

UNIVERSITY OF CALIFORNIA  
RIVERSIDE

Human Eyes – Ray Tracing and Modeling approach

A project submitted in partial satisfaction  
of the requirements for the degree of

Master of Science

In

Computer Science

By

Ulises Amaya

August 2005

Project Committee:  
Dr. Victor Zordan, Chairperson  
Dr. Eamonn Keogh

Copyright by  
Ulises Amaya  
2005

The project of Ulises Amaya is approved:

---

---

Committee Chairperson

University of California, Riverside

# Human Eyes – Ray Tracing and Modeling Approach

Ulises Amaya Master's project  
Advisor: Dr. Victor Zordan

University of California, Riverside  
Computer Science and Engineering Department  
uamaya{at}cs.ucr.edu

## ABSTRACT

*In this project a ray tracer is constructed with the flexibility to specify multilayered objects with properties such as transparency, absorption, shininess, reflectivity and refractivity among others that create an approximate model of a human eye. The iris is modeled using a number of probability and randomness functions to specify the colour and intensity at every point.*

## 1. Introduction

Ray tracing is a computer graphics global illumination rendering method. It traces the rays of light backwards. A ray is shot from a selected viewpoint through a 2D grid or image plane and into a virtual world and finally back to the light source. Intersections with the objects are computed and the pixel is coloured according to a number of factors specified by the objects, the lights and the optical physics laws.

Through various mathematical models which are shown in this document we are able to construct the iris by specifying the RGB value at every point in the corresponding pixel. The ray tracing gives us the intersection, then we get the illumination properties of the material and the mathematical model gives us the RGB value. We also use controlled randomness to model an approximation of what happens in the biological model of the eye.

## 2. Background

Ray tracing has been around for a while now. There has been much research in this CG technique. This technique is an approach on how we see in the real world. Light is generated from many sources, and travels through many mediums. It may bounce around multiple objects as it gets absorbed and scattered. By absorbed we mean that the light as a form of energy is partially transformed to another form. But in ray tracing what we care about is the actual light that gets to our eyes. So in this case we say that it is absorbed and thus we lose some light.

In ray tracing a ray is shot from a selected viewpoint or camera through a 2D image plane and into a virtual world. The ray is traced as it travels and intersects the objects in this world. The closest intersection is stored and secondary rays are spawned from the intersection point. These secondary rays may be reflected rays or refracted rays in the case that the surface is transparent or translucent. Another type of rays named shadow rays is also used. These rays test whether the light sources 'sees' the intersection point directly or not. This is also used for specular light calculations. So for every intersection we spawn a number of rays which then become primary rays themselves and spawn more rays as they intersect other objects and so on making this a recursive ray tracer.

We are using the well known Phong illumination model which assumes constant intensity throughout the scene. The colour and appearance of the intersected point is given by the intensity at this point as well as the contribution from the ambient, specular and diffuse lights.

This model is given by the equation of:

$$I_x = A_x K_a D_x + A_t L_x [ K_d D_x (N \cdot L) + K_s S_x (R \cdot V)^n ]$$

Where  $I_x$  is the resulting colour,  $A$ ,  $D$ ,  $S$ , are the corresponding contributions from ambient, diffuse and specular lights.  $R$ ,  $L$  and  $V$  are the reflection, light and view vectors correspondingly and  $N$  is the normal to the intersected point. Notice that the model has been extended to incorporate attenuation as well. We will talk more about this during the description of the model.

### 3. Extended Algorithm

The basic algorithm that we use is ray tracing. The rays are traced up to a 3<sup>rd</sup> level of depth. This level was chosen arbitrarily, a depth of 5 was also tested with no visible results. It is to notice that in ray tracing, intersections and depth level contribute considerably to the time complexity. This algorithm was extended by including attenuation and absorption. Attenuation simulates the scattering of light as it travels through the environment. Absorption is the amount of light being 'lost' as it goes through a medium inside an object.

