A Simulation of Neutrino Interaction in Detectors

Santanu Antu, Bard College – SIST Intern

Introduction

In this project, a simulation of Neutrino Interaction in detectors of different geometrical shapes was produced using ray tracing techniques. Ray tracing is a well-studied algorithm in computer graphics that efficiently computes the reflection of light on different objects to produce a realistic image. For the purpose of neutrino interactions, we only consider simple shapes (cubes, sphere, cylinder).

Basic Algorithm of Ray-tracing

In general, a light source emits rays in all different directions, and the rays get reflected on all the different surfaces around it. However, only a small amount of that reflected light happens to reach our eyes/camera. Tracing all the different light rays emitted from the source is computationally expensive and irrelevant. Hence, instead of tracing all the rays from the light source, we trace imaginary rays from the camera/eye to the light source, and calculate the illumination and color of the pixels based on the objects it hits.

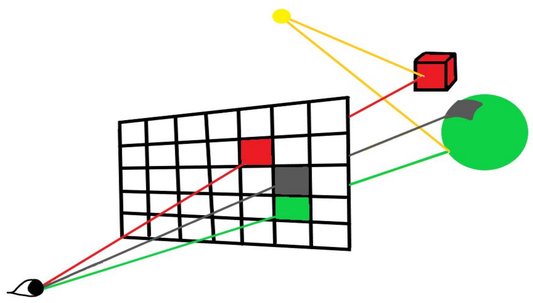

Fig 1: Assigning colors to the pixels based on the rays and the color of the objects

The intersection point of a ray with a sphere can be obtained simply by parameterizing the line and inserting the vector components of it into the equation of the sphere. Let Δ be the discriminant of the resulting second order equation.
We have two situations:

$$\Delta \begin{cases} > 0 & \text{if ray intersects at 2 points} \\ = 0 & \text{if ray is tangential to the sphere} \\ < 0 & \text{if ray doesn't intersect the sphere} \end{cases} \qquad (1)$$

For the last two cases, we simply assign black color to the pixel, and the area is shaded.

The algorithm to find the intersection point between a ray and a cuboid is a little more complicated. For simplicity, a 2D box parallel to the coordinate axes is defined by the minimum and maximum x and y coordinates. Let the minimum and maximum "time" that parametrizes our ray intersect the lines parallel to the coordinate axes be t_{x0}, t_{x1}, t_{y0} and t_{y1}. The idea is that if $t_{y1} < t_{x0}$ or $t_{x1} < t_{y0}$, then the ray doesn't intersect the box.

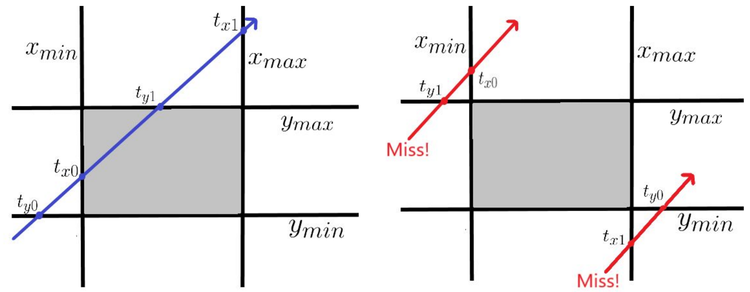

Fig 2: Computation of the intersection point of a ray and a 2D box

We used a 3D version of this algorithm to find the intersection of a ray and a cuboid.

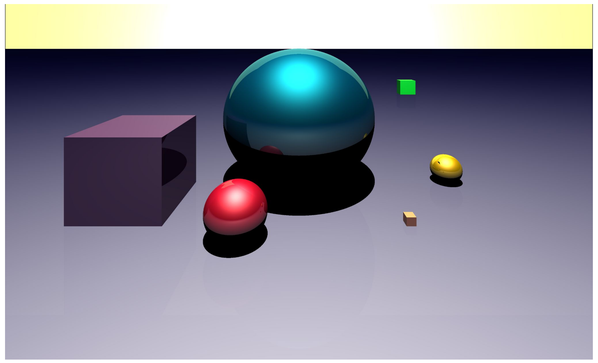

Fig 3: A ray traced image with objects of basic shapes

Application to Neutrino Interaction

Due to the miniscule probability of interaction with ordinary matter, neutrino usually don't reflect off the surface of the objects (different detector components in our case). However, we can use most of the geometrical part of light ray tracing to model how a neutrino would propagate through detectors of different geometries.

Probability Distribution of Interaction Points

If we shoot a number of neutrinos into a material, the number as a function of distance follows an exponential distribution. So, if λ is the mean free path (which depends on the material the neutrino is travelling through), the probability of interaction is given by:

$$P(x) = 1 - e^{-x/\lambda} \qquad (2)$$

Here, x is the distance travelled in the material. We modelled an arbitrary detector with 3 boxes with different materials in them. Then we calculated the expected probability of interaction in each boxes. Finally, we made a simulation with 10000 rays, and computed at which segment they interact in the detector.

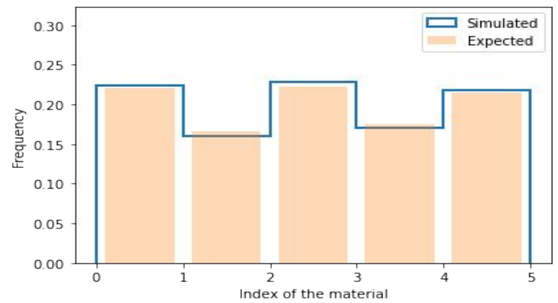

Fig 4: Histogram comparing expected and simulated probabilities of interaction segments

Then we fixed a source at point (0,0,-2) in our detector and threw neutrinos with random y-direction within a 45 degree cone around the z-axis. The angle was chosen arbitrarily. By choosing different angles, we can model different neutrino beams. We calculated the interaction points for the random neutrinos and plotted the YZ projection.

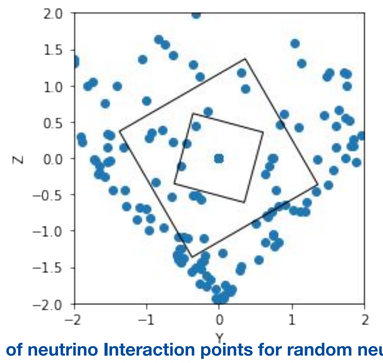

Fig 5: YZ projection of neutrino Interaction points for random neutrinos within 45 degrees of the Z-axis. Different boxes can hold different materials