3GC3 - Kieran Henderson

Pref quals for high quality triangle mesh

- Close to EQUILATERAL triangles - Vertex close to 6° - Angles close to 60°

Manifolds

- Every edge is shared by exactly 2 triangles - There is a single complete loop of triangles around each vertex

Manifolds with Boundary

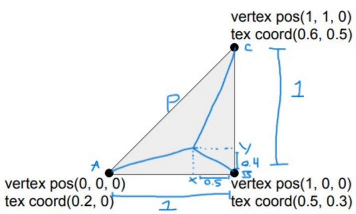
- Every edge is shared by 1 or 2 triangles - Vertex connect to a single edge connected triangle loop (can not connect 2 different groups of triangles) - Every manifold is also a manifold with boundary

Texture Wrapping

GL_CLMAP_TO_EDGE = take outer most pixels and repeat until the edge is reached
GL_REPEAT = repeat the image in a grid with all oriented the same way
GL_MIRRORED_REPEAT = repeat image in a grid with all mirrored along axis between original and new

Barycentric interpolation of texture coords

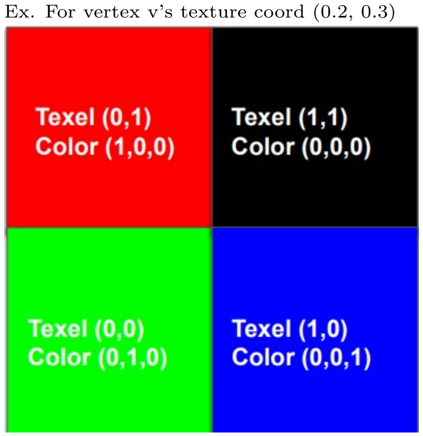
Ex. For texture position (0.5, 0.4, 0)

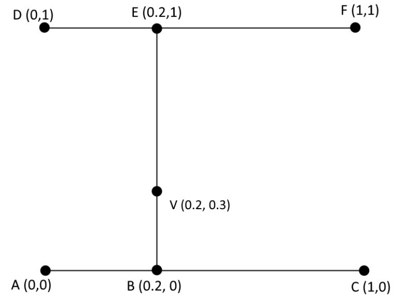


$S_{ABC} = \frac{AB*AC}{2} = \frac{1*1}{2} = 0.5$
$S_{PAB} = \frac{AB*PX}{2} = \frac{1*0.4}{2} = 0.2$
$S_{PBC} = \frac{BC*PY}{2} = \frac{1*0.5}{2} = 0.25$
So $P = \alpha A + \beta B + \gamma C$ Where α, β, γ are the barycentric coords of P
$\alpha = \frac{S_{PBC}}{S_{ABC}} = \frac{0.25}{0.5} = 0.5$
$\gamma = \frac{S_{PAB}}{S_{ABC}} = \frac{0.2}{0.5} = 0.4$
$\beta = 1 - \alpha - \gamma = 1 - 0.5 - 0.4 = 0.1$
So $P = 0.5A + 0.1B + 0.4C$
$= 0.5(0.2, 0) + 0.1(0.5, 0.3) + 0.4(0.6, 0.5)$
$= (0.1, 0) + (0.05, 0.03) + (0.24, 0.2)$
$= (0.39, 0.23)$

Bilinear texture filtering

Ex. For vertex v's texture coord (0.2, 0.3)





Color of E:
$E = 0.8C + 0.2D = 0.8(1, 0, 0) + 0.2(0, 0, 0) = (0.8, 0, 0)$
Color of B:
$B = 0.8A + 0.2C = 0.8(0, 1, 0) + 0.2(0, 0, 1) = (0, 0.8, 0.2)$
Color of V:
$V = 0.7B + 0.3E = 0.7(0, 0.8, 0.2) + 0.3(0.8, 0, 0) = (0.24, 0.56, 0.14)$

Level of mipmap

mipmap level = $log_2 D$ where D = length of longest edge
Just approximate the size of the longest edge and use calculator
$log_2(x) = log(x)/log(2)$

Types of maps

Nothing:

Displacment map:

Bump map:

Properties of light sources

Point Light - Illuminate in all directions and attenuated with distance
Directional Light - The light rays are nearly parallel to each other and there is no attenuation with distance
Spot Light - illuminates in a specific direction and attenuated with distance
Ambient Light - Tries to approximate global illunination and has a constant irradiance at every location and in all directions

Specular Exponent

Smaller specular exponent = larger shiny/white area
Larger specular exponent = smaller shiny/white area

Properties of Curves

Single Polynomial - Interpolation, C^2 continuity, ~~Locality~~
Natual Cubic Spline - Interpolation, C^2 continuity, ~~Locality~~
Hermite Cubic Spline - Interpolation, ~~C^2 continuity~~ C^1, Locality
Vardinal/Catmull-Rom - Interpolation, ~~C^2 continuity~~ C^1, Locality

Hermite vs Vardinal cubic Splines

Hermite - Users have to specify first derivatives of all control points
Cardinal - First derivatives are calculated from the neighbor control points

Terminology

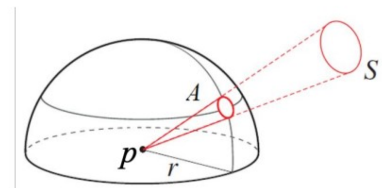
Radiant Flux Φ - Power with units in W or J/s
Irradiance E - Power per unit area with unit in $\frac{W}{m^2}$
Radiance L - Power per unit area per direction with unit in $\frac{W}{m^2*sr}$
Radiant Intensity I - Power per direction with unit in $\frac{W}{sr}$

