Shaders and Visual Realism

"Toy Story" - Pixar, 1995 – Using RenderMan

"Bioshock"


"Unreal Engine 3.0"


UNREAL
speedtree


UNREAL
speedtree

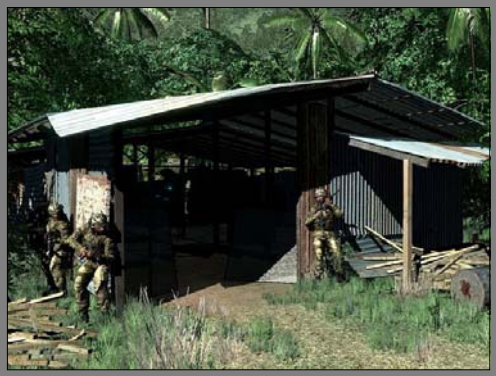
"Alan Wake"

"Alan Wake"

"Alan Wake"

"Alan Wake"

"Unreal Tournament 2007"



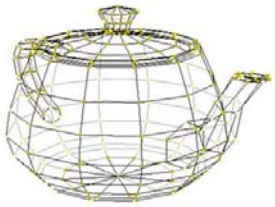
"Crysis"

"Test Drive Unlimited"

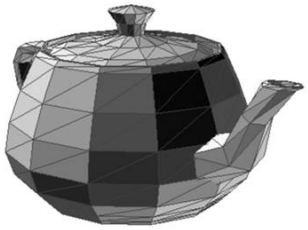
"Test Drive Unlimited"

Most of the following images were taken from
"Shaders for Game Programmers and Artists" by Sebastien St-Laurent,

Vertices

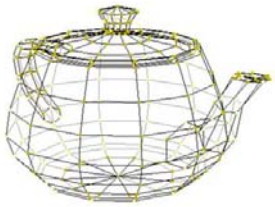


Polygons

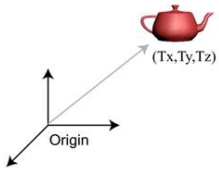


Pixels

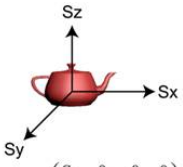
Vertex Transformation



Translating and Scaling

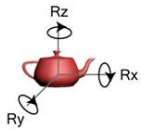


$$\text{Translation} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ Tx & Ty & Tz & 0 \end{pmatrix}$$



$$\text{Scale} = \begin{pmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

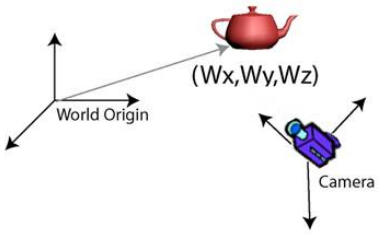
Rotating



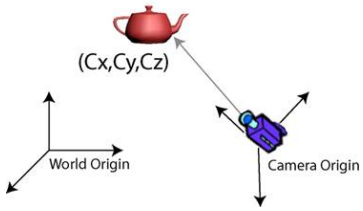
$$\text{Rx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Ry} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Rz} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

