The Principles of Ray Tracing

Exercise No. 1 – Submitted 31 May 2010

Jan Wieferich
Matr.: 2455022, University of Bremen
jan.wieferich@uni-bremen.de

Introduction

The principle of ray tracing (or backward ray tracing) is introduced best at the example of the camera obscura, an analogue imaging technique. In short, based on a pinhole, the outside world is projected upside down on the opposite wall of the camera. As the rays of light from the outside scene are traveling in straight lines through the pinhole they cross instead of scatter and are projected upside down as an two-dimensional representation of the outside world.

In digital imaging, especially in generative computer graphics, the same optical principles are used to generate an two dimensional representation of an three dimensional scene. While digital cameras are using image sensors to read the intensity of light at one picture element (pixel) the generative approach of ray tracing is based on the other way around to compute an two dimensional representation of an artificial three dimensional scene, fully generated by computing from the starting point of the virtual camera.

One key problem during the computation of artificial images is that it is not reasonable to compute every physical feature of light in detail (e.g. light photons). Thus the attempt of ray tracing is an approach of approximation. Based on the target of ray tracing, the final image defined by the virtual camera, for each pixel the way of light is traced from the camera into the virtual scene.

1. The Ray

This backward approach takes the geometry of the virtual scene into account, computing the intersections as points in space. The ray could be mathematically represented as a line from origin of the virtual camera (better said the projection reference point) through a pixel into the scene geometry, for each pixel of the image to be computed.

At an intersecting point the color of the object hit by the ray is stored (also the transparency) and new rays are started at this intersection point, tracing further intersecting points or a light source. Thus the color of the light that left this source is retrieved and also the colors of each intersecting object are taken into account to finally compute the color of the pixel. Thus the ray is not only defined as an direct ray from the light sources also the indirect parts of illumination are taken into account (reflections, refractions and shadows).

Furthermore, not every ray could be traced to the light source or to an intersection point within the scene geometry. A pixel that is not hitting the scene geometry nor the light source is not illuminated.

2. Primary Rays

In case of an intersection hit the principle of ray tracing includes an distinction of rays, computed as an recursion. Prior the **primary rays** a generated from the origin of the camera through each pixel of the projection plane. Thus a primary ray can hit a light source, an object within the virtual scene or nothing. This first ray shoot into the virtual scene is also called camera ray or visibility ray. The first object hit of the primary ray represents the object (or its part) visible to the camera.

3. Secondary Rays

In case that a primary ray hits an object within the virtual scene an intersection point of this object is found. This means that this point hit is within a shadow, not directly illuminated by an light source. To retrieve the diffuse amount of light illuminating this point three new rays are generated at this spot, the **secondary rays**. The first type of a secondary ray represents a shadow ray, the second a reflection ray (e.g. the object hit is a mirror or shiny surfaces). The third kind of a secondary ray is a ray of refraction, in case the object hit within the scene is a translucent object (e.g. glass).

3.1 Shadow Rays

At a intersecting point a new ray is generated in the direction of the light source which is called **shadow ray**. In case this ray hits another opaque object in this direction a shadow is determined for this intersecting point of the scene and its intensity is set to zero. Otherwise the color of the intersection point is taken into account.

3.2 Reflection Rays

Secondly a additional ray is generated if the material of the object hit is defined as reflective. This **reflection ray** might hit another object at an intersecting point of the scene. The direction of this ray is defined by the law of reflection that states that the angle of incidence equals the angle of reflection.

3.3 Refraction Rays

If the object hit at the point of intersection is defined as translucent (color with transparency) an additional **refraction ray** is generated. Its direction is defined by the Snell's law of refraction, similar to the law of reflection but not bouncing of the object hit. Refraction is referring to light passing through an object. While passing through a translucent object the way of light is bend along its path. Thus the way of light through different media could be taken into account (e.g. water).

4. Recursion

The algorithm of the ray tracing recursion (also named recursive ray tracing) starts with the primary rays to achieve intersection points within the scene. Based on the intersection points found within the scene additional secondary rays are generated as new lines through the scene, starting at the intersection point found by the primary rays.

For each new (closest) intersection found by the secondary rays of reflection and refraction the recursion is repeated and new sets of rays are generated through the scene as long as no termination condition is achieved. Firstly, the ray tracing recursion terminates if no intersection with an object within the scene could be determined. Furthermore the amount of rays to be traced is limited by the depth of recursion, so the computation is aborted by a predefined maximum of intersection points found and secondary rays generated.

4.1 Ray casting

A non-recursive variant of ray tracing is only generating primary rays to determine the first object intersected by a ray of light. Thus only direct illumination of the objects are taken into account to calculate the color of the each pixel of an image and ray casting never traces secondary rays, so reflections, refractions and shadows are not taken into account to calculate the color of an pixel.

5. Determination of Object Visibility

By simply casting the primary rays into the virtual scene it could be determined which object is visible to the camera. Firstly, objects which are not intersected by the rays traced from the cameras origin through each pixel of the image won´t be visible. Furthermore, if more than one intersection point is determined by an ray, the first intersection will occlude the second intersected object (in case the first object intersected is opaque). Also the length of the line representing the ray is measurable from the cameras origin to the intersection. Instead of retrieve the objects color and to write it as the color of the final pixel this distance could be stored as an color value to retrieve an depth map of the image generated. This depth map could be used to discard hidden objects.