1 - IMPLEMENTATION

Two types of implementation are given for iris segmentation in this paper. First **segmentfull** algorithm includes no optimisation to Daugman's method for iris&pupil segmentation and scans the full image for possible results, while the second algorithm (**segmentopt**) includes optimisations like coarse-to-fine scan, solution to specular reflections and benefits from Theodore&Camus[2] proposals which also speeds-up algorithm runtimes drastically.

For both types of the algorithm entrance needs only two parameters (**rmin,rmax**) to search possible iris locations on the image and calculate integrodifferential operation. First intuition about the dataset to analyse is mostly eyeballs are fit to horizontal axis of the images. We can deduce our min/max radii param. based on this observation. Regarding to the value for the the ratio of the radius of the eyeball to the radius of the iris, 2 is reasonable. So we have following coarse estimations which gives good results running the algorithm;

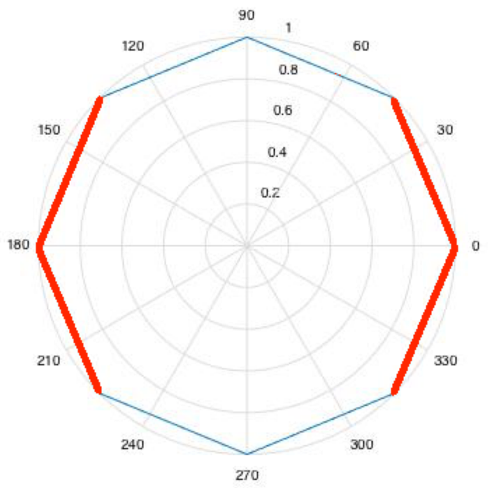
rmin <- min(*ImageWidth*, *ImageHeight*) / 5
rmax <- max(*ImageWidth*, *ImageHeight*) / 3 * 2

Also mostly but not all, particular images in the dataset meets analysis criterions: "*… Subjects to position their own eyes within the field of view of a single narrow-angle camera.*"[1] Some specific images fails to meet these constraints esp. which captured while the person blinking. Daugman's integrodifferential operator which is implemented in the algorithm is given below;

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right| \quad (1)$$

Algorithm implementation uses this operator for both searching iris and pupillary boundaries. As we can see later in the Analysis section, failure to the correct segmentation of the boundaries other than capture motion as stated before can be expressed as: "*The operator in (1) serves to find both the pupillary boundary and the outer (limbus) boundary of the iris, although the initial search for the limbus also incorporates evidence of an interior pupil to improve its robustness since the limbic boundary itself usually has extremely soft contrast when long wavelength NIR illumination is used.*"[1] We will try to compensate this issue in the optimised version of the algorithm (using **segmentopt** function).

The contour (line) integral for both limbus and pupillary circular boundaries given in the equation (1) is calculated by polygon approximation in **contouri** function as stated by Dougman, "*The path of contour integration in (1) is changed from circular to arcuate, with spline parameters fitted by standard statistical estimation methods to describe optimally the available evidence for each eyelid boundary.*"[1]. Also solution for the eyelid occlusion problem for iris segmentation proposed by giving different weight values for the sides and top/bottom section of the iris boundary when calculating the polygon approximation.[3] In the algorithm, n-side for polygon approximation is set as 600.

iris boundary weight scheme for occlusion (polygon approximation with n=8)	
	

After calculating the contour integrals and differentiating with respect to changing size of the radius using integrodifferential operation (**integrod** func.), gradient density fields are smoothed by a LP (gaussian) filter. Then find/search (**search** func.) operation is executed to find max. gradient in order to segment iris boundaries. Same operations take place to select possible pupillary boundaries. Second execution is constrained based on the criterion, *"Although the results of the iris search greatly constrain the pupil search, concentricity of these boundaries cannot be assumed. Very often the pupil center is nasal, and inferior, to the iris center. Its radius can range from 0.1 to 0.8 of the iris radius. Thus, all three parameters defining the pupillary circle must be estimated separately from those of the iris."*[1]. Search operations implements a full-search scan over the entire image.

