem can be mitigated somewhat with Histogram Equalization. A threshold of 25 on Histogram Equalized images, fares considerably better across several sequences. Note that this will still fail, if parts of the image are significantly darker than the pupil, thereby messing up the equalization.

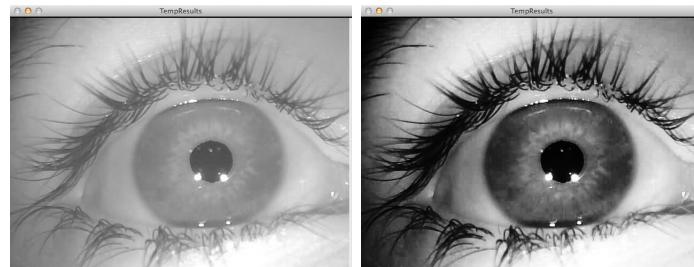


Figure 3: Eye1 before and after Histogram Equalization

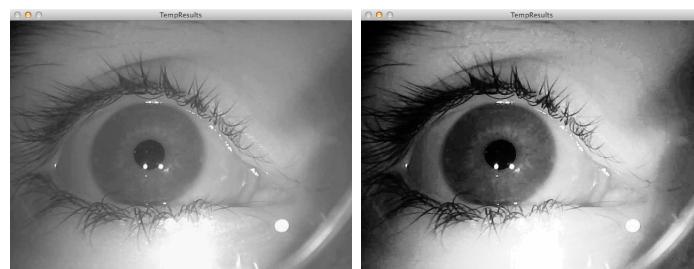


Figure 4: Eye3 before and after Histogram Equalization

**Morphology** Using Morphology, we can improve the detected pupil. The problem with the glints obscuring part of the boundary can be mitigated with the 'closing' operator - used to fill in holes in binary images. Fig 5 shows binary images before and after applying the closing operator. Notice how the noise inside the pupil is completely removed, and the glints are mostly removed. A downside to using the closing operation, is that adjacent, sparse structures may merge and resemble circles.



Figure 5: Eye1 before and after Closing (5 iterations, 5x5 CROSS structuring element)

**Tracking** The pupil tracker can be further improved, by using information about the pupil positions from the previous frame. We do it as follows:

1. Search within some threshold distance from each pupil in previous frame
2. One or more pupils were found within the distance, return those
3. No pupils were found within the distance, search the entire image

Because of the fallback clause in '3', it is very unlikely that the true positive is not detected in each frame. The only case I can think of where this approach fails, is if the pupil is obscured for a frame (subject blinking for example), while a false positive is still detected. In that case, the pupil will improperly detected for as long as the false positive continues to be present.

## 2.2 Pupil Detection by circular Hough transformation

In an attempt to make our pupil detection more robust we now investigate the result of applying a circular hough transformation on the eye images.

The main challenge is finding the correct parameters for the process. We consider the following parameters:

**Gauss kernel size** The size of the gaussian kernel that is applied to the image before the hough transformation

**$\sigma$  - value** the standard deviation value used to construct the gaussian kernel.

**Accumulator threshold** The minimum cumulative vote threshold in which some parameters for a circle are considered.

**Minimum and Maximum Radius** The minimum and maximum radius of circles to consider.

The next step is to experimentally find the parameters that yields the best result over all sequences.

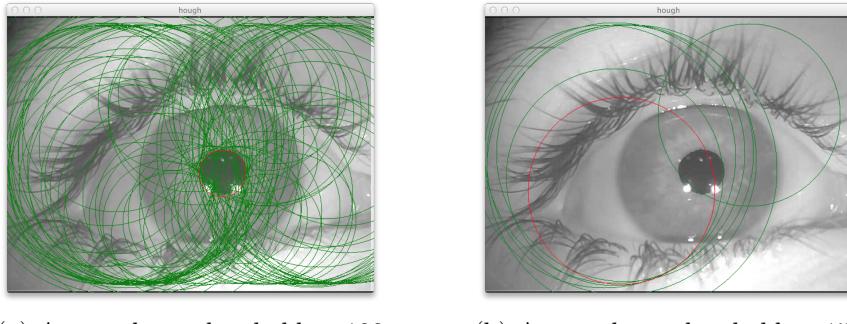


Figure 6: Finding the accumulator threshold values

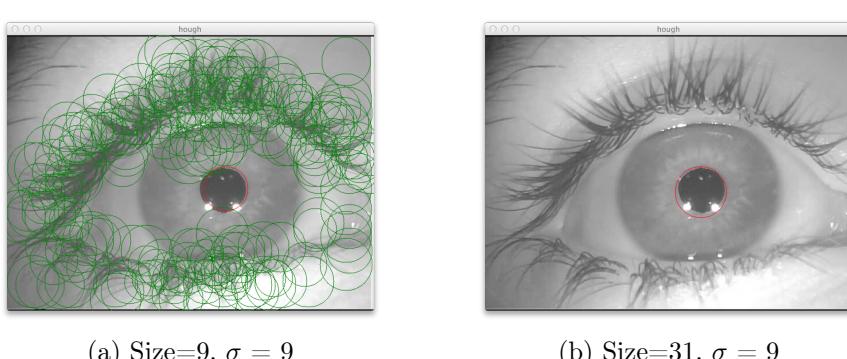


Figure 7: Preprocessing by smoothing with a gaussian kernel

**Discussion** The circular hough transformation yields the most robust pupil detection we have been able to produce so far.

The intuition behind this is that in an ideal setting, where the eye for example is not distorted by perspective, the pupil is going to be near-circular, so the process of hough transforming and then voting for circles is likely to succeed.

In a non ideal setting, for example in a frame were the eye is seen from the side the pupil is not going to yield the same circular properties, and the process is going to fail.

By preprocessing the image by smooting some of the noise is going to be filtered out. The idea is to choose a kernel that is roughly of the same size as the pupil. In this manner smaller features, such as eye lashes will be smoothed away, but still maintaining the pupil feature.

The drawback of setting a constant size for the gaussian kernel is that it is not scale independent. An ideal kernel in some frame may smooth away the pupil in some other frame if the subject moved closer or further away from the camera.

### **3 Glint Detection**

### **4 Eye Corner Detection**

### **5 Iris / Limbus Detection**

### **6 Conclusion**

## **References**

[1] Foo

## **Appendix**