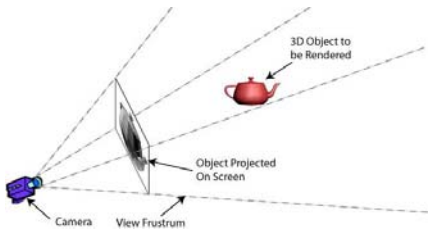
World Coordinates



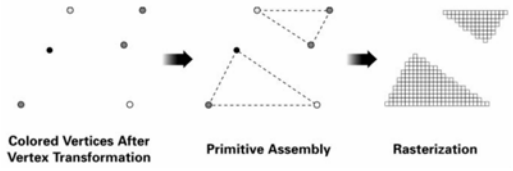
Camera Coordinates



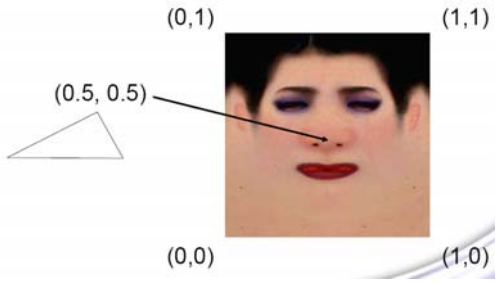
View Projection

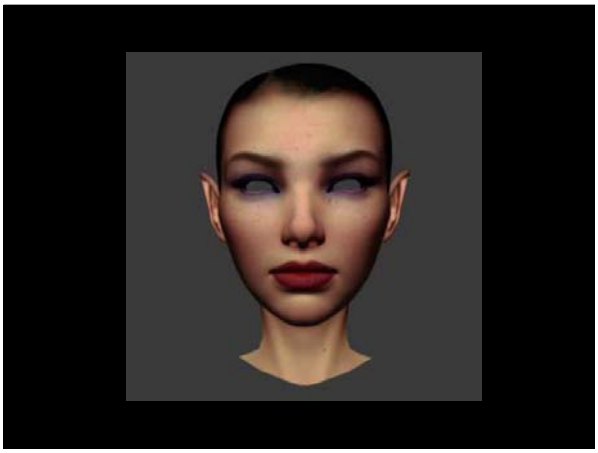


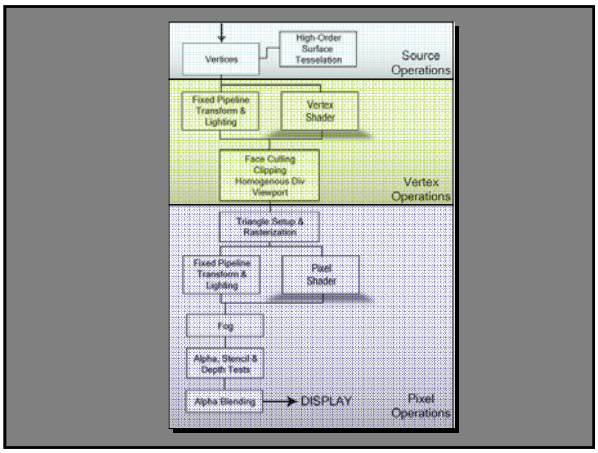
Rasterization

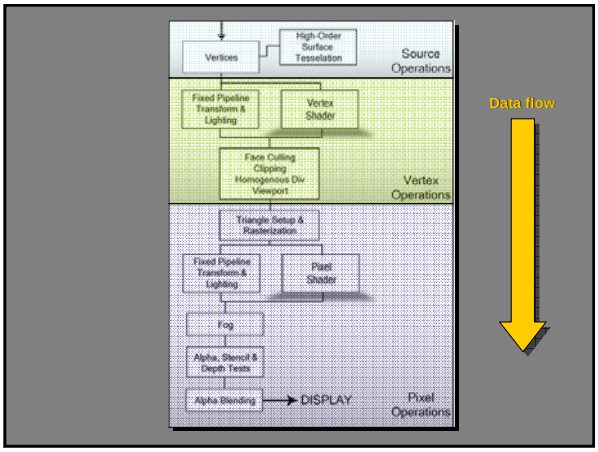


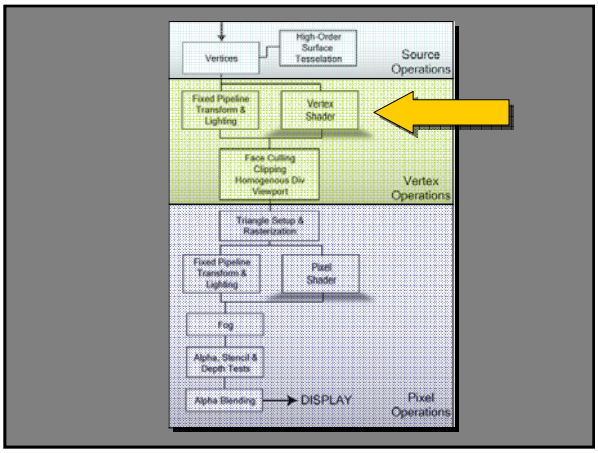
Texture Mapping

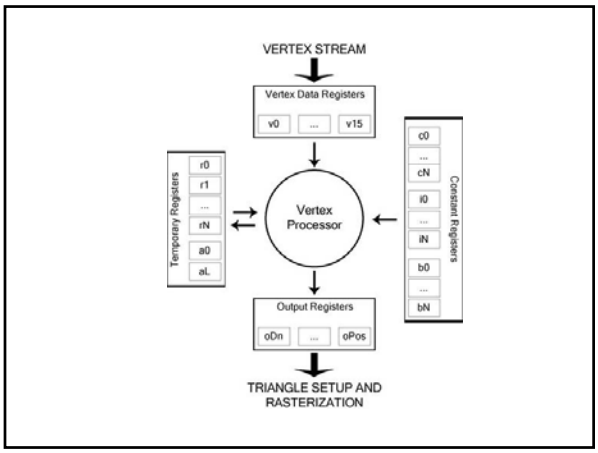





Vertices
High-Order Surface Tesselation
Source Operations
Fixed Pipeline Transform & Lighting
Vertex Shader
Face Culling Clipping Homogenous Div Viewport
Vertex Operations
Triangle Setup & Rasterization
Fixed Pipeline Transform & Lighting
Pixel Shader
Fog
Alpha, Stencil & Depth Tests
Alpha Blending
DISPLAY
Pixel Operations