In addition to these algorithm steps **segmentopt** differs from **segmentfull (full-search)**. Optimised algorithm mostly benefits from Theodore&Camus proposals;

"The algorithm operates in the following fashion. First, bright specularities in the image are filled in to ameliorate their effect on subsequent processing. Second, candidate seed locations are generated to provide initial conditions for the pupil/iris boundary fitting operation. Third, for each candidate seed, (x,y), pupil and iris boundary parameters are recovered."[2] These optimisation techniques can be summarised like below;

<u>Specularity filling:</u>

"The first step of the developed approach is to attenuate the influence of specularities in the image, which can disrupt a pupil-iris or iris-sclera boundary. These very bright areas are formed from reflections off of the cornea of the eye, or off of the lenses of glasses or contacts if present."[2]

In our algorithm implementation we use MATLAB's image morphological enhancement method **imfill** function on negatives of the images in the dataset;

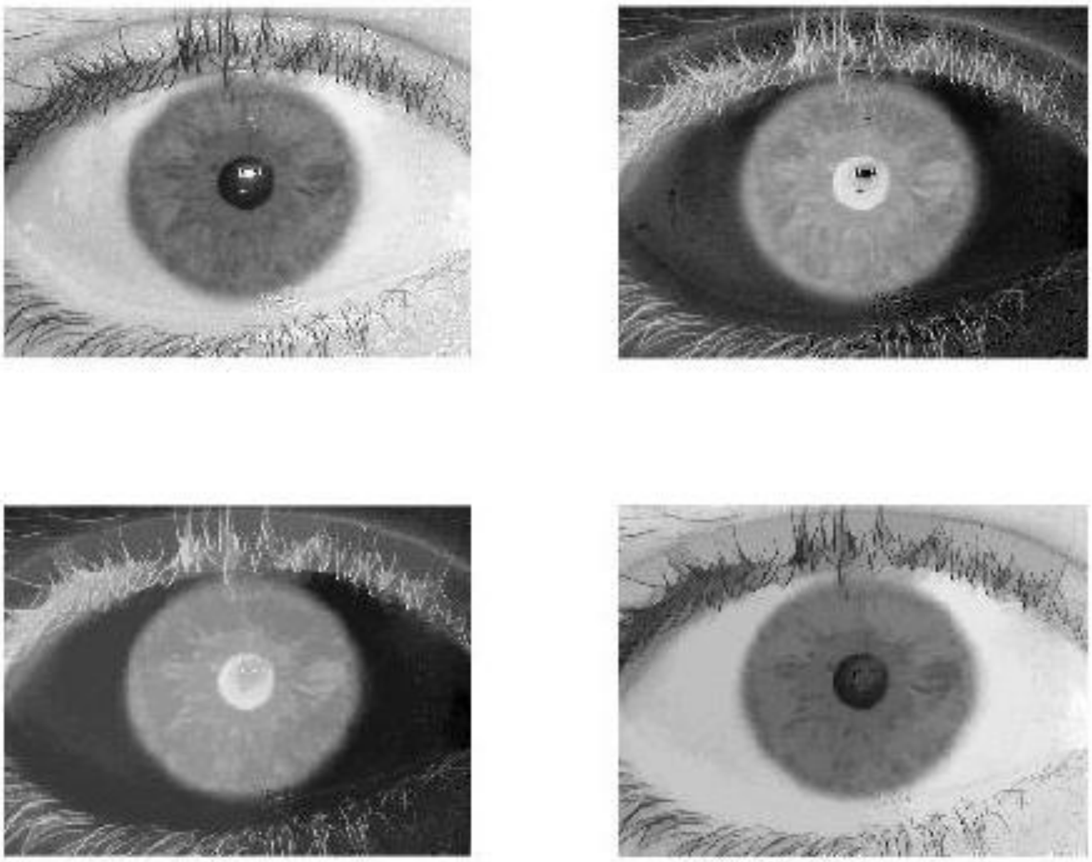
```
% Remove specular reflections by using the
% morphological operation `imfill`
I=imcomplement(imfill(imcomplement(I),'holes'));
```

imfill method is described like following;

> *"fills holes in the binary image BW. In this syntax, a hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image."*[4]

This morphological enhancement enables us to reduce the effect of luminal specularities in an image like reflections esp. around cornea area. A result of the operation can be seen below executed on a sample image;

imfill functions' morphological enhancement



<u>Seed point selection:</u>

"Seed points for the boundary fits are selected as points that correspond to local minima of image intensity. For current purposes, a local minimum is defined as an image intensity value below a certain global threshold and also one that is the smallest value within a 5x5 pixel region. For cases discussed in this paper, the global threshold was set to 128 from a range of [0,255]."[2]

Algorithm is implemented with suggested values to threshold and find local minimums in the image for providing initial conditions for the pupil/iris boundary fitting operations (coarse scan).

