Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory
T-111.500 Seminar on computer graphics L

3.8.2004

A Survey of Real-Time Shading

Miro Lahdenmäki
55089K

A Survey of Real-Time Shading

Miro Lahdenmäki
HUT, Telecommunications Software and Multimedia Laboratory

`mlahdenm@cc.hut.fi`

Abstract

In this paper various real-time shading methods will be examined including Bidirectional Reflectance Distribution Function illumination models, isotropic as well as anisotropic. Different shading models, environment mapping models, bump mapping models and shadow models will be discussed.

1 INTRODUCTION

One of the primary goals and motivations in real-time shading is to create realistic interactive graphics but also non photorealistic effects such as a cartoon look can be pursued. Natural materials and natural phenomena such as fire, smoke, clouds and water are often simulated. Some fields in which real-time shading is used are architecture, computer aided design and modeling, animation and games. In architecture for example real-time shading is used to get a very realistic preview of the designed building before construction is begun. In animation real-time shading helps animators because they are able to view movement instantly instead of waiting from minutes to hours to render the scene.

In real-time graphics you have to keep in mind that performance should be consistent and sufficient to achieve acceptable interactivity. Computing power, memory restrictions and graphics hardware all pose limitations to real-time shading methods.

In this paper various real-time shading methods will be examined. First there is an an overview of real-time shading. Then there is a brief walkthrough on the physics of light and real world material properties. After that BRDFs are discussed and a short look is made at 3D computer graphics basics. After that various subfields of real-time shading are discussed. In the end some conclusions are made.

2 An overview of real-time shading

Real-time shading is composed of a number of different techniques. Modern graphics processing units make it possible to do more complex things in real-time. With the use of bidirectional reflectance distribution functions more realistic materials and phenomena can be modelled. Traditional shading models like Gouraud shading are still in popular use but more and more games use more advanced techniques such as Phong shading, bump mapping, glossy reflections, shadows and post-processing effects.

3 Physics of light

First some basic concepts need to be defined. *Radiance* means the number of photons arriving per time at a small area from a particular direction. *Irradiance* can be seen as the density of the incident flux falling onto a surface. Radiation *flux* signifies the energy per unit time (or power) passing through a surface. *Radiosity* can be seen as the density of the outgoing flux leaving from the surface. The *power* or the total flux of the light source is defined in watts.

- A *point light source* with *isotropic* radiance. Isotropicity means that the attributes of the matter are the same in all directions.
- A *spot light source* is a cone of light. It has the radiation characteristics of $cos^n\theta$, where θ is an angle subtended by a curve in a plane.
- *Area light sources* are point light sources with a non-uniform (*anisotropic*) directional power distribution. Anisotropicity is the opposite of isotropicity so the attributes don't have to be the the same in all directions.

For *atmospheric attenuation* with distance (r) for point light sources an equation $1/(ar^2 + br + c)$ is used. Although physically the correct equation would be $1/r^2$. The replacing equation is used because of missing ambient light.

The physics of light propagation including reflection, refraction and total internal reflection are stated in the law of reflection, Snell's law and the Fresnel equations.

Reflection is the process by which electromagnetic flux incident on a surface leaves the surface without a change in frequency.

Refraction is the change in the direction of a wave due to a change in its velocity. It happens when a wave travels from a medium to another.

Total internal reflection can happen when light arrives to the junction of a higher refractive and a lower refractive medium. It will happen when the angle of incidence is greater than the critical angle. Light will then reflect back internally.

Reflectance is the fraction of the incident flux that is reflected.

4 Materials in the real world

Real world materials are diverse. There are dielectrics, metals, composite materials and other materials. Dielectrics are non-conductive materials like glass or wood which don't have free electrons. As a result they don't reflect well and are often translucent.

Metals are opaque and very reflective and their *color reflectance curve* is complex which means they reflect light of different wavelengths very differently. Composite materials such as paints have properties that are a combination of the reflective properties of metals and dielectrics.

