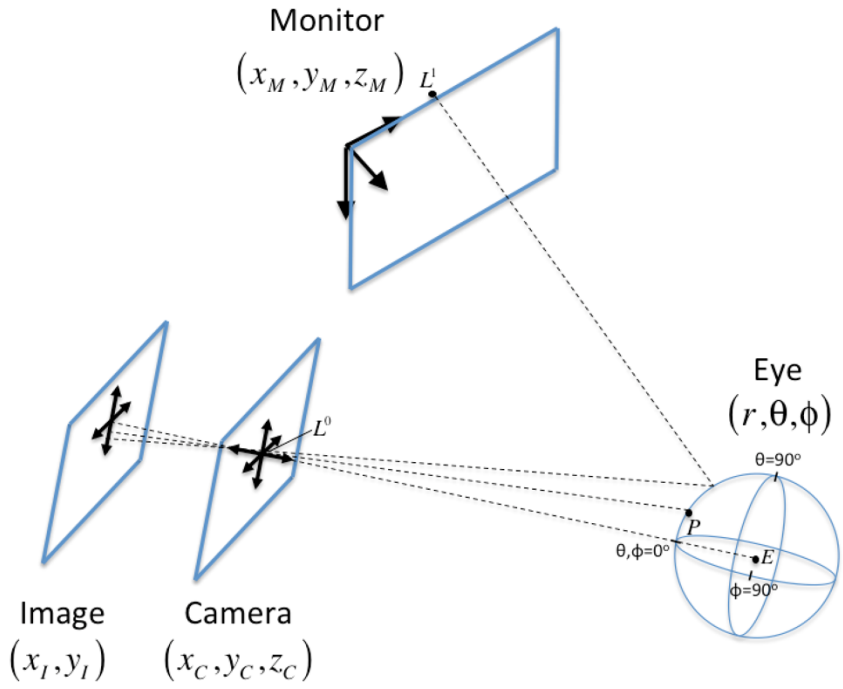
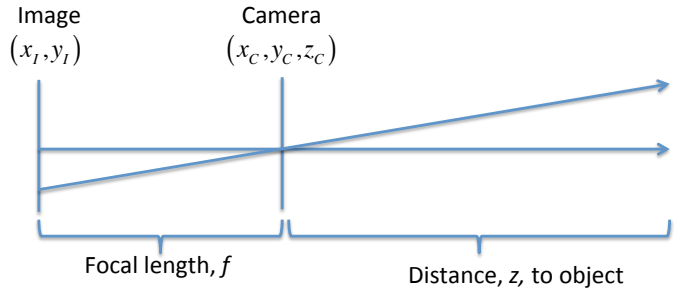
Derivation of Eye Tracking Equations
SAB (5/21)

Definition of coordinate frames. There are four coordinate frames, *M,* the monitor (x_M, y_M, z_M), *E*, the eye, (r, θ, ϕ) in spherical coordinates, *C*, the 3D camera frame (x_C, y_C, z_C), which has an origin at the location of an equivalent pinhole camera, and I, the camera image frame (x_I, y_I). The camera is not assumed to be horizontal, but points to the center of the eye.
The objects that we have are the LEDs L^0 (located at the camera), the monitor LED, L^1 and P, the pupil center. An additional LED can be added for an independent measure, but this is not strictly necessary. Each object can be expressed in each coordinate system, e.g. L^0 in the eye coordinate frame is $L_E^0 = (r^0, \theta^0, \phi^0)$, using the notation that the upper index indicates the object and the lower index indicates the coordinate frame when needed.



Monitor. Define the pixel at the upper left of the monitor as (x_M, y_M, z_M)=(0,0,0) and the direction normal to the plane of the monitor is the vector (0,0,1). Units are in monitor pixels.
Eye. We begin by defining the spherical coordinate system relative to the camera so that the elevation direction of θ = 0 is pointed directly at the camera, (θ = 90 deg. is straight up relative to the camera's view), and for azimuth ϕ = 0 is towards the camera, ϕ = 90 deg. is in the horizontal plane to the left (counterclockwise viewed from above). At the end, the pupil angle can be converted to the body orientation of the animal, but this conversion does not play a role in tracking or identifying the target of gaze.
Camera. Define the location of L^0 as the origin, so $L_C^0 = (x_C^0, y_C^0, z_C^0) = (0,0,0)$.