<u>Coarse-to-fine scan:</u>

"Images passing a minimum focus criterion were then analyzed to find the iris, with precise localization of its boundaries using a coarse-to-fine strategy terminating in single-pixel precision estimates of the center coordinates and radius of both the iris and the pupil."[1]

"A multi-resolution coarse-to-fine search approach is used, seeking to maximize gradient strengths and uniformities measured across rays radiating from a candidate iris or pupil's central point."[2]

After selecting the seed points in before step, for each candidate seed pupil and iris boundary parameters are recovered in 15x15 window scans (fine scan).

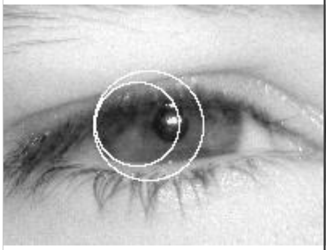
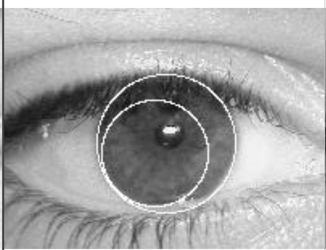
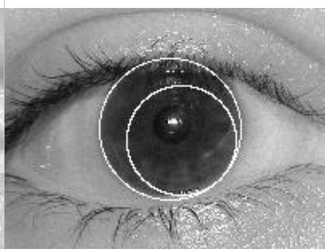
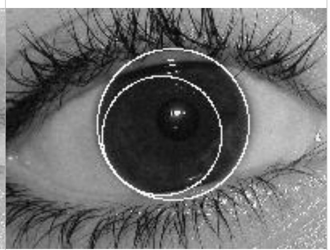
2 - ANALYSIS

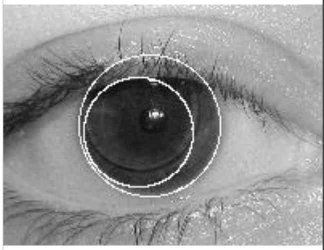
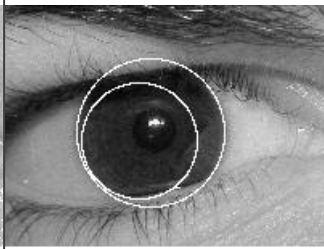
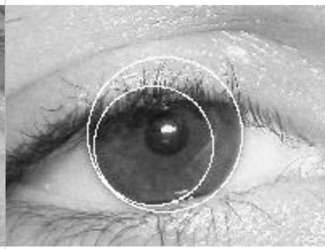
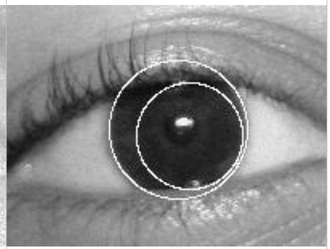
Given algorithms are tested on UBIRIS.v1 database[5]. While full-search scan (**segmentfull**) algorithm runs about ~25-30 secs. for each image, it takes ~3 secs. for optimized algorithm (**segmentopt**). The algorithm's total running time for 241 images in the dataset is about ~1100 secs. (2,53 GHz Intel Core 2 Duo, 8 GB 1067 MHz DDR3 machine).

Incorrect segmentation results are shown below for two different sigma values;

segmentopt incorrent segmentations - filenames	
(sigma=0.5)	(sigma=1.0)
-iris segmentation (total=2) Img_28_1_1_out Img_4_1_1_out -pupil segmentation (total=8) Img_200_1_1_out Img_203_1_1_out Img_207_1_1_out Img_211_1_1_out Img_223_1_1_out Img_232_1_1_out Img_43_1_1_out Img_80_1_1_out	-iris segmentation (total=1) Img_28_1_1_out -pupil segmentation (total=7) Img_103_1_1_out Img_207_1_1_out Img_211_1_1_out Img_223_1_1_out Img_232_1_1_out Img_43_1_1_out Img_80_1_1_out