Material surfaces can be profiled with attributes such as reflectance, emissivity, transmittance, fluorescence, phosphorescence and anisotropicity. Surface reflectance can be subdivided into *diffuse* and *specular* components or *Lambertian reflectance* and *Phong reflectance*. The diffusiveness of the surface corresponds to its mattness and specularity corresponds to its shininess. *Transmittance* is the fraction of incident light at a specified wavelength that passes through a surface. *Fluorescence* is an optical phenomenon in which a molecule absorbs a high-energy photon and re-emits it as a lower-energy (longer-wavelength) photon. *Phosphorescence* causes some materials to glow a while after the incident illumination stops. Last, an *anisotropic* surface will look different viewed from different viewing angles around the normal of the surface (*surface normal*). Examples of such materials are brushed metals and velvet.

Some materials and surfaces represent exotic off-specular and retro-reflectional properties. With *off-specular* reflection the peak of the reflection is not centered at the reflection direction. In *retro-reflection* the peak is toward the incident illumination. Examples include the Moon and road markings.

Transmission, fluorescence, phosphorescence and diffraction often have to be left out in real-time graphics.

4.1 Light-Material Interaction

Perfect reflection and transmission constitute perfectly smooth surfaces. Part of the light is reflected and part of it is refracted.

Non perfect reflection means the surface is rough and light is reflected from it in various ways. Some of it is reflected straight off, some can bounce on the surface and some will penetrate the surface and bounce back.

All these contribute to the look of the surface.

5 BRDF

The *Bidirectional Reflectance Distribution Function* defines how much light is reflected for a given light and view direction at a point. It defines what the surface looks like. Non-negativity, conservation of energy and reciprocity are required of BRDFs that are considered physically plausible.

Reciprocity means that if the incoming and outgoing directions are swapped, the value of the BRDF doesn't change.

By *conservation of energy* in this context it is meant that the quantity of reflected light must be less than or equal to the quantity of incident light.

BRDFs can be obtained by measuring BRDF values directly. Also analytic reflectance models may be used that are either physically-based or empirical models.

The exit radiance for a point is calculated by summing emitted radiance of that point and all incident light modulated with the BRDF. Incident light may also be reflected from another point. In real-time shading several simplifications have to be made.

No indirect lighting may be allowed. Incoming radiance may be limited to light sources adding a visibility term. An ambient term can be added to compensate for the missing indirect part. Almost all real-time rendering algorithms leave interreflected light out. Mirror objects or reflective planar surfaces are handled separately.

Only *distant illumination* may be used. The visibility term is usually dropped as well. Incident lighting is calculated only once and reused for all the points on the object. This is called environment mapping and discussed later in more detail.

Light sources may be restricted to point lights only. With them lighting can be simply summed over all point lights without need for full integrals. Visibility may be left out.

5.1 Isotropic BRDFs

Isotropic BRDFs model materials whose surface reflectance doesn't change when rotating the surface around its surface normal. Smooth plastics are an example of such materials.

5.2 Anisotropic BRDFs

Anisotropic BRDFs model surfaces with strongly oriented microgeometry elements. Such materials include brushed metals, hair, cloth and velvet.

5.3 Ideal diffuse reflectance

The *Lambertian diffuse lighting model* published by Lambert in 1760 describes a non shiny surface where any incident light is reflected evenly in all directions taking into account the general surface reflectivity. As incident light energy is spread over an area the intensity of the incident light at that distance modulated by the reflectivity of the surface (*albedo*) is multiplied by the cosine of the normal $\overrightarrow{N}$ of the surface and the vector pointing toward the illuminating light $\overrightarrow{L}$. This is known as *Lambert's cosine law*.

The assumption of the diffuse model that light is reflected evenly in all directions is a rough approximation of real surfaces. An ideal diffuse surface is at a microscopic level a very rough surface. It is only approximately correct for materials such as chalk, clay and some paints.

5.4 Ideal Specular Reflectance

Ideal specular reflectance describes materials whose microscopic surface elements are oriented in the same direction as the surface itself. Mirrors and highly polished metals are examples of such materials. Pixel color is view dependent.

5.5 Non-ideal Reflectors

Snell's law of reflection only applies to ideal mirror surfaces. Real materials are often far from ideal mirror reflectors. But they are not ideal diffuse surfaces either. Most of the reflected light is expected to travel in the direction of the ideal ray. Microscopic surface variations may cause some of the light to be reflected slightly offset from the ideal reflected ray. From farther away from the direction of the ideal ray less and less light can be expected to be reflected. This can be described as *directional diffuse reflection.*

5.6 Glossy Reflection

In practice the BRDF is approximated as the sum of three different components.

