Abstract

Translucent materials such as skin, marble and wax are hard but necessary to render with realistic results. If not done correctly the result will look computer generated. In this paper a short summary of earlier methods is given and two current methods are described a little deeper. To make a conclusion of what method to use renderings from different methods are included in the report.

1 Introduction

Translucent materials such as skin, marble and wax are hard to render because of their special properties. Light can enter these materials, scatter inside and then exit the material in another direction and point than where it entered. How the light is transported inside an object depends on its geometry and the materials optical properties. To achieve photo-realistic renderings it is important to capture the special light effects such as volumetric caustics and shadows which may appear when light is scattered in a translucent material.

2 Background

Points inside mediums get light intensity from reduced incident light from light sources and diffuse intensity which is light that have been scattered inside the medium. As usual an earlier method is proposed by Jos Stam. In 1995 he published [3] describing a method to approximate multiple scattering with a diffusion process he calls diffusion approximation. He shows an equation for calculating approximated diffuse intensity. With this method it was possible to render highly scattering homogeneous materials. For example realistic looking clouds. Another method by Jensen et al. [4] suggest acquiring a BSSRDF (bidirectional scattering surface reflectance distribution function). The usually used BRDF assumes light entering and leaving material at the same point the BSSRDF is a more exact BRDF and able to handle these materials. In the paper they describe the BSSRDF model as a sum of a dipole diffusion approximation (for multiple scattering) and
Current research is focused on rendering more correct translucent materials than previous methods which are missing effects that real translucent materials create as internal caustics, volumetric shadows and volumetric caustics. Donner et al. [2] handles this using Photon Diffusion. Other problems with earlier methods is the absence of an intuitive way of gathering correct information about materials and editing their properties. Wang et al. [1] propose a method for both gathering material info and rendering in real-time.

3.1 Rendering Translucent Materials Using Photon Diffusion

Donner et al. [2] extends [4] by not only placing one dipole source as in figure 1 (a) but several in the direction of the incident beam (ray). This results in a more accurate result.

Their method can be described as: First use standard photon mapping to trace from light sources towards translucent objects. When hitting object and refracting into the object three different actions can occur. The photon might either hit another interface and refract out of the material again (single scattering), be absorbed or it can scatter multiple times. When multiple scattering occurs they store the position and incident angle of the photon and then continue to trace it until either of the other actions occur. They only store the photon again if it exits the material and enters again. Using photon mapping many important features of translucent materials can be simulated such as caustics, self-lighting and indirect global illumination.

This Photon approach can create millions of dipole diffusion sources. To handle this an octree with a maximum of eight sources per leaf is created. From a leaf a single source is created.

Single scattering is rendered through ray marching through object and valuating the radiance from the photon map. Multiple scattering is rendered from the sources stored in the the octree. When calculating the diffuse reflection they take into consideration the angle between the shading point and the surface normal, distance to surface and thickness of material as well as the incident angle on the surface of the incident photon as in figure 1.

When rendering a shadow ray is sent from the shading point. If the ray intersects with a blocking geometry (same that absorbs photons during the tracing earlier) no contribution from the source is added.
3.2.1 Acquisition of Material Model

They use a high end digital camera to capture different illuminations of materials. A projector is used to illuminate a material using different intensities and angles creating an image sequence which they store in an HDR image. In the image information about multiple scattering is stored. Using a GPU implementation they create the material model from the HDR image by solving the inverse diffusion problem (figure 2 in [1]).

3.2.2 Polygrid Construction

The polygrid is essential for rendering and is created by approximating the polygon object using polycubes. Inside the cubes regular grids are created. Adjacent cubes are connected and different adjustment schemes is used to match the polygrid to the objects boundary. When done a polygrid consisting of boundary and inner nodes is obtained.
3.2.3 Rendering and Editing

During rendering a 2D texture is created from the polygrid where material properties for each node is stored. This makes it possible to linearly calculate the diffusion intensity on the GPU. For efficiency adjacent nodes need to be stored in the same texture. Equation for calculating diffusion intensity used are similar to the ones in [3].

The usage of a HDR image to represent multiple scattering gives the possibility to edit properties by painting desired behavior.

4 Conclusion

Scattering of light when rendering translucent materials is an advanced but important topic. Without correct scattering images tend to look very computer generated. Both methods discussed produce great looking results but it is hard to draw a conclusion of which might be the better method. Result renderings in the papers are hard compare as they have focused on rendering different translucent materials. [1] have the benefit of being a method taking everything into consideration from creating materials to rendering them. It also runs in real-time which makes it possible to edit materials while showing the result. The downside is that it is a complicated method involving many steps compared to [2].

References


