Lecture 10

required vertex attribute: vertex position

common attribute: color, normal, texture coordinate

Texture mapping: converting a triangle with an image

Texture coordinate -> a pair of floating point numbers

(u, v) or (s, t) that in range [0, 1].

which is also called UV-coordinates

Primitive assembly: breaking down a vertex stream into a sequence

into a sequence of base primitives such as

lines, points, or triangles

face culling: remove triangles from pipeline who is not facing the camera.

recall: rasterization -> the process of converting base primitives to

fragments.

interpolation -> turn vertex attribute into fragment value.

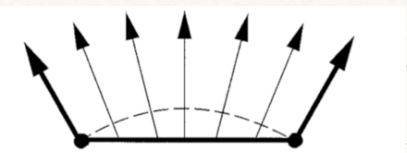
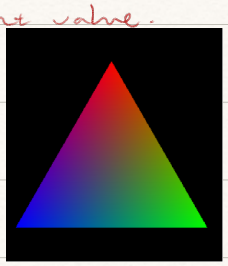


Figure 2: Interpolation of normals between vertices



nearest neighbour: pick the color that is closest to the texture coordination

(Bi) linear filtering: weight based on the distance from the texture

pre-sample operation: — depth

blending

Depth testing : check a pixel is covered before covering it

=> if no, simply cover the pixel

=> if yes, overdraw it, or leave it unmodified.

depth tested is used to determine whether overdraw it or not

=> glDepthFunc() : applied fragment that the depth value is higher

the actual comparison is based on depth buffer.

Blending : blend function determine how that fragment's color is mixed with existing color

$f_{s,R}$, $f_{s,G}$, $f_{s,B}$, $f_{s,A}$: scale factor for the source RGBA

$f_{d,R}$ ... $f_{d,A}$: destination.

new color in frame buffer $new_i := S_i f_{s,i} + D_i f_{d,i}$

by default, source factor = 1, destination factor = 0

=> $f_{s,i} = S_A$, $f_{d,i} = 1 - S_A$