- Ideal specular reflection, which corresponds to mirror reflection.
- Lambertian diffuse reflection.
- Glossy reflection, which corresponds to directional diffuse reflection.

5.7 BRDF models

There are many different *BRDF models*. Some of these are data-driven, some are analytical. Analytical models are divided into physical and empirical models. Examples of physical models include the He-Torrence-Sillion-Greeneberg model, the Oren-Nayar model and the Cook-Torrance model. Examples of empirical models include the Lafortune model, the Ashikmin model, the Ward model and the Phong model.

5.7.1 The Phong Model

The Phong Illumination Model tries to mimic real objects by combining diffuse, specular and ambient elements to calculate shading for each point. The Phong model makes several simplifications. Only point light sources are supported. Only surface geometry is considered. Diffuse and specular reflections are modeled only locally so there are no second-order reflections. Ambient lighting is added to compensate this. Specular colour is the same as light colour. The Phong model can be considered as a special case of BRDF based lighting although it doesn't meet the criteria of a real BRDF.

5.7.2 The Cook-Torrance Model

The Cook-Torrance model is based on the microfacet model where a surface is thought to be made of planar microscopic mirroring facets. The Lambertian diffuse model is used for diffuse reflections. For specular reflections the microfacet orientation and the masking and shadowing effects of the microfacets on each other are considered. The model also incorporates the *Fresnel term.* It models the increasing specularity observed near grazing angles.

5.7.3 Comparison

The Cook-Torrence model and the Ashikhmin model perform quite well in comparison to other BRDFs like Blinn-Phong, Lafortune and Ward. Lafortune performs consistently worse near the grazing angle. Blinn-Phong and Ward have no Fresnel effect and as a result they perform worst.

5.7.4 BRDFs with shaders

Vertex shaders can be used for slowly changing BRDFs but there can be Gouraud artifacts and only shift-invariant BRDFs can be used. In a *shift-invariant BRDF* the parameters of the reflectance model cannot change. With fragment shaders Gouraud artifacts can be avoided and *shift-variant BRDFs* are possible so the parameters of the reflectance model can change per pixel. Also the normals and tangents can be changed enabling bump-mapping which is covered later. (Kautz & Seidel, August 2000)

Older hardware doesn't support certain functions used for per-pixel operations. If BRDF parameters are stored to textures dependent texturing can be used for the unsupported operations and functions. In *dependent texture lookup* the color of the first texture serves as the coordinates of a second texture. Bilinear texture filtering can also cause artifacts.

5.7.5 BRDFs using analytical decomposition

The BRDF may be decomposed analytically sampling factors and storing them as a 2D texture. An example is the Torrance-Sparrow Model or the Banks Model.

5.7.6 BRDFs using numerical decomposition

With numerical decomposition fast point evaluation is possible. As a result of having 2D functions instead of a 4D function a good compression rate for the data is attained. The difficulty is determining how to decompose the function. *Singular Value Decomposition* or *SVD* (Fournier, 1995) can be used which requires lots of memory during decomposition and the result contains negative numbers. *Normalized Decomposition* can be used with single-term approximation which doesn't require much memory and the results contain only positive numbers. Quality will depend heavily on input data.

To optimize input data the *Gram-Schmidt Half angle Parameterization* can be used which has good separability and is relatively easy to compute. The results can be stored to *Hemispherical Maps* where the hemisphere is projected onto a texture map. Rendering is simple after this and involves the use of just two texture maps.

BRDF maps can be extended with texture maps by means of adding, modulating or using a mixture of methods. Even different BRDFs can be mixed according to the texture map.

Multi-term Approximations may sometimes be appropriate. But they are more expensive as more textures are needed.

