



Real-Time Effects in Computer Graphics

How to Create Mirror Reflection and Shadows with OpenGL

Christoph Bichlmeier

12 July 2006

chair for computer aided medical procedures

department of computer science | technische universität münchen



Overview

- Techniques Mirror Reflection
 - Stencil mirror
 - Mirror reflection via projective texture
- Techniques Casting Shadows
 - Shadow mapping: projective texture
 - Shadow volumes



Mirroring

Reflect coordinates through an arbitrary plane defined by a point on the plane and its normal vector:

```
void mirrorMatrix(GLfloat m[4][4], // OUT: resulting matrix
                GLfloat p[3],    // IN: point on the plane
                GLfloat v[3])    // IN: plane perpendicular direction
{
    GLfloat dot = p[0]*v[0] + p[1]*v[1] + p[2]*v[2];

    m[0][0] = 1 - 2*v[0]*v[0];
    m[1][0] = - 2*v[0]*v[1];
    m[2][0] = - 2*v[0]*v[2];
    m[3][0] = 2*dot*v[0];

    m[0][1] = - 2*v[1]*v[0];
    m[1][1] = 1 - 2*v[1]*v[1];
    m[2][1] = - 2*v[1]*v[2];
    m[3][1] = 2*dot*v[1];

    m[0][2] = - 2*v[2]*v[0];
    m[1][2] = - 2*v[2]*v[1];
    m[2][2] = 1 - 2*v[2]*v[2];
    m[3][2] = 2*dot*v[2];

    m[0][3] = 0;
    m[1][3] = 0;
    m[2][3] = 0;
    m[3][3] = 1;
}
```

Special case:

The mirror lies in the $z=0$ plane. This special case reflection is a simple scaling matrix that negates the Z coordinate:

glScalef(1.0, 1.0, -1.0);

Improving Shadows and Reflections via the Stencil Buffer

*Mark J. Kilgard **

NVIDIA Corporation

<http://developer.nvidia.com/attach/6641>



Stencil Mirror

Basic idea

Employ the stencil buffer to mask the reflected scene, i.e. the reflected objects are only rendered where the mirror plane is positioned.

Stencil buffer

- Stencil & Depth buffer share memory.
- Handles additional attributes for every pixel (usually 8 bits for 256 states).
- ***glEnable/glDisable(GL_STENCIL_TEST)***
- ***glStencilFunc();***
Stencil test: Test if *ref* UND *mask* UND current value on the stencil buffer for the particular pixel fails or not.
- ***glStencilOp()***
Defines what to do when the stencil test succeeds, e.g. replace the stencil attribute with the *ref* value.



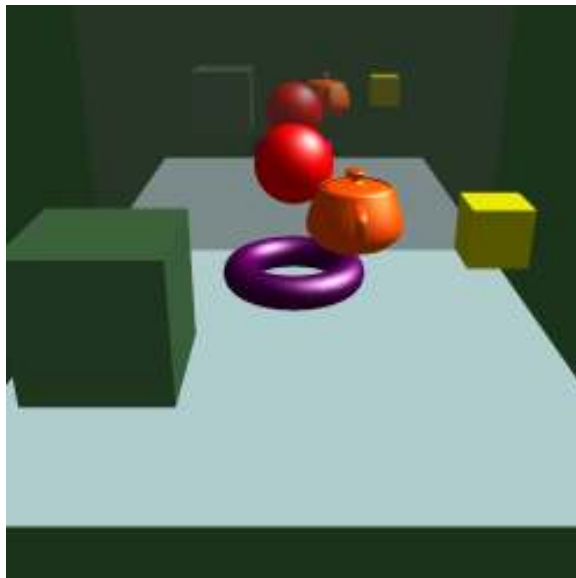
Stencil Mirror

Algorithm (download source code)