The latter is the one that contributes more to the look of the eye. The first is included so that the system is more complete in case that the animator wants to include objects at very different depth distances. Attenuation is basically  $1/\text{distance}^2$ . Absorption is modeled using Beer's Law:

$$\text{Light out} = \text{light in} * e^{-(e * c * d)}$$

where  $e$  inside the parenthesis is the amount that the medium absorbs,  $c$  is the density of the medium and  $d$  is the distance that the light has traveled.

Constructive Solid Geometry (CSG) is also incorporated in the system. Using CSG we are able to construct each part of the eye separately. This is done so that we can specify for each part a different index of refraction (IOR), as well as coefficients for the amount of refraction, reflection, attenuation, absorption, colour and shininess. This allows

us to create a virtual eye which is similar to the biological eye. Figure 1 illustrates a cross section of the eye so that the reader becomes familiar with it.

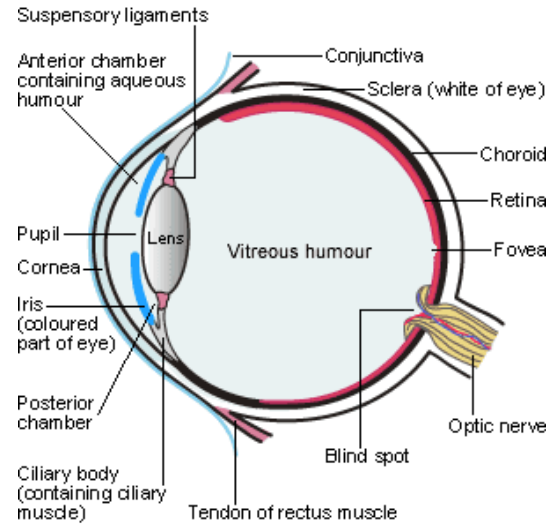


Fig. 1 A cross section of the human eye

#### 4. Construction of the Eye

We begin by using CSG to construct the sclera with two layers. The top layer allows light to go through a bit more than the lower one. Both have the same IOR and same white colour. Then we continue to construct the cornea. Again it has two layers and both are mostly transparent. The outer layer has the IOR of the cornea found in the literature, which is of 1.37. The second layer acts as the interface between the cornea and the iris. In the real eye there is a small amount of aqueous humor which has a different IOR and so the second layer has the IOR of the aqueous humor which is of 1.33. Following is the construction of the retina which is modeled very similar to the sclera.

The main difference is that the absorption is higher and the colour is red since it is the one mainly made out of blood vessels. We now arrive at the lens. The lens is the other

main part of the eye that is responsible for refracting the light after the cornea. It was modeled as close to the real one as possible. The literature shows human lens as slightly non symmetric. The created lens has IOR of 1.4 and is almost fully transparent. The lens again has two layers which make it possible to change the IOR from the lens to the aqueous humor. Here we show a few images illustrating the different parts of the virtual eye using CSG.