6 BTF

Bidirectional-Texture-Function Dana *et al.* , 1999 or *BTF* is used in rendering materials with complex meso-structures such as certain cloths. It addresses texture shadowing and masking as well as inter-reflections and sub-surface scattering while traditional texture mapping gives only a flat appearance. The BTF captures all light and view-dependent effects of a material by storing the attenuation of light at every position, for every light and view direction. The problem is the huge amount of data that it requires. To address this several compression approaches have been presented. *Principal Component Analysis* or *PCA* has proven to be an excellent solution compressing the data to $1 : 115$ while retaining a very good quality.

7 PRT

In *Precomputed Radiance Transfer* or *PRT* the visibility and shading are precomputed. Spherical harmonics are used for compression. Low-frequency lighting is presumed. Higher-frequency lighting needs higher order spherical harmonics. What is attained is that shadow computation is independent of the number of light sources. Soft shadows are cheaper than hard shadows. Transfer vectors can be calculated offline. Lighting coefficients can be computed at run-time. On the downside, models are assumed to be static and external objects can't cast shadows only partly over an object. Glossy objects are supported. Local lighting is not supported. By using a visibility rasteriser for a low-resolution model it is possible to do PRT in real-time for small dynamic models.

8 3D computer graphics

> 3D computer graphics are distinct from 2D computer graphics in that a three-dimensional virtual representation of objects is stored in the computer for the purposes of performing calculations and rendering images. In general, the art of 3D graphics is akin to sculpting or photography, while the art of 2D graphics is analogous to painting. [WIKIPEDIA]

To render (draw) objects to screen they have to be converted to something the computer knows how to handle. Modern computers use polygons to draw 3D graphics. *A polygon* is a closed two-dimensional path composed of some number of sequential straight line segments. The process of transforming the representations of the objects into polygons is called *tessellation*. This is used for *polygon-based rendering* in which objects are broken down into *meshes* which are nets of interconnected triangles.

Rendering is the final process in which the final image is drawn. On a computer the image is composed of *pixels*, tiny dots of color.

The *resolution* of the image means its level of detail. A higher-resolution means more pixels.

A *vertex* is a point in 3D space with a particular location.

9 Shading models

9.1 Flat shading

Flat shading is a lighting technique in which polygons are shaded based on the polygon's surface normal and the direction of the light source. It is a very simple and fast shading model but it gives low-polygon models a faceted look.

9.2 Gouraud shading

Gouraud shading was invented by Henri Gouraud in 1971. It is a basic shading method for adding a curved feel to a polygon that would otherwise appear flat. Lighting is only calculated at vertices that define the triangles of the polygons. Color is interpolated linearly across a polygon. *Linear interpolation* is estimating the intermediate values of two points with the weighted average formula.

The Lambertian diffuse model is usually used with Gouraud shading. A problem with Gouraud shading is that if the diffuse model is extended with specular reflections and a specular highlight occurs near the center of a large triangle, it will not show at all. This is because of the fact that lighting is only calculated at vertices.

9.3 Texture mapping

Texture mapping is a method for adding detail to surfaces with the use of images that are mapped onto polygons. This method is used as a simple and economic way as opposed to adding more geometric detail to the models. Individual details in the texture cannot be shaded differently. Multiple textures can be combined to simulate more complex materials.

> A simple texture mapping can be coded simply taking the position of each pixel in the interested object and re-mapping it on the texture image using linear interpolation. [WIKIPEDIA]

Nowadays more sophisticated techniques are used in real-time rendering such as bilinear filtering, trilinear filtering, mipmapping and anisotropic filtering.

Bilinear filtering reduces aliasing by averaging the colors of four adjacent pixels of the source texture. *Aliasing* causes artifacts such as moire patterns when a high-resolution image is represented at a lower resolution.

Mipmapping is used to increase rendering speed and minimize aliasing effects. Mipmapping uses several lower resolution versions of a texture map which represent different scales. *Trilinear filtering* applies bilinear filtering on two of the closest mipmap layers and takes the average of these two results for the target texture.

The previous methods don't yield as good results when the texture is viewed from an angle. This is because the area which they sample is always circular but when viewed from an angle it would actually be more like an ellipse. *Anisotropic filtering* uses an elliptic sampling area whose exact shape is calculated from the viewing angle.