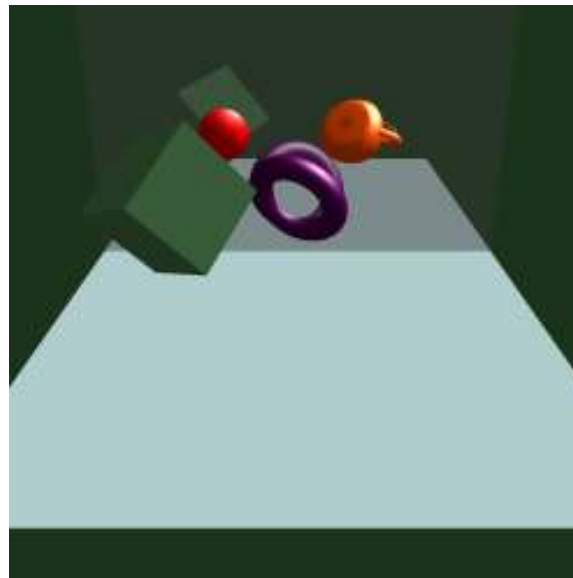
- Initialize modelview and projection matrix
- Enable stencil buffer
- Render mirror plane and initialize stencil buffer.
- Save/Push current modelview matrix
- Multiply **reflection matrix** onto the current modelview matrix to reflect it.
- Enable & set a clipping plane, which is at the same position as the planar mirror plane.
- Draw scene (the reflected one). Rendering is restricted by the stencil buffer test.
- Disable clipping plane
- Disable stencil buffer
- Restore/Pop modelview matrix
- Draw scene (real one)



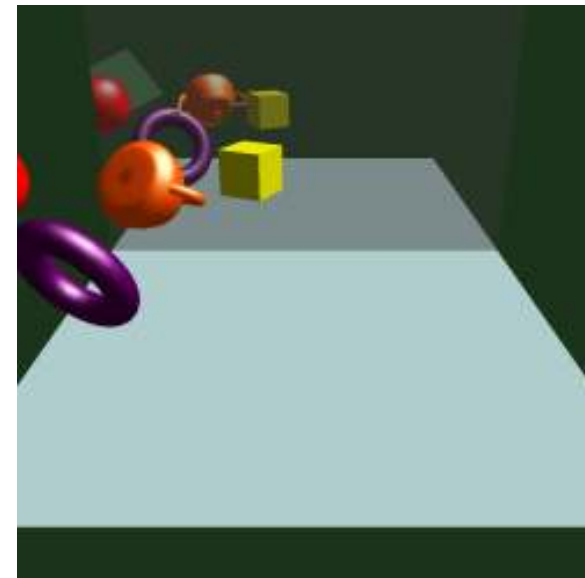
Stencil Mirror



scene



Plane clipping at mirror plane



Stencil clipping outside mirror



Projective Texture Mapping for Mirror Reflection

Basic idea

- Draw reflected scene and store it to a texture with *glCopyTexImage2D()*
- Project the texture onto the mirror plane – Projective texture mapping.

Algorithm

- Initialize modelview and projection matrix
- Multiply reflection matrix onto the current model view matrix
- Enable the clipping plane for the mirror
- Draw scene (the reflected one)
- Store a snapshot to a designated texture
- Disable clipping
- Clear color buffer (reflected Scene is not longer needed)
- Draw scene (real one)
- Draw mirror and project (GL_EYE_LINEAR) the texture onto the mirror plane.