Acronyms

BRDF - Bidirectional Reflectance Distribution Function
AABB - Axis-Aligned Bounding Box
BVH - Bounding Volume Hierachy
BSP - Binary Space Partition

Hemisphere Projection

Object has area S=100, and it is projected onto hemisphere with radius r=3, the projected area is A=45. What is the solid angle of this object with respect to the point p?



solid angle = $\frac{A}{r^2} = \frac{45}{9} = 5$

Rendering Equation

$L(P, \omega_o) = L_e(P, \omega_o) + \int_\Omega f_r(P, \omega_i, \omega_o) L_i(P, \omega_i) cos\Theta_i dw_i$
L_e = emiting light radiance
f_r = reflection of incoming light
ω_i = incoming direction
ω_o = outcoming direction
L_i = incoming light radiance

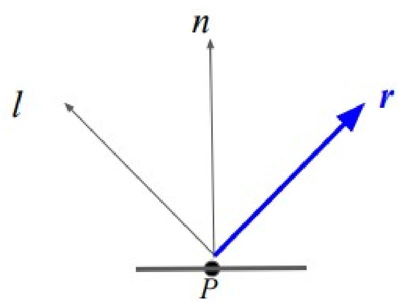
Reflection models

Phong - Uses the angle between the view and reflection vector for computing specular reflection
Blinn-Phong - Uses the angle between the normal and halfway vector for computing specular reflection

Shading models

Gouraud - Uses pervertex color computation
Phong - Uses perfragment color computation

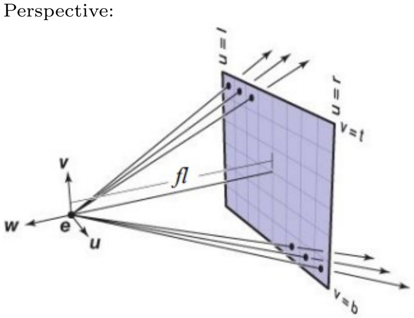
Mirrored reflection calculation



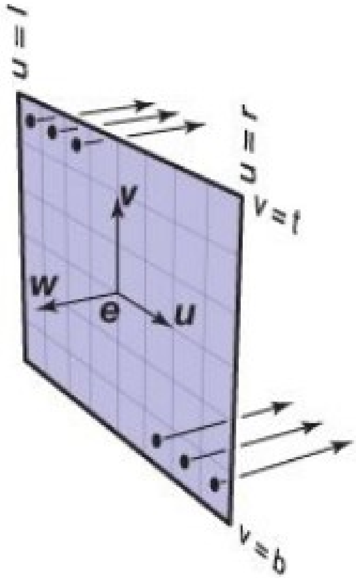
$\vec{r} = -\vec{l} + 2(\vec{l} * \vec{n})\vec{n}$

Ray tracing Modes

Perspective:



Orthographic:

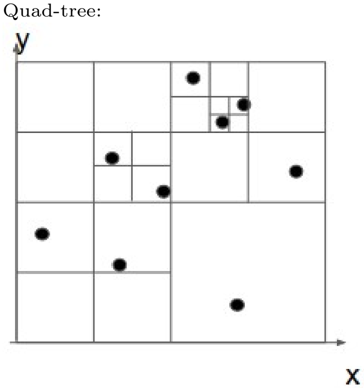


Raytracing Calculation

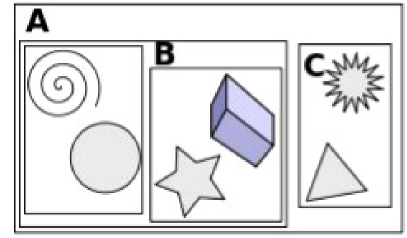
Ray r = o + t d, where o(2,3,4), d(-1, 0, 0), what is the t value when it intersects with plane y=3x?
$r = o + td = (2, 3, 4) + t(-1, 0, 0) = (2 - t, 3, 4)$
$y = 3x \rightarrow 3 = 3(2 - t) \rightarrow y/3 = 2 - t \rightarrow 1 = 2 - t \rightarrow t = 1$

Data structures

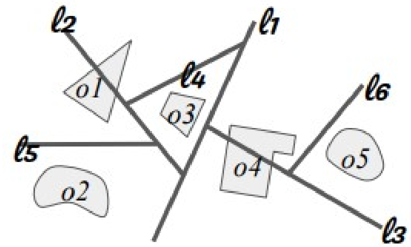
Quad-tree:



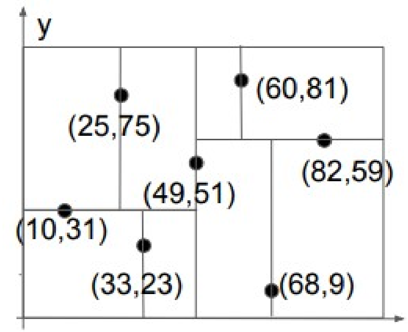
BVH:



BSP:



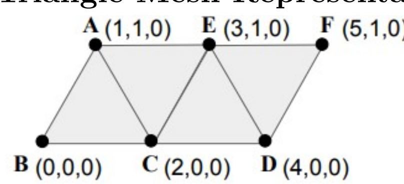
KD tree:



Data structure Applications

Quad - Image Compression
BVH - Ray Tracing
BSP - Painter's Algo
KD tree - Nearest Neighbor Search

Triangle Mesh Representation



vertices = [(1, 1, 0), (0, 0, 0), (2, 0, 0), (4, 0, 0), (3, 1, 0), (5, 1, 0)]
triangles = [(0, 1, 2), (0, 2, 4), (4, 2, 3), (4, 3, 5)]