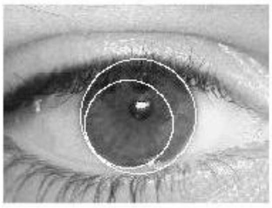
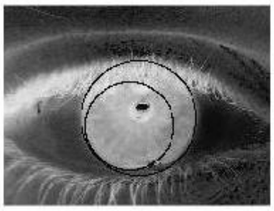
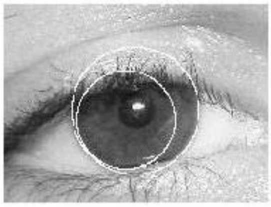
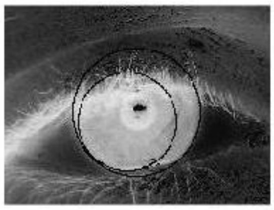
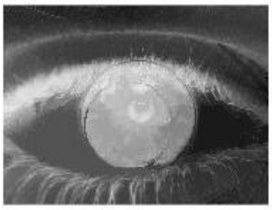
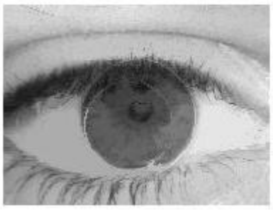
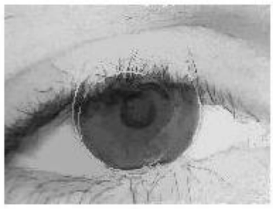
As expected more smooth gaussian filter (sigma=1.0) gives slightly better results for max. gradient selection, hence better segmentation.

segmentopt incorrent segmentations - images			
Img_28_1_1_out	Img_103_1_1_out	Img_207_1_1_out	Img_211_1_1_out
			
Img_223_1_1_out	Img_232_1_1_out	Img_43_1_1_out	Img_80_1_1_out

segmentopt incorrent segmentations - images			
Img_28_1_1_out	Img_103_1_1_out	Img_207_1_1_out	Img_211_1_1_out
			

- For incorrect pupil segmentations, besides "Img_103_1_1_out" and "Img_43_1_1_out" images don't have radical gradient changes in their intensity values around pupillary boundaries. From this viewpoint, we can propose improvements to specularity filling technique in our algorithm. Since it's very coarse in nature.

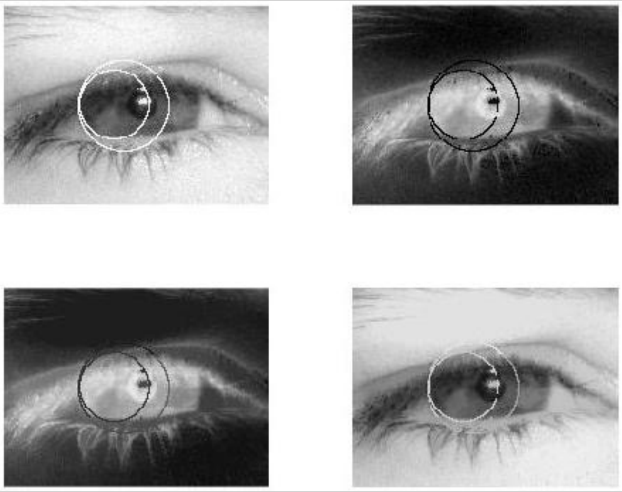
- Still, incorrect pupil segmentations for files "Img_103_1_1_out" and "Img_43_1_1_out" seems problematic. Second step of the algorithm - after iris segmentation - searches for the pupillary boundary in 15x15 neighbourhood of the iris center, which falls inside both of the pupils. But as we can see below, shapes of the pupils seem slightly distorted after specularity filling step. And also pupillary boundary calculation include iris boundaries in these examples. We should not allow this misstep in our algorithm by selecting window size (15x15) relatively according to image size and considering [0.1-0.8] iris radius value.

Img_103_1_1_out (filled)		Img_43_1_1_out (filled)	
			
			

- For the only case of incorrect iris segmentation as seen in file "Img_28_1_1_out" the first realisation is image is taken while the person is blinking. The shape of the iris boundary doesn't give much data for contour integral calculation and also wrinkles in the eyelids also included for calculation which gives a bad result.

Iris segmentation

Img_28_1_1_out (filled)



REFERENCES

[1] John Daugman , “How Iris Recognition Works”, Prec of 2002 International Conference on Image Proccessing, Vol 1., 2002.
[2] Theodore A. Camus & Richard Wildes, “Reliable and Fast Eye Finding in Close-up Images”, Pattern Recognition, 2002. Proceedings. 16th International Conference, Volume:1 2002.
[3] Anirudh Sivaraman, “Iris segmentation using Daugman's integrodifferential operator”, 20 Jul 2007 (Updated 31 Jul 2015)
[4] http://www.mathworks.com/help/images/ref/imfill.html
[5] UBIRIS.v1 database (http://iris.di.ubi.pt/ubiris1.html)