Advantage

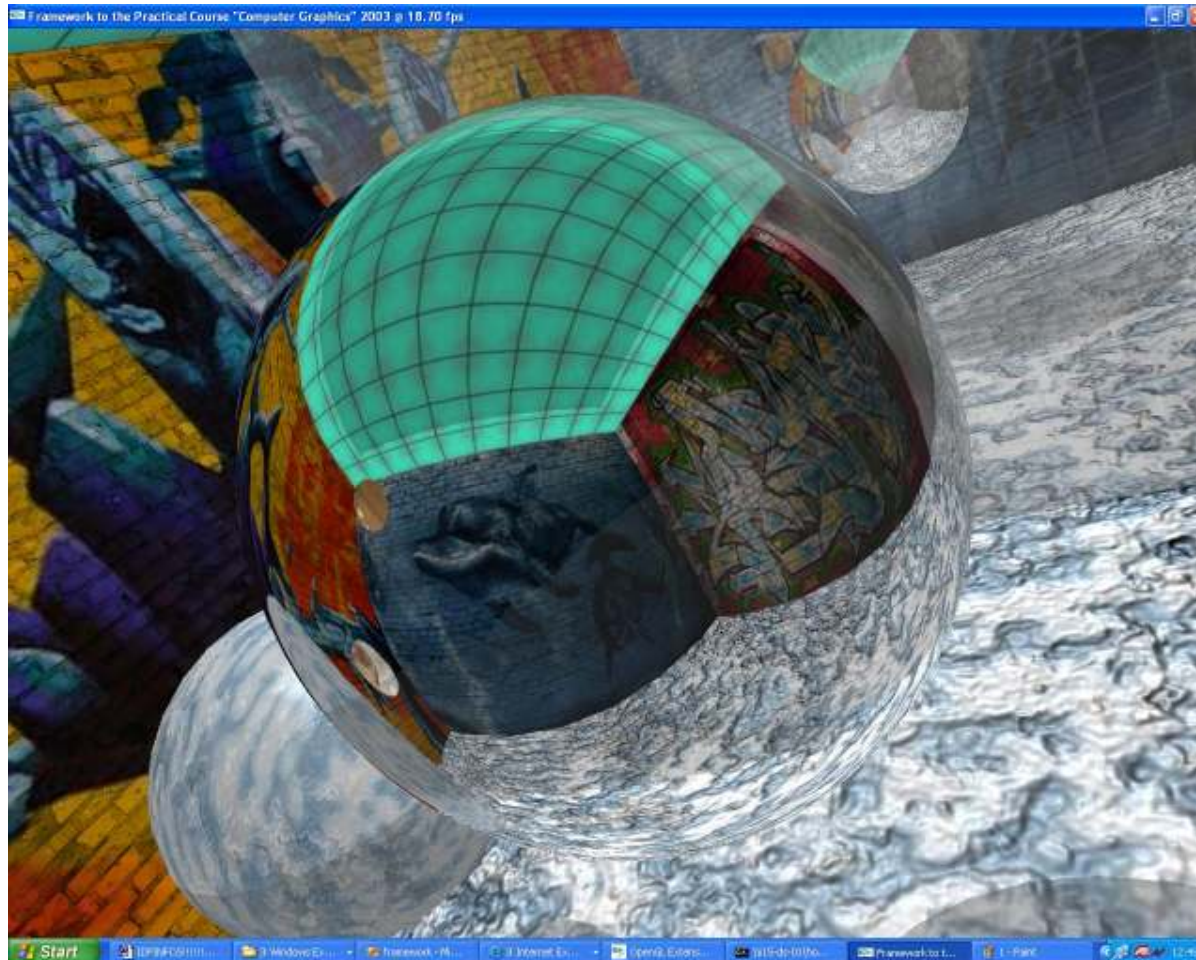
Texture that holds the image of the reflection can be used for further effects, e.g.

Distortion of the Reflection & Multitexturing.



Environment Reflection

Keyword: "Cube Mapping"





Shadows

- Objects that hide a point from the light source are called **occluders**.
- Objects of the scene that are potentially illuminated by light source are called **receivers**.

Types of shadow

- **Attached shadows:** When the normal of the receiver is facing away from the light source, can be performed by definition of a shading model, `glShadeModel(GL_SMOOTH oder GL_FLAT)`;
- **Cast shadows:** When a shadow falls on an object whose normal is facing toward the light source.
- **Self-shadows:** When the shadow of an object is projected onto itself.
- **Hard shadows & Soft shadows**



Taken from Hasenfratz et al. A Survey of Real-time Soft Shadows Algorithms, Computer Graphics Forum