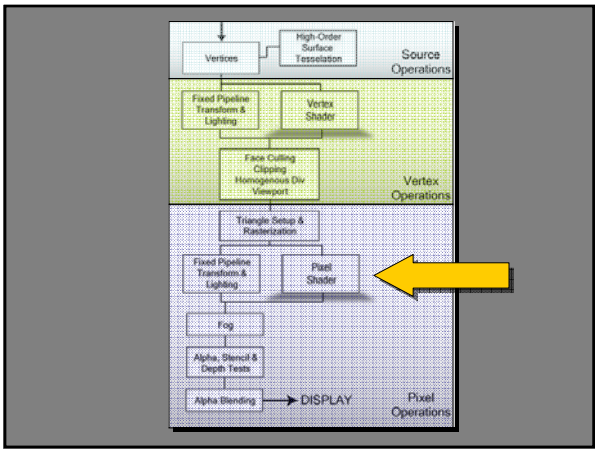

Vertices
High-Order Surface Tesselation
Source Operations
Fixed Pipeline Transform & Lighting
Vertex Shader
Face Culling Clipping Homogenous Div Viewport
Vertex Operations
Triangle Setup & Rasterization
Fixed Pipeline Transform & Lighting
Pixel Shader
Fog
Alpha, Stencil & Depth Tests
Alpha Blending
DISPLAY
Pixel Operations
Data flow

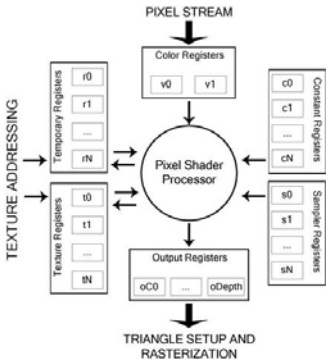

Vertices
High-Order Surface Tesselation
Source Operations
Fixed Pipeline Transform & Lighting
Vertex Shader
Face Culling Clipping Homogenous Div Viewport
Vertex Operations
Triangle Setup & Rasterization
Fixed Pipeline Transform & Lighting
Pixel Shader
Fog
Alpha, Stencil & Depth Tests
Alpha Blending
DISPLAY
Pixel Operations



Vertex Shader

```
// Simplest Vertex Shader
// input vertex
struct VertIn {
        float4 pos   : POSITION;
        float4 color : COLOR0;
};
// output vertex
struct VertOut {
        float4 pos   : POSITION;
        float4 color : COLOR0;
};
// vertex shader main entry
VertOut main(VertIn IN, uniform float4x4 modelViewProj) {
        VertOut OUT;
        OUT.pos = mul(modelViewProj, IN.pos); // calculate output coords
        OUT.color = IN.color; // copy input color to output
        return OUT;
}
```





Pixel Shader

```
// Small Pixel Shader (Grayscale Converter)
// input pixel
struct PixIn {
        float3 color      : COLOR0;
        float3 texcoord : TEXCOORD0;
};
// output pixel
struct PixOut {
        float3 color : COLOR0;
};
// vertex shader main entry
PixOut main(PixIn IN, uniform sampler2D texture : TEXUNIT0) {
        PixOut OUT;
        float3 color = tex2D(texture, IN.texcoord).rgb;
        OUT.color = dot(color,float3(0.299,0.587,0.184)).xxx
        return OUT;
}
```

Sceen Effects

- Pixel shader renders to a temporary texture that it then processes with filters before returning the color values.

Sceen Effects: Glow

Sceen Effects: Depth of Field

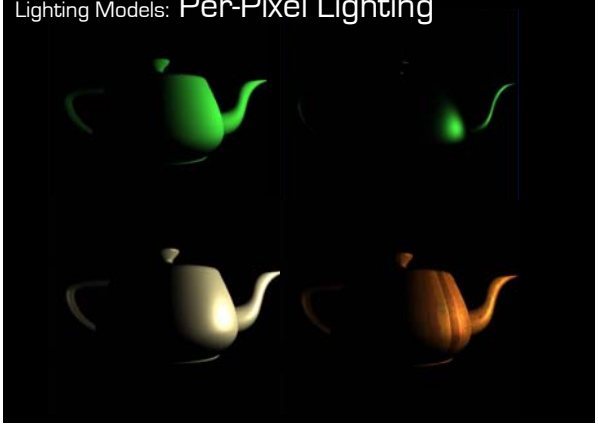
Sceen Effects: Distortion

Sceen Effects: High Dynamic Range

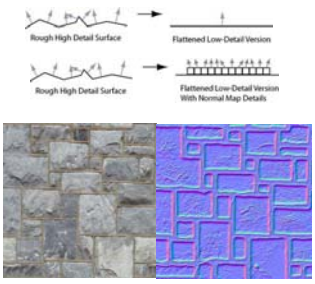
Lighting Models

- Shaders calculate new color values by applying various lighting models, involving parameters such as surface normals, light angle and view angle.

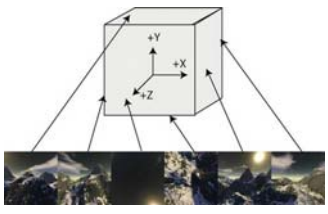
Lighting Models: Per-Pixel Lighting



Lighting Models: Normal Mapping



Lighting Models: Environment Reflection



Lighting Models: Shadows

Non-Photorealistic Rendering

Non-Photorealistic Rendering