Image. Define the location of the L^0 reflection in the image as the origin, so $L_I^0 = \left(x_I^0, y_I^0\right) = (0,0)$. Units are in camera pixels. Note that calculations will treat the camera as a pinhole camera, for which the focal length is just the distance from the pinhole to the image. The actual focal length of lenses will end up dropping out of the calculation and will not matter.



Calculation of pupil angle.

1. Define LEDs in Eye coordinates. This will be measured, and the *r* coordinate will not matter.

2. Transform the pupil location from eye coordinates into camera coordinates. The coordinates of the pupil, *P*, in the eye frame are $\left(r^P, \theta^P, \phi^P\right)$. Note that E, the origin of the eye frame, is different from the origin of the camera frame. Converting from simple polar to Cartesian coordinates is done as $(x, y) = (r\cos\theta + x', r\sin\theta + y')$, where (x', y') is the origin of the polar frame expressed in the Cartesian frame. In spherical coordinates, the pupil location expressed in camera coordinates is

$$P_C = (x_C^P, y_C^P, z_C^P) = (r\cos\theta^P \sin\phi^P, r\sin\theta^P, r\cos\theta^P \cos\phi^P) + E_C\,, \qquad (1)$$

with $E_C = (0,0,z_C^E)$. The index P is dropped on the radius because it is the radius of the eye (or rather the cornea). It is important to note that the axis conventions are different from normal here. Usually, the z direction is straight up, corresponding to θ = 90 deg. But here, the z axis of the camera is pointing horizontally. So x, y and z axes are switched from the usual spherical coordinate notation. The usual x becomes our z, y becomes our x, and z becomes our y. In addition, there are two conventions to express the angle θ. One is as elevation, meaning that 0 deg. is horizontal, that is what we are using. The other is that θ is inclination, with 0 deg. pointing straight up. This is more commonly seen, and in this case the $\sin\theta$ and $\cos\theta$ is switched.

3. Transform the pupil location from camera coordinates into image coordinates. Camera coordinates are a 3D system, but the image is 2D. This transformation is done according to $(x_I, y_I) = (x_C \cdot f/z_C, y_C \cdot f/z_C)$. This means that the x and y coordinates are scaled by the ratio of the image and object distance, the usual relationship for magnification. So then the pupil in image coordinates is

$$P_I = (x_I^P, y_I^P) = (r\cos\theta^P \sin\phi^P \, f/z_C \, , r\sin\theta^P \, f/z_C) \, , \qquad (2)$$

with the E_C dropping out because $x_C^E = y_C^E = 0$. Note that z_C is not transformed to spherical coordinates because it will end up falling out of the calculation. To look at a simple example, take the case where the pupil is on the horizontal axis of the eye (θ = 0), and $f = z_c$ .Then in the image $P_I = (x_I^P, y_I^P) = (r\sin\phi^P, 0)$, a simple polar coordinate relationship. When ϕ^P is close to zero (the pupil is facing the camera), the *x* position in the image, x_I^P will change a lot. But when ϕ^P is close to 90 degrees, x_I^P will change little. And the maximum value of x_I^P is r .

4. Transform the LEDs into image coordinates. The camera LED, L^0 is assumed to be at the origin of both the camera and image frames. For the monitor LED, L^1, we again start with equation (1) for the camera, and change it equation (2) for the image, but because the camera is viewing reflections, the angles are halved, compared to what the actual LED angle is. In other words, if the LED is actually at a point 20 degrees above vertical with respect to the eye, ($\theta^1 = 20^o$) , the camera sees the reflection at a point on the eye that is 10 degrees above vertical. So for the monitor LED,

$$L_I^1 = (x_I^1, y_I^1)(r\cos(\theta^1/2)\sin(\phi^1/2) \, f/z_C \, , r\sin(\theta^1/2) \, f/z_C). \qquad (3)$$

Note here that because we are imaging the reflection, r is not the distance in eye coordinates to the actual LED, but the distance to the apparent location of the reflection, which is the same as the distance to the pupil in eye coordinates.

5. Solve for the pupil angle. We now combine equations (2) and (3). First, we have

$$y_I^P = r\sin\theta^P \, f/z_C$$
$$y_I^1 = r\sin(\theta^1/2) \, f/z_C.$$

Solving for θ^P , we have $\theta^P = \sin^{-1}(y_I^P \, z_C/fr)$. Then we can replace z_C/fr from the other equation and get

$$\theta^P = \sin^{-1}\left(\frac{y_I^P}{y_I^1}\sin(\theta^1/2)\right) \quad . \qquad (4)$$

This is equation says that you take the sine of the apparent angle of the LED, scale it by the ratio of the pixel value in the image of the pupil to the monitor LED, and then take the inverse sine to get the pupil angle.