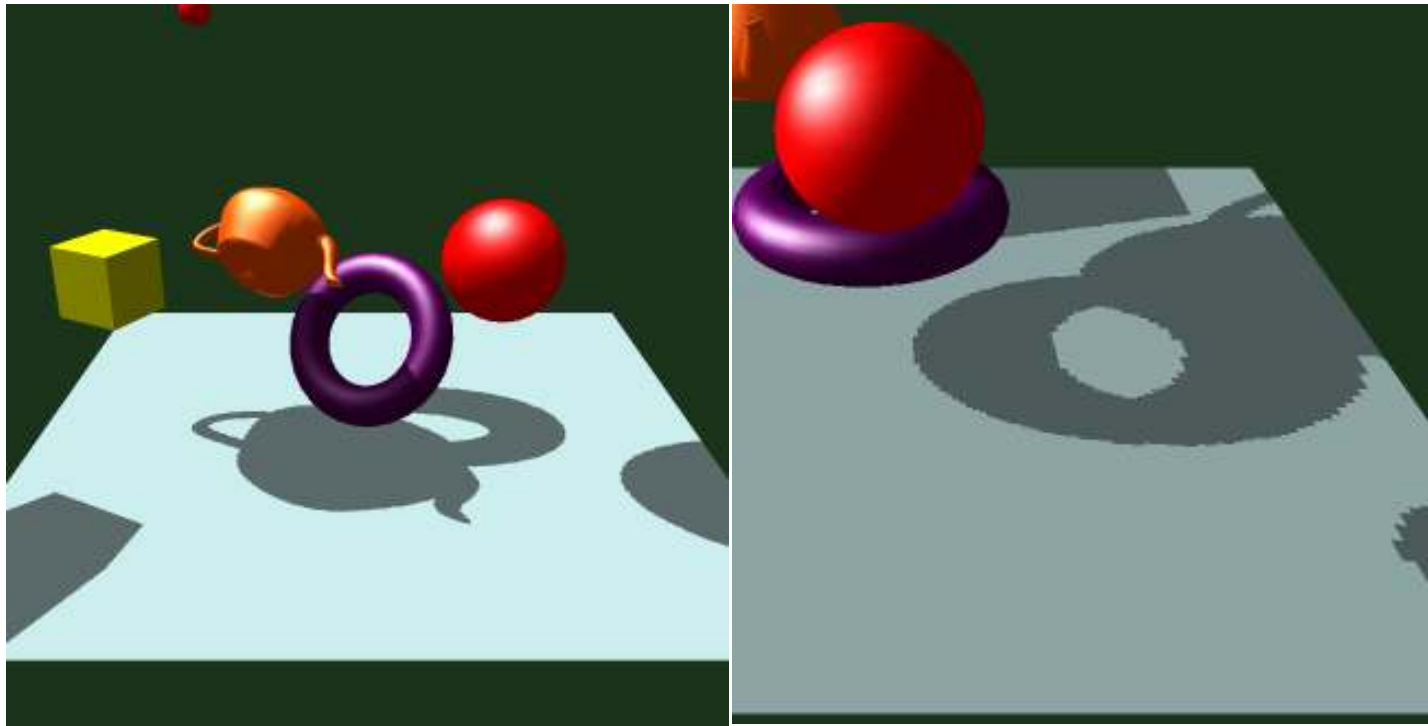
Shadow Mapping

Algorithm

- Set point of view to the light position and render scene.
- Store a snapshot (values of the depth buffer) to a designated depth texture.
- Draw scene from the original point of view with ambient light and project the depth map over the scene (this is not trivial – imagine a very flexible video projector).
- Calculate the distance A of every visible fragment (pixel) to the light source and compare it with the appropriate value B of the texture.
 - If $A > B$: Pixel is in the shadow
 - If $A = B$: Pixel is visible by the light.
- If the pixel is visible, add specula light to illuminate it.
- OpenGL extension for the comparison between Z values, `GL_ARB_SHADOW` (available on Silicon Graphics Hardware above Infinite Reality 2, on NVidia graphics cards after GeForce3 and on ATI graphics cards after Radeon9500.)



Shadow Mapping



Aliasing



Shadow Volumes

Basic Idea

Construct shadow volumes for occluders.