9.4 Phong shading

Instead of calculating lighting only at vertices, as in Gouraud shading, in *Phong shading* lighting is calculated per pixel. This is done by normalizing the interpolated normals at each point and using them in the lighting model to calculate the pixel color. This fixes the specular highlight problem. It doesn't change the fact that the underlying surface is composed of polygons. This can be apparent near the edges of polygons where color can change unexpectedly, especially with specular highlights.

The lighting model commonly used with Phong shading is the Phong Illumination Model discussed before.

9.5 Environment mapping

Traditional texture mapping doesn't look realistic for specular surfaces because the texture is fixed on the surface. If the camera were to be moved the image should follow with it according to the laws of light reflection. *Environment mapping* addresses this by generating a 2D texture map for directional information. The map contains information of the radiance for all individual directions to a certain point in space. The model assumes that the mapped objects and lights are infinitely far away but works well enough for most situations when you keep in mind its limitations.

There are various ways to map the radiance from different viewing directions to the 2D parameter domain. These are called *parameterizations*. Sampling ratio differs with different parameterizations. Optimally uniform sampling is reached where there would be a constant number of rays per solid angle for all directions. The sampling ratio would then be $1:1$.

9.5.1 Spherical map

A *spherical map* is an image of a mirroring sphere under an *orthographic projection* which means practically that the camera is in infinity or that the mapping rays are parallel. A spherical map is easy to generate and it is well supported by hardware. Its sampling ratio is not good at $1 : \inf$ and it is not *view independent* so you cannot change the view freely.

9.5.2 Cube map

A *cube map* is composed of six images taken from the center of a cube through its faces. It has a fairly good sampling ratio of $1 : 5 : 1$. It has good hardware support. You have to somehow handle the six textures and there may appear seams with mipmapping.

9.5.3 Parabolic map

A *parabolic map* is an image of a reflective paraboloid under an orthographic projection. Two images are required for a full environment map that covers all directions. It has a better sampling ratio than cube maps at $1 : 4$. Parabolic maps don't need hardware support although support is available. They require two texturing steps or a single pass with multi-texturing.

9.6 Glossy environment mapping

To mimic different materials with less than perfect reflection different BRDFs need to be used. While environment maps store the incident light *Glossy environment maps* store reflected light for all possible surface orientations and view directions.

There are several different models for Glossy environment maps with static lighting. Diffuse Environment maps are accurate and efficient, (MILLER, 1985), (Greene, 1986). Phong Environment maps. (MILLER, 1985). Isotropic BRDF Environment maps. (Kautz *et al.* , 2000). Lafortune Environment maps. Banks Environment maps that use the anisotropic Banks model support light shining through from behind the object. (Kautz *et al.* , 2000). Isotropic BRDFs Environment map model by (Cabral *et al.* , 1999) are very accurate but require lots of memory. All isotropic BRDFs can be used. Quality depends on the knowledge of the central reflection direction.

There are also several models for *Dynamic Environment maps*. Diffuse Environmental maps are very fast and accurate. (Ramamoorthi & Hanrahan, 2001). Approximate Isotropic BRDF Environment maps are not as fast for high-frequency BRDFs. (Kautz *et al.* , 2000)

9.7 Planar reflections

Planar reflections like water or mirrors are rendered in two passes. One pass is made for the normal scene and another for the reflected version. The reflected version can be evaluated by either reflecting the objects about the plane containing the mirror or reflecting the viewpoint. *The Stencil buffer* is used to draw the reflection to the mirror only.

9.8 Bump mapping

Bump mapping was introduced by Blinn in 1978. By tweaking the surface normals used for shading an object it simulates small-scale surface variation. This is done per pixel. By using bump mapping instead of actual geometry lower processing requirements with nearly equal graphical quality can be obtained. As the bumps manifest from altering the interpolated normals of the polygons the bumps will not cast shadows and they will not extend beyond the edges of the object. There are many variations of bump mapping.

9.8.1 Emboss technique

Emboss bump mapping is not actual bump mapping in that normals are not used at all. The impression of a bumpy surface is achieved by perturbing a copy of the bump toward the illuminating light and subtracting the original bump from that. It needs two texture lookups or multipass functionality to do this. It doesn't support view-dependent effects but only diffuse lighting. Also if the illuminating light is directly over the surface the bumps will not show at all.