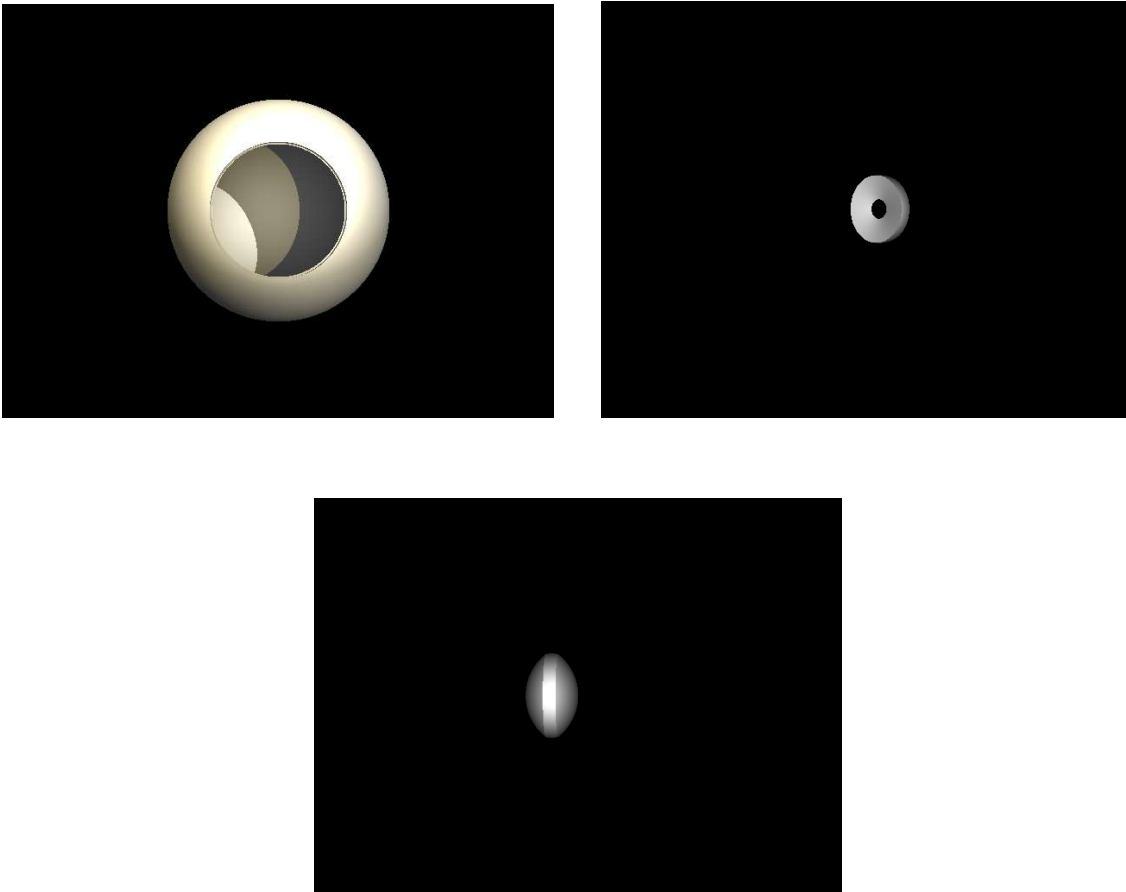


Fig 2. Sclera, Iris and lens created using CSG. The sclera is the white part of the eye which is hollow and will host the iris and lens. The iris base is a truncated cone with a hole in the middle and the lens is slightly non-symmetric having one side bigger than the other.



## 4.1 Iris

The iris is the most interesting and complicated part of the eye. With its complicated patterns which are unique to every individual and the colours that embellish these patterns, we have had a tough time finding the right mathematical functions that would allow us to model the iris. The iris base is a cone-shaped disc shown in figure 1. The colours are given to it depending on the position of the pixel. Basically for every pixel we compute, through a series of functions that we will describe now, its corresponding colour. First there are a few things to notice. By looking at various real examples we can observe that close to the center of the eye there is a ring of sharper or slightly different colour than the rest of the iris. Also, that the patterns tend to be radial rather than tangential and finally that there is a darker ring in the contour of the iris as it merges into the sclera.



Fig. 3 A picture of a real iris, notice that close to the pupil there is an area which has a different colour.

The approach we take is to get the pixels in a tangential way, this means that we get all the pixels that lie in certain angle and compute their colour. The colour is computed according to a combination of functions. The first that we will describe is an exponential probabilistic function. This function grows rapidly and allows us to darken