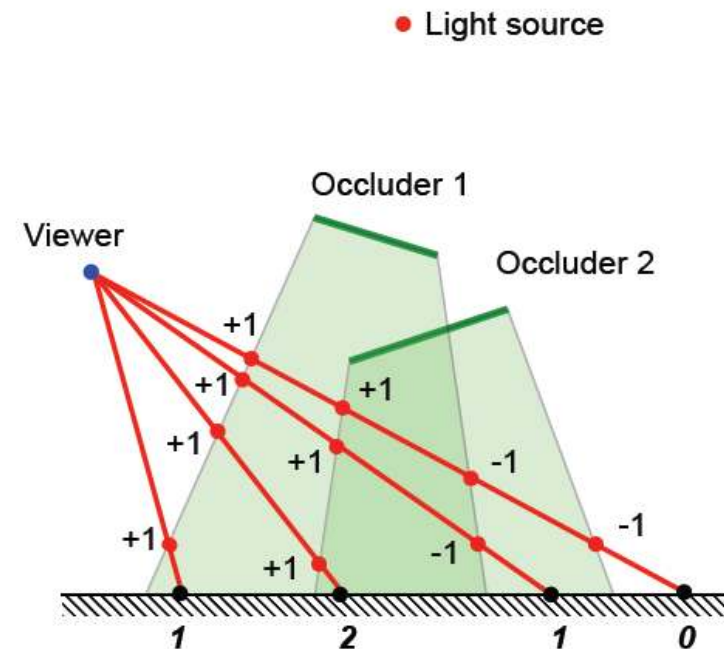
- Find the silhouette of occluders along the light direction.
- Extruding this silhouette along the light direction to form shadow volumes.
- Computing the silhouettes in a dynamic, complex scene is quite expensive. Simplest method is to keep edges that are shared by a triangle facing the light and another in the opposite direction.
- 😊 😊 😊 😊 😊
For first tests you could position spheres (`gluSphere()`) in your scene. Their shadow volumes are Cylinders (`gluCylinder()`). Therefore you can omit the calculation of the silhouettes.
😊 😊 😊 😊 😊



Shadow Volumes

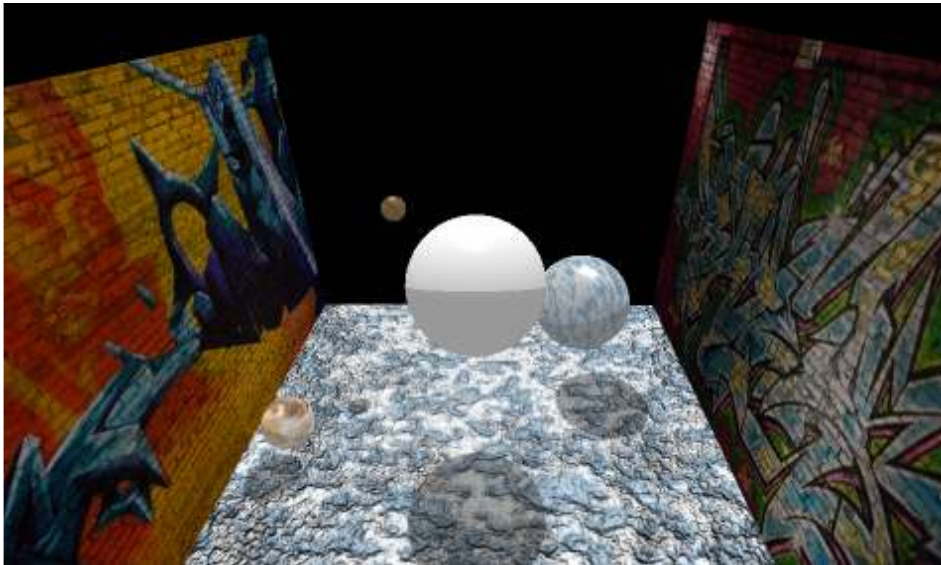
Algorithm

- Render the scene with only ambient lighting.
- Rendered faces of the shadow volumes in the stencil buffer with depth test enabled
 - in a first pass, front faces of the shadow volumes are rendered incrementing the stencil buffer;
 - in a second pass, backfaces are rendered, decrementing it.
- **Pixels in shadow** are between front and back faces of the shadow volume, and have a **positive value (≥ 1)** in the stencil buffer. **Illuminated pixels** have the **value 0**.
- This way to render volumes is called *zpass*.
- render the scene illuminated with stencil test enabled: only pixels which stencil value is 0 are rendered, others are not updated, keeping their ambient color.

