Interactive Refraction

1: Introduction

My project is an implementation of the paper "An Approximate Image-Space Approach for Interactive Refraction" by Chris Wyman. I am using GLSL shading language to implement the refraction concept.

1.1: Theory

Current GPU-based Interactive refraction algorithms are restricted to render refraction through a single surface. The paper "An Approximate Image-Space Approach for Interactive Refraction" , presents an algorithm that allows refraction of a distant environment through two interfaces.

Algorithm
Here an approximated approach is used to calculate refraction through two interfaces with values easily computable via rasterization. The method proposed uses two passes to compute the doubly refraction. After rasterization, for each pixel, the following information is known

1: the incident direction V
2: the hitpoint P1
3: the surface normal N1 at P1
Using the above information and applying Snell's law, we can easily calculate the transmitted direction $T_1$.

To compute the second refraction direction $T_2$, the point $P_2$, first refraction direction $T_1$ and the normal $N_2$ are necessary. The $T_1$ is obtained using the above method and the other two requirements are obtained by approximating as follows.

**Calculation of the point $P_2$**

The point $P_2$ can be given by the below equation,

$$P_2 = P_1 + dT_1;$$

where $d$ is the distance $\|P_2 - P_1\|$.

Using the value of $P_1$ and calculated value of $T_1$, the value of $P_2$ can be calculated by approximating as follows.

$$\tilde{P}_2 = P_1 + \tilde{d}T_1 \approx P_1 + dT_1$$

where

$$\tilde{d} = \frac{\theta_t}{\theta_i} d_V + \left(1 - \frac{\theta_t}{\theta_i}\right) d_N.$$

where $d_V$ is approximated by the distance between back-facing and front-facing polygons whereas $d_N$ is approximated by the size of the object.

**Calculation of Normal $N_2$**

In the first pass the surface normals are rendered as color and stored into the texture. During the second pass, the approximated point $P_2$ is projected into the texture space and indexed into the texture to find the normal
Calculation of Second refraction Direction T2

Using the calculated valued of N2, P2 and T1, the T2 can be calculated using snell's law. Then the value of T2 is used to index into environment map to find the doubly refracted color.

My Implementation

1: First I created the 3D scene using SkyBoxes and placed the sphere in the center to reflect/refract the scene(finalrag.cpp).

2: Cube mapped the sphere.

3: Wrote shaders(reflection.vert-->vertex and reflection.frag-->fragment) to implement reflection.

4: Implemented single refraction using a set of shaders(refraction_1.vert-->vertex and refraction_1.frag-->fragment) by rendering the scene only once.

5: Then implemented the first pass by rendering back faces using culling and storing normals and depth values into texture memory using shaders(pass_1.vert-->vertex and pass_1.frag-->fragment). I calculated the incident direction in vertex shader and sent it to fragment shader to calculate the refraction direction to get better results.

6: Wrote another set of shaders(refraction_1f.vert-->vertex and refraction_1f.frag-->fragment) to implement single refraction with fresnel terms. First I calculated reflection term and then I got refraction term by subtracting the reflection term by 1.
7: Then implemented the second pass to calculate the second refraction using the above described method using a set of shaders(refraction_2.vert-->vertex and refraction_2.frag-->fragment). I calculated $n_i/n_t$ (refractive index of air/refractive index of sphere) value in opengl program and passed to the shaders as uniform variables.

8: Wrote shaders(refraction_2f.vert-->vertex and refraction_2f.frag-->fragment) for second refraction with fresnel terms.

9: Then I coded to provide simple GUI as shown in the below images.

**Results**

Below screen shot shows reflection.
Below screen shot shows single refraction.
Below screen shot shows single refraction with fresnel terms.
Below screen shot shows double refraction.
Below screen shot shows double refraction with fresnel terms.