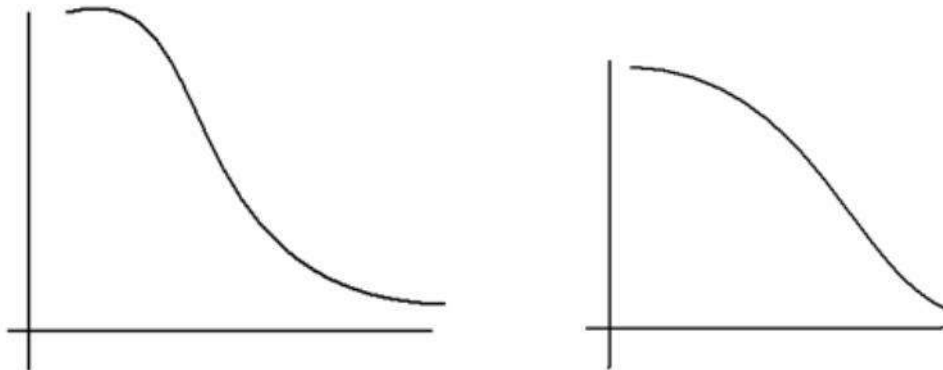
the outer edge of the iris which corresponds to the last 20% of it. Using Gaussian white noise we are able to make this darker ring look more biological as opposed to only a ring by itself. The colours used are blue and green, red is maintained at 0 but by varying the red we can obtain colours such as for brown eyes. This function is described by the following equation.

$$P = (1-P) * (1 - e^{(I * (-1 * k))}) + P$$

Where P is the probability, I and k are initial parameters. The next two functions are the ones that give colour to the rest of the iris.

$$\text{Colour 1} = \frac{-A(C * t - C * x) - A(-C + C * x)}{A(C * x) - A(-C + C * x)}$$

Where A is the *arctangent*, C is a constant set by the user, t is the radius from the center of the pupil and x is a random number between 0 and 1. By varying the constant C, the user can make the function decrease at a different rate. Fig 4 shows 3 different ways that the function can decrease. This number is chosen randomly and this is how we colour tangentially the lines in the iris. So that the lines are not only discrete the result of 'Colour 1' is multiplied by a cosine function that makes the transition between one line and the next one to be smooth. 'Colour 1' goes from green into blue in a continuous and smooth way.



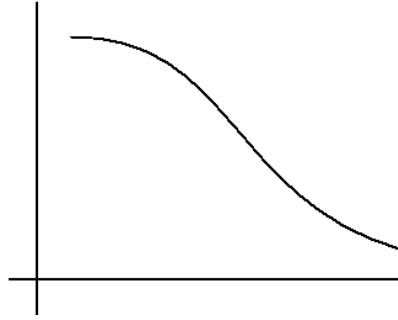


Fig 4. 3 different ways in which Colour 1 can go from green into blue

With all of the above we achieve an iris which goes smoothly from green into blue, that changes intensity depending on the radius distance and that near the outer ring it darkens quickly. The tangential lines are obtained the same way by working with the arctangent but this time with more precision. Fig 5 shows examples of high quality renderings. The blue reflection on top is a sphere in the world not shown in the image.