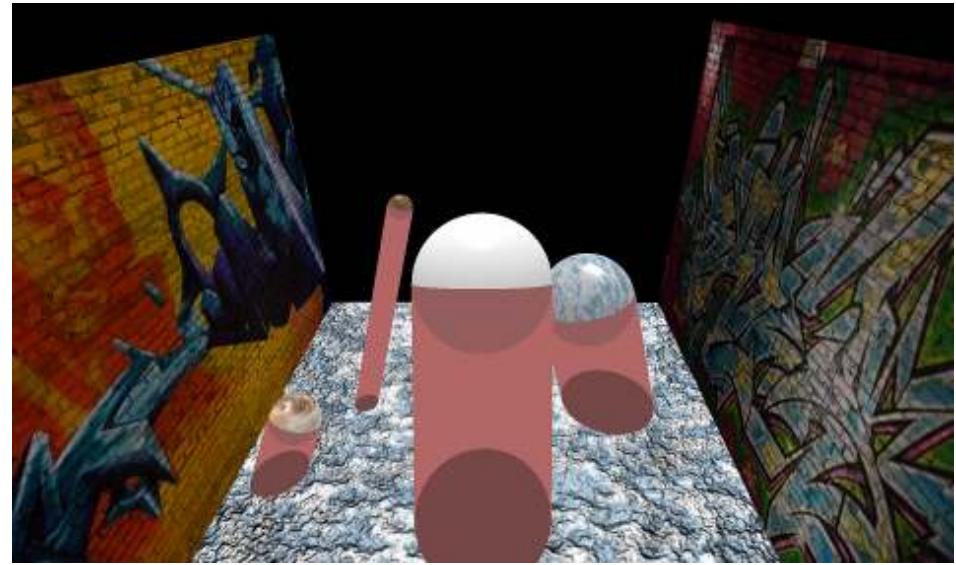


Taken from Hasenfratz et al. A Survey of Real-time Soft Shadows Algorithms
Computer Graphics Forum

Shadow Volumes



Shadow result



Shadow volumes



Compare Shadow Mapping and Shadow Volumes

Shadow Mapping

Advantages

- can be implemented entirely using graphics hardware
- creating the shadow map is relatively fast, although it still depends on the number and complexity of the occluders
- handles self-shadowing.

Drawbacks

- it is subject to many sampling and aliasing problems
- it cannot handle omni-directional light sources
- at least two rendering passes are required (one from the light source and one from the viewpoint)

Shadow Volumes

Advantages

- Works for omni directional light sources
- Renders eye-view pixel precision shadows
- Handles self-shadowing

Drawbacks

- Computation (silhouettes) time depends on the complexity of the occluders
- Method requires at least two rendering passes.
- Rendering the shadow volume consumes fillrate of the graphics card.



Literature

- Hasenfratz J.-M.; Lapierre M.; Holzschuch N.; Sillion F.; IMAG-INRIA A.G.; A Survey of Real-time Soft Shadows Algorithms Computer Graphics Forum, Volume 22, Number 4, December 2003, pp. 753-774(22)
- Mark J. Kilgard* *, Improving Shadows and Reflections via the Stencil Buffer *NVIDIA Corporation*
<http://developer.nvidia.com/attach/6641>
- Franklin C. Crow. Shadow algorithms for computer graphics. *Computer Graphics (SIGGRAPH 1977)*, 11(3):242–248, 1977.

Source code is and example application can be downloaded from the website of the course in a few days.

Thank you!