

Creating More Realistic Lens Effects

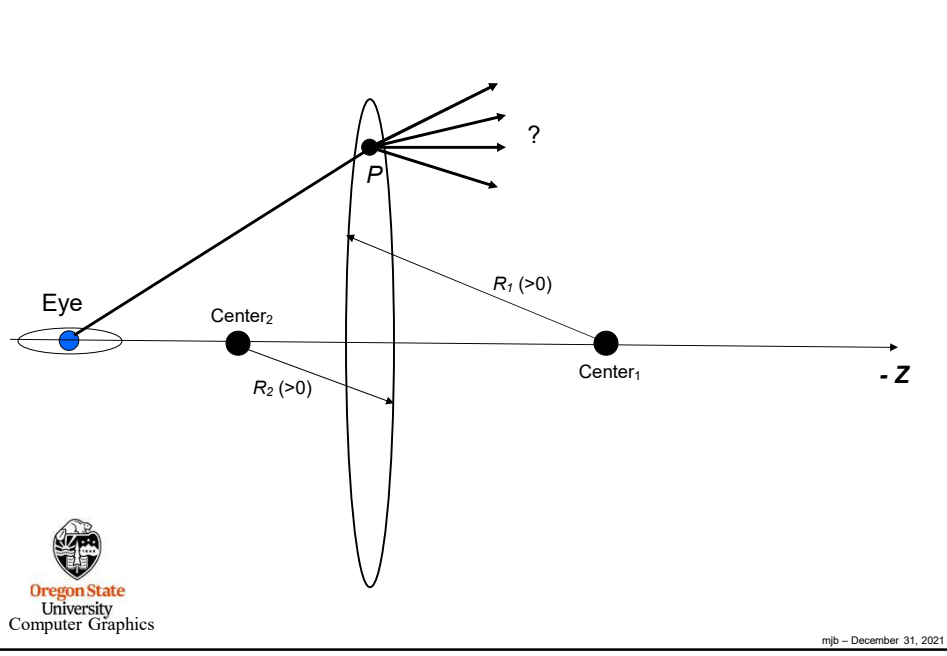
Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu

Oregon State University
Computer Graphics

lens.pptx

mjb - December 31, 2021

Convex Lens Definitions



Concave Lens Definitions

3

The diagram shows a concave lens (diverging lens) with an eye on the left. The optical axis is labeled $-Z$. The eye is represented by a blue circle. The lens is shown as a vertical shape that is wider at the top and bottom and narrower in the middle. Two centers of curvature are marked: $Center_1$ on the left and $Center_2$ on the right. The radii of curvature are labeled $R_1 (<0)$ and $R_2 (<0)$. A point P is marked on the lens surface, and several rays are shown originating from P and diverging. A question mark is placed near the rays.

Oregon State University
Computer Graphics

mjb - December 31, 2021

3

lens.vert - Setup for a Cube Map Texture

4

The diagram shows a lens with an eye on the left. The optical axis is labeled $-Z$. The eye is represented by a blue circle. The lens is shown as a vertical shape with a point P on its surface. A vector $FromEyeToPt$ points from the eye to $Point P$. A normal vector $Normal1$ is shown at P . A refracted vector $v1$ is shown passing through P . A second normal vector $Normal2$ is shown at P . A red double-headed arrow indicates the thickness of the lens, labeled "zero thickness".

```

out vec3    vRefractVector;
uniform float  uR1, uR2;
const float   ETA = 0.66;
const vec3    EYE = vec3( 0., 0., 0. );
...
vec3 P = vec3( gl_ModelViewMatrix * gl_Vertex );

vec3 FromEyeToPt = normalize( P - EYE ); // vector from eye to pt
vec3 Center1 = vec3( 0., 0., P.z - uR1 );
vec3 Normal1;
if( uR1 >= 0. )
    Normal1 = normalize( P - Center1 );
else
    Normal1 = normalize( Center1 - P );

vec3 v1 = refract( FromEyeToPt, Normal1, ETA ); // eta = in/out
v1 = normalize( v1 );
vec3 Center2 = vec3( 0., 0., P.z + uR2 );
vec3 Normal2;
if( uR2 >= 0. )
    Normal2 = normalize( Center2 - P );
else
    Normal2 = normalize( P - Center2 );

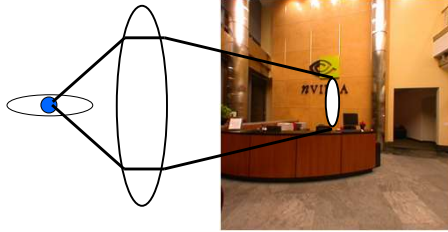
vRefractVector = refract( v1, Normal2, 1./ETA ); // 1./eta = out/in
gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;

```

Oregon State University
Computer Graphics

4

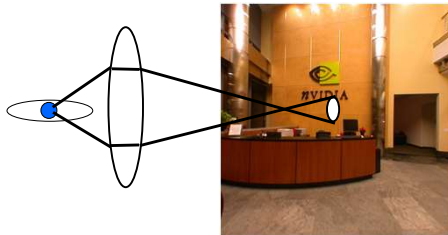
Convex Lenses ($R1 > 0, R2 > 0$)



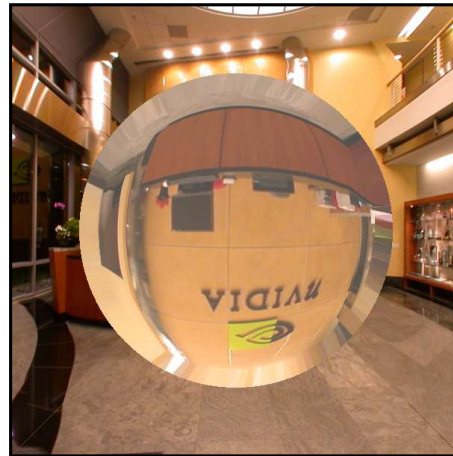
Magnifying glasses and zoom lenses work this way



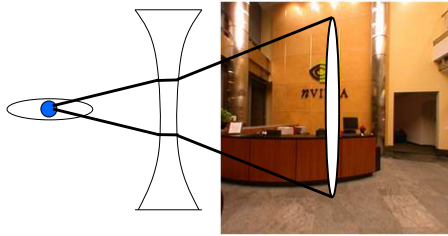
Convex Lenses ($R1 > 0, R2 > 0$)



Some telescopes work this way



Concave Lenses ($R1 < 0, R2 < 0$)



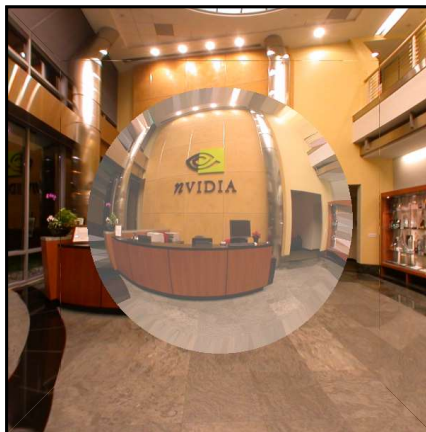
Fisheye lenses work this way



mjb - December 31, 2021



Convex



Concave



mjb - December 31, 2021