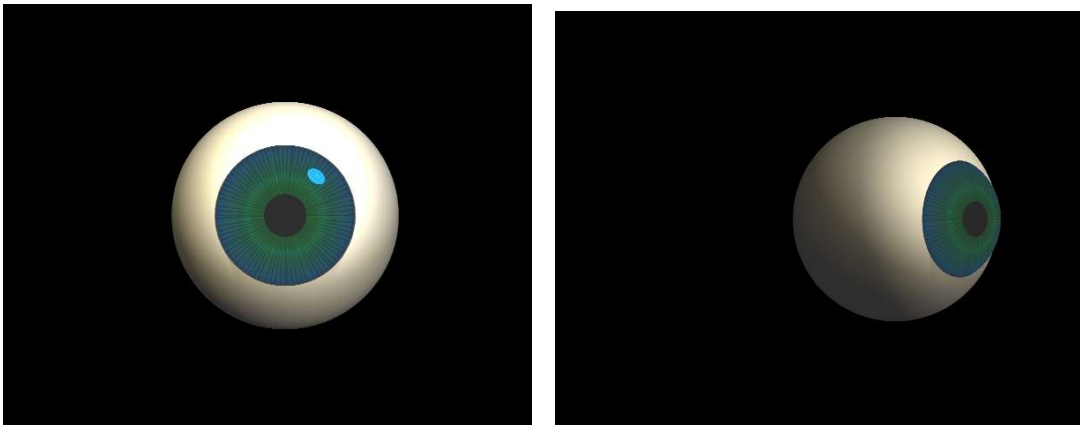
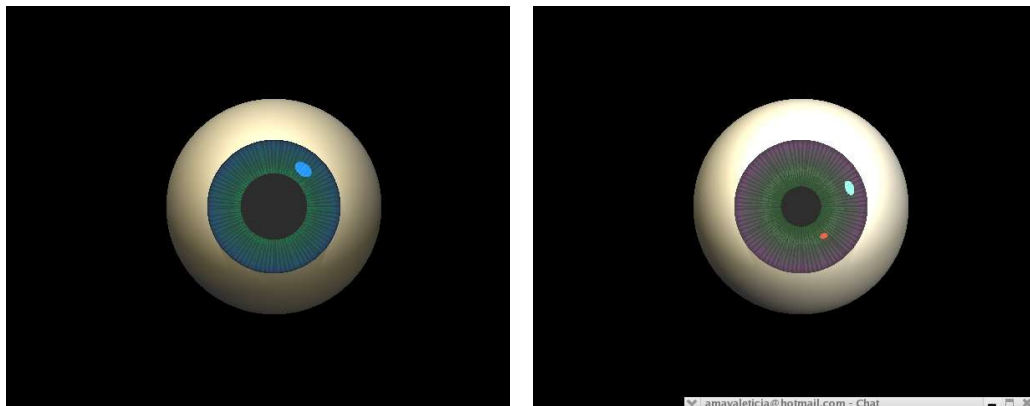


Fig 5. Final rendering of the virtual human eye

## 5. Discussion and Results

We have presented an approach to create virtual human eyes with a combination of a ray tracer and a mathematical model for the iris. This allows the flexibility to create different colours for the iris, eyes with a contracted or dilated pupil, different looks depending on the absorption of the iris or other parts of the eye, red eyes like the ones in the pictures with flash, and eyes with colours that are non-human. The code is implemented in C++ and allows for easy change of functions that model the eye and straight forward modification of the characteristics of the parts of the eye layer by layer such as shininess, absorption, ambient, diffuse and specular components, amount of refraction and reflection, index of refraction and density of the material. It is also important to mention that super sampling is incorporated in the code which allows the user to specify the number of rays shot through each pixel in the image plane. The high quality renderings were made with 16 rays per pixel and the computation time is of the order of 15 minutes on a P IV machine with 1Gb of RAM. The code is not optimized and can be done so by using a KD-Tree data structure. The code only incorporates as primitives the cone, sphere and plane, a cube could also be desirable to make more and different models. The code can be further benefited if a better illumination model is incorporated as well as more functions to model different types of irises such as the one of a cat. Figures 6 and 7 show different types of results and how the eyes would look with the surrounding environment of flesh and eyelashes. The surrounding environments were not created by us and are used only for illustrative purposes.



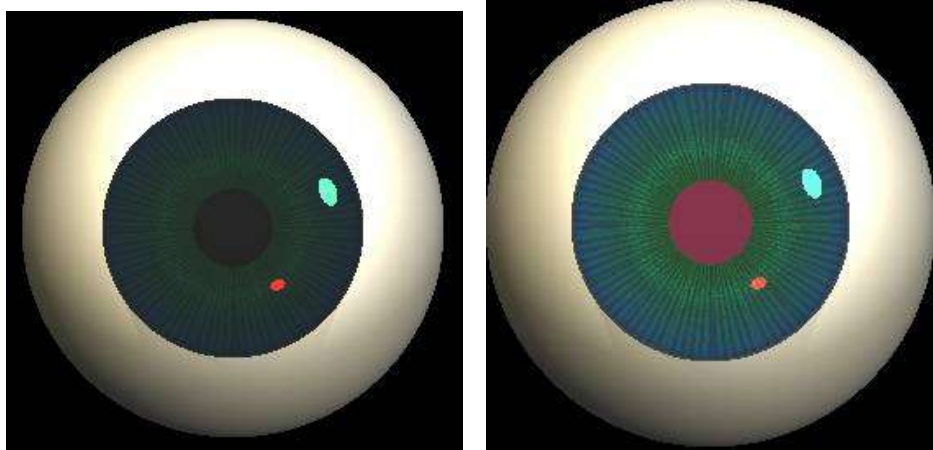


Fig 6. Final renderings of a dilated eye, different colours, a red-eye like the ones that appear in a flash photograph, and an eye with exaggerated density.