We solve for the azimuth of the pupil in a similar way. First, we have

$$x_I^P = r\cos\theta^P \sin\phi^P \, f/z_C$$
$$x_I^1 = r\cos(\theta^1/2)\sin(\phi^1/2)\, f/z_C.$$

Solving for ϕ^P , we have $\phi^P = \sin^{-1}((x_I^P/\cos\theta^P) z_C/fr)$. Then we can again replace z_C/fr from the other equation to get

$$\phi^P = \sin^{-1}\left(\frac{x_I^P/\cos\theta^P}{x_I^1/\cos(\theta^1/2)}\sin(\phi^1/2)\right) \quad (5)$$

It is written this way to show that it is again the same equation as for equation (4), taking the sine of the apparent angle of the LED, scaling it by the ratio of the pixel value in the image of the pupil to the monitor LED, and then taking the inverse sine to get the pupil angle. Except in this case the pixel values are scaled by the cosine of the elevation. This makes sense, because at the equator, the azimuth will change the position a lot, but near poles azimuth changes the position very little, so a small pixel change means a large azimuth change. To compute this equation, the value of θ^P is taken from equation (4), which is computed first.

6. <u>Identify the monitor pixel intersected by the pupil angle.</u> First, the spherical coordinates of the pupil are converted to a Cartesian vector, by

$$[\boldsymbol{P}]_C = \begin{bmatrix} x_C^P \\ y_C^P \\ z_C^P \end{bmatrix} = \begin{bmatrix} r\cos\theta^P \sin\phi^P \\ r\sin\theta^P \\ r\cos\theta^P \cos\phi^P \end{bmatrix},$$

from (eq. 1). Then the basis must be changed from the camera to the monitor. The basis for the coordinate frame of the camera is $C = \begin{bmatrix} x_C^{e1} & y_C^{e1} & z_C^{e1} \\ x_C^{e2} & y_C^{e2} & z_C^{e2} \\ x_C^{e3} & y_C^{e3} & z_C^{e3} \end{bmatrix}$, with each column being a vector that defines the axes of the camera relative to the standard basis of the table, and the third z vector pointing towards the eye. Here the units (whether mm or cm) will be important. If the matrix M similarly represents the basis of the monitor, but with the units being in pixel dimensions, then through a change of basis the pupil vector in the basis of the monitor is

$$[P]_M = M^{-1} C [P]_C \, , \quad (6)$$

Next, a ray must be traced to intersect with the monitor. The general formula for the intersection of a ray and a plane is:

$$p = \boldsymbol{v}\frac{(p_0 - v_0)\cdot \boldsymbol{n}}{\boldsymbol{v}\cdot\boldsymbol{n}} + v_0 \, ,$$

Where p is the point of intersection, $\boldsymbol{v}$ is the vector representing the ray with starting point v_0, $\boldsymbol{n}$ being a vector normal to the plane and p_0 is a point on the plane. If we

choose the point p_0 on the monitor as the location of pixel (0,0), and the normal vector $\boldsymbol{n}$ is (0,0,1), then the pupil vector $[\boldsymbol{P}]_M$ originating at the eye center E_M in monitor coordinates intersects the monitor at

$$V = \boldsymbol{P}\frac{-E_z}{P_z} + E_M, \tag{7}$$

where E_z and P_z are the z coordinates in the monitor frame of the eye center and pupil vector, respectively, V is the pixel coordinates, and all units are again in pixels.

Finally, if one wishes to convert the pupil vector to the animal frame of reference, one can do so

$$[\boldsymbol{P}]_A = A^{-1}C[\boldsymbol{P}]_C\ , \tag{8}$$

where A is the matrix that represents the basis for the animal's coordinate frame with true horizontal and vertical axes. This also then allows a conversion of $[\boldsymbol{P}]_A$ back to spherical coordinates in the animal's reference frame (with 0^{o} elevation as horizontal and 0^{o} to the front of the animal) as $(\theta_A, \phi_A) = \left(\sin^{-1}P_z, \tan^{-1}\left(P_y/P_x\right)\right)$, to make a correspondence between visual angle and pixels.