9.8.2 Offset maps

In an *offset map* each *texel* (textured picture element) in the bump map texture stores two offsets that tell how much the interpolated surface normal $\overrightarrow{N}$ needs to be altered to get the tweaked normal $\overrightarrow{N'}$. The tweaked normals need to be normalized after this.

Normalizing means transforming a vector to length one while preserving its direction.

9.8.3 Height fields

In this technique each texel of the bump map texture indicates the height of the surface fragment. Offsets are obtained by estimating the gradient. *Height fields* are easy to create with a 2D graphics program but require on-the-fly approximation of derivatives. The technique is also prone to filtering artifacts.

9.8.4 Normal maps

It is possible to just simply store the tweaked normals instead of the offsets. This way no re-normalization is required. They do have to be stored in local coordinate space and have to be transformed into appropriate coordinate space. Any lighting model can be used. *Normal maps* can be converted from heightmaps. There are plugins for this for e.g. Photoshop and GIMP. High-resolution polygonal model can also be converted to a low polygonal model and a normal map. There are plugins for this approach for 3D Studio Max and Maya. With normal mapping it is also possible to implement distance-indexed detail scaling. With this technique the detail of the normal map of a given texture can be decreased and less computing is sacrificed for distant surfaces that don't need as much detail.

Even with static scenes where lighting could be precomputed it may be a good idea to use bump mapping. Light maps would have to be high-resolution and they cannot capture specular highlights. You would have to use many lightmaps whereas bump maps can always be reused.

Various vectors need to be normalized in the normal map calculations. Normalization can be done using the mathematical $normalize()$ function or mipmapped cube map lookups. With cube maps each face stores a normalized version of a vector. Cube map lookups are faster but result in artifacts especially when used on the half angle vector $\overrightarrow{H}$ which directly affects specular lighting. So a performance boost while retaining good quality can be achieved by using mipmapped cube map lookups to normalize the surface normal $\overrightarrow{N}$, the vector pointing toward the illuminating light $\overrightarrow{L}$ and the vector pointing toward the camera $\overrightarrow{V}$. [NVIDIA2004]

9.8.5 Extension: Reflective bump mapping

Reflective bump mapping is also called *environmental bump mapping*. The results of one texture lookup can be used to compute texture coordinates for another texture lookup. This technique can be used with normal mapping and offset mapping but it requires rotations into cube-map space with the latter.

9.9 Horizon mapping

As bumps are only represented through normals, instead of actual geometry, standard shadowing techniques cannot be used. In *horizon mapping* the horizon is precomputed and stored at every point in the bump map. At run-time it is checked whether light direction is above the horizon. The height of at least eight directions have to be stored to attain acceptable quality. The horizon is reconstructed by interpolating from the samples at run-time. Linear or higher-order basis functions can be used. Horizon mapping requires many texture lookups and a fairly complex shader. Since the horizons are precalculated the bumps cannot be animated. Also the bumps don't cast shadows on other surfaces. There are yet no solutions for filtering horizon maps at the time of writing this paper. (Heidrich, 2001)

9.10 Gloss maps

Gloss maps are often used together with bump mapping. They model surfaces whose shininess varies such as wet marble or ceramic tiles. This is done by combining a diffuse texture, a gloss texture and a normal map with a specular exponent k. The Blinn-Phong model is commonly used for lighting.

10 Light maps

Light maps can be thought of as a special case of PRT. Only fixed diffuse lighting is possible. A light map stores incident lighting from a fixed set of light sources and includes indirect lighting as well. Light maps are often computed using radiosity. By separating lighting from textures memory is saved. Textures can be reused in different lighting situations. Also accurate lighting models are possible because lighting is precomputed. As lighting usually varies slowly light maps can have a lower resolution.

11 Shadows

Shadows add realism to scenes by serving as a depth cue and by supplying contact points. They make it easier to perceive the position of the illuminating light. Hard shadows and soft shadows can give even more information. Soft shadows are caused by the light source being partially occluded. There are many techniques but each have their trade offs.