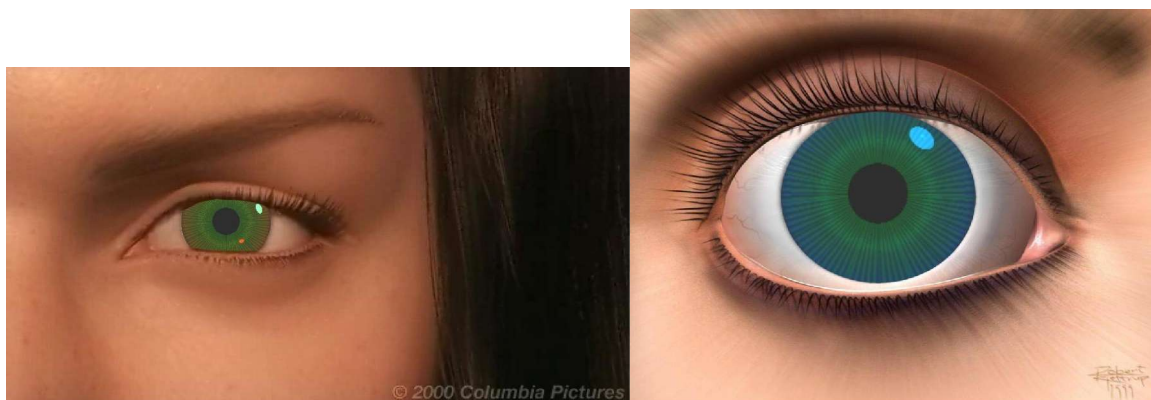


Fig 7. The iris is incorporated in virtual environments not created by me to show that it really gives a feeling of a natural eye.

## Acknowledgements

I would like to thank Dr. Victor Zordan for his guidance during the course of this project. The software discussed in this report was built on top of the CS130 undergraduate computer graphics project by Mikiko Matsunaga. I would also like to thank all that participated in the scientific discussions for their valuable input.

## References

1. The human eye: structure and function Clyde W. Oyster, 1999.
2. Halliday & Resnick, Fundamentals of Physics, 3E, Wiley 1988
3. Computer Graphics: principles and practice. J. D. Foley. 1995
4. Ray Tracing information:  
[www.siggraph.org/education/materials/HyperGraph/raytrace/rtrace0.htm](http://www.siggraph.org/education/materials/HyperGraph/raytrace/rtrace0.htm)
5. Jennifer Beers, *The RT Ray Tracing Project*, University of Washington, Bothell.
6. Beer's Law [www.shu.ac.uk/schools/sci/chem/tutorials/molspec/beers1.htm](http://www.shu.ac.uk/schools/sci/chem/tutorials/molspec/beers1.htm)
7. Eye structure images and information:  
<http://www.mydr.com.au/default.asp?Article=3429>  
[http://www.tedmontgomery.com/the\\_eye/overview.html](http://www.tedmontgomery.com/the_eye/overview.html)  
<http://hyperphysics.phy-astr.gsu.edu/hbase/vision/eyescal.html>
8. Eye surroundings in fig. 7:  
Aki from the movie "*Final Fantasy: The Spirits Within*"  
[http://www.desktopexchange.com/gallery/albums/3D-wallpapers/3D\\_eyes\\_big.jpg](http://www.desktopexchange.com/gallery/albums/3D-wallpapers/3D_eyes_big.jpg)