Automatic Temporal Eye Model Fitting
Lech Świrski & Neil Dodgson

Motivation
We wanted to do gaze estimation on a head-mounted eye-tracker without requiring the user to do any calibration. We developed a temporal approach which, given a set of eye images taken over some time period, automatically fits and refines a simple 3D eye model. We then find the gaze using this model. Our approach only relies on pupil shape, and does not require glints, controlled lighting, multiple cameras or user input.

Overview
Our approach uses an existing frame-independent pupil detection to find an initial 2D pupil location estimate. We individually unproject these, and combine them to initialise a 3D model. We then refine the 3D model by optimising the eye position, gaze angle and pupil size to best fit the original image data.

Pupil Detection
We use our existing pupil detector which finds a best fit for a pupil ellipse calculated from image edges in the pupil region.

Pupil Edge Distance Optimisation
Initially, we individually unproject each 2D pupil ellipse into a fixed-radius circle in 3D, by finding circular intersections of a cone going through the ellipse. This gives two possible circles, symmetric about the major axis of the pupil ellipse.

Pupil Ellipse Unprojection
Unprojecting this ellipse gives a best fit for a pupil ellipse, and we can then use this to find the gaze using this model. We then find the gaze using this model. Our pupil detection gives, as output, the edge pixels which were used to calculate that ellipse.

Pupil-Contrast Optimisation
We intersect the minor radii of the pupil ellipses to find a projection eye centre, and unproject that at a fixed distance. We then refine the 3D model by optimising the eye position, gaze angle and pupil size to best fit the original image data.

Pupil Centroid initialisation
We intersect the minor radii of the pupil ellipses to find a projection eye centre, and unproject that at a fixed distance. We then refine the 3D model by optimising the eye position, gaze angle and pupil size to best fit the original image data.

Pupil Centroid optimisation
The elliptical model, with the eye position optimised using a least-squares solver over all the eye images, minimising the sum of squared distances of the edge pixels from the corresponding projected pupil ellipse.

Optimisation
The previous steps only give an initial estimate of the eye and pupil position. We refine this by parametrising the eye position and pupil positions on the eye, and projecting the pupil circles back as ellipses onto the eye images. We want these ellipses to best fit the image data, while still constraining the pupil circles to all lie on the same sphere. We considered two metrics for calculating how well the ellipses fit the images.

Eye Centre initialisation
We intersect the minor radii of the pupil ellipses to find a projection eye centre, and unproject that at a fixed distance. We then resolve the two-circle ambiguity for each pupil by picking the circle whose normal points away from the centre.

Eye Radius initialisation
We know that the pupil circles all lie tangent to a sphere, so the eye radius is the distance from eye centre to pupil centre along the pupil normal. For each pupil, we find this distance by intersecting the pupil centre projection line with a line parallel to the pupil normal from eye centre. The eye radius is initialised as the average of these distances, and the pupils circles are reprojected to lie tangent to the surface of the eye sphere.

Eyemodel initialisation
We know that the pupils circles all lie tangent to a sphere, so the eye radius is the distance from eye centre to pupil centre along the pupil normal. For each pupil, we find this distance by intersecting the pupil centre projection line with a line parallel to the pupil normal from eye centre. The eye radius is initialised as the average of these distances, and the pupils circles are reprojected to lie tangent to the surface of the eye sphere.

Model optimisation
We know that the pupils circles all lie tangent to a sphere, so the eye radius is the distance from eye centre to pupil centre along the pupil normal. For each pupil, we find this distance by intersecting the pupil centre projection line with a line parallel to the pupil normal from eye centre. The eye radius is initialised as the average of these distances, and the pupils circles are reprojected to lie tangent to the surface of the eye sphere.

Evaluation
We evaluated our algorithm on a novel realistic rendered dataset. We used an existing model of a head from an online database, added eyelashes, and retextured it to approximate the appearance of an eye under IR illumination. We then rendered this using Blender’s path-tracing. This model gives us full control over eye gaze, pupil radius, camera and illumination, and we could therefore use it to obtain an absolute ground truth.

We found that our algorithm gives much higher gaze accuracy than naive unprojection, but lower accuracy than calibrated approaches. We believe that we are approaching the limit of gaze accuracy obtainable from a pupil ellipse, and that further improvements must use additional data, such as calibration.

http://www.cl.cam.ac.uk/research/rainbow/projects/eyemodelfit