11.1 Shadow maps

In *Shadow maps* the idea is that a point is lit if it is visible from the light source. To compute the shadows two rendering passes are required. The first pass is rendered from the light's viewpoint and the second from the observer's viewpoint. During the first pass the nearest depth values are stored to the depth buffer. The resulting depth buffer is the shadow map. On the second pass, from the observer's viewpoint, transform is made to the light's coordinate space for each pixel and the depth of the pixel is compared against the shadow map. If the depth value of the pixel is greater than that of the shadow map then the pixel is in shadow.

Limitations of Shadow maps include the *field of view problem*. A point to be shadowed can be outside the field of view of the shadow map. This can be avoided by using cubical shadow maps and only spot lights.

A *bias* needs to be added to the shadow map depth values to prevent the shadows from appearing in erroneous places as a result of rounding errors. Too big a bias will result in a gap between the object and its shadow.

If the shadow map is under-sampled it will result in aliasing effects. Also reprojection can result in aliasing when the camera and the light are pointing toward each other.

Filtering the shadow map is not meaningful but *percentage closer filtering* can be used by filtering the result of the test with the weighed average of the comparison results. This makes finding the right bias harder but results in nice antialiased shadows.

11.2 Shadow volumes

Shadow volumes represent the volume of the space in shadow. The shadow volume can be computed for each polygon separately. If a point is inside a shadow volume cast by a particular light the point does not receive any illumination from that light. This can be found out by shooting a ray from the eye to the visible point and incrementing or decrementing a counter each time a shadow volume polygon is intersected. If the eye is in shadow the situation must be handled differently. One way is to adjust the initial counter value but this is expensive. Another way is to clip the shadow volumes to the view frustum and include these new polygons. The third way is to start at infinity, draw back-facing shadow polygons, increment the counter if z-fail, draw front-facing shadow polygons and decrement the counter if z-fail.

Shadow volumes introduce a lot of new geometry. It is expensive to rasterize long skinny triangles. For really complex scenes the stencil buffer counter may overflow. Rasterization of polygons sharing an edge must not overlap and must not have a gap.

12 Conclusions

In this paper various techniques involved with real-time shading have been discussed from lighting models defined by Bidirectional Reflectance Distribution Functions to environment mapping, bump mapping and shadows. What wasn't covered in this paper among other things was High Dynamic Range Lighting which will undoubtedly become a standard feature in the near future.

Many of the methods presented in this paper could improve on the shading currently in use. Implementing these techniques involves many pitfalls of which only some were discussed. Using a high-level shading language can ease the work a bit.

There are still some phenomena to which the current models don't give solutions, such as diffraction through a glass. Also there aren't many solutions to dynamic models and wall-to-wall global illumination models are still a glimpse from the future.

REFERENCES

Cabral B., Olano M. & Nemec P. 1999. Reflection Space Image Based Rendering. *Pages 165–170 of: Computer Graphics (ACM SIGGRAPH '99 Proceedings).*

Dana Kristin J., van Ginneken Bram, Nayar Shree K. & Koenderink Jan J. 1999. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, **18**(1), 1–34.

Fournier Alain. 1995. Separating Reflection Functions for Linear Radiosity. *Pages 297–305 of:* Hanrahan P. M. & Purgathofer W. (eds), *Rendering Techniques '95*. NY: Springer.

Greene Ned. 1986. Environment Mapping and Other Applications of World Projections. *IEEE CG&A*, **6**(11), 21–29.

Heidrich W. 2001. Interactive Display of Global Illumination Solutions for Non-Diffuse Environments - A Survey. *Computer Graphics Forum*, **20**(4), 225–243.

Kautz J., Vázquez P.-P., Heidrich W. & Seidel H.-P. 2000. A Unified Approach to Prefiltered Environment Maps. *Pages 185–196 of: Proc. of the 11th Eurographics Workshop on Rendering.*

Kautz Jan & Seidel Hans-Peter. August 2000. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. *Pages 51–58 of: Eurographics/SIGGRAPH Workshop on Graphics Hardware 2000.*

MILLER D.P. 1985. *Planning by Search Through Simulations*. Ph.D. thesis, Yale University.

Ramamoorthi R. & Hanrahan P. 2001. An Efficient Representation for Irradiance Environment Maps. *Pages 497–500 of: Computer Graphics (ACM SIGGRAPH '01 Proceedings).*