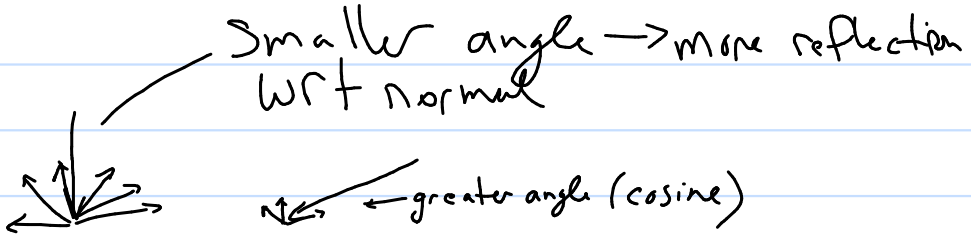
Oct 24/13



Lambert's Cosine Law

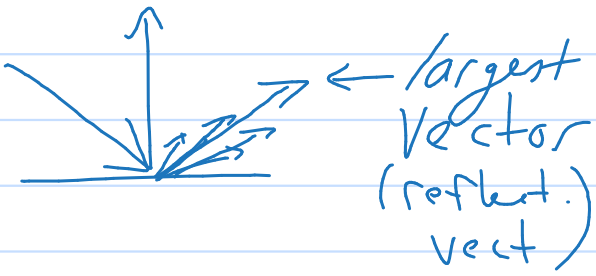
$I = k_d I_d \cos\theta$

$= k_d I_d (n \cdot l)$ ↑ wrt normal ↑

Dot PRODUCT

Diffuse Term

Specular Reflection



$K_s I_s (V \cdot r)^{\alpha}$

Oct 30/13

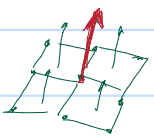
Normal - cross product of 2 vectors of the plane
- sphere -> 1st derivative to get tangent plane

Flat Shading

- applies phong model to single point -- cheap, not realistic.

Vertex Normals

- Vertex normal = avg of 4 adjacent plane normals



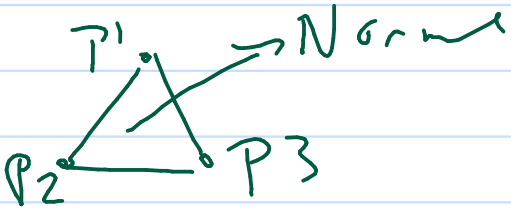
Gouraud - explain on the FINAL
- problem: "mach banding"

Phong Shading

- better.

Smooth Shading
↓
Normal per edge.

glNormal
glVertex
glNormal
⋮

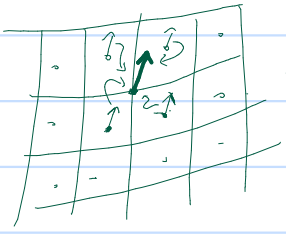


OpenGL *

glBegin(POLY...);
glVertex
glVer
glVer

glNormal(nx,ny,nz);

Flat Shading, only 1 Normal



Calc the normal of the vector.

Texture Mapping

* Bump Mgp -> bonus *

Texels - pixels in textured image

Bump Mapping

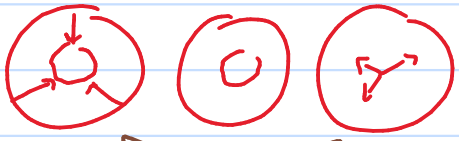
- Change the normals, not the polygons

Texture Mapping -
Environment Mapping -

Nov 6/13 - Texture Mapping

- rasterizer does linear interpolation for s,t mapping
- need parametric of texture coords

- EXAM - sketch the 3 mapping circles

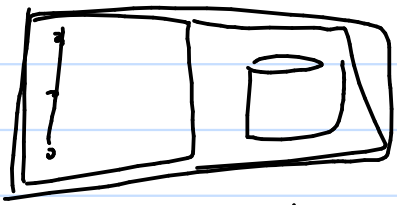


Same for cylinder

OpenGL

glTexImage2D(...)

★ S,T -> same as 1/2π if straight line



- Need to figure out texture coords! -- put into another array; "VAO"

GL_MODULATE

glBindTex...
glTex..

- "replace" mode -- no normals required. -- only for basic flat shapes

- Ground plane -- use multiple!

Nov 13

Cohen-Sutherland ex.

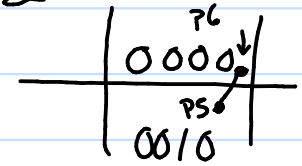
EXAM

Outcodes again...

if 0000 all inside

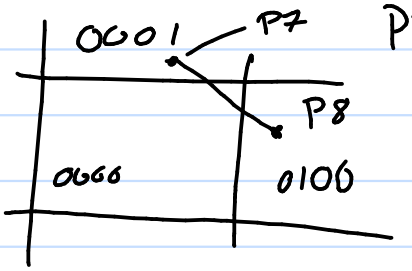
else: AND the two codes; if != 0, then throw the segment away

else:



take p with nonzero code
- look for 1st 1 bit;
- Compute the intersection with that bit's boundary, toss
assign "inside" outcode --> 0000

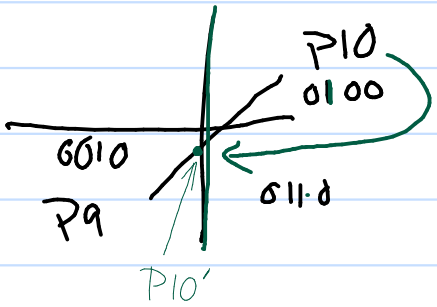
(1)



(2)

P7' = 0000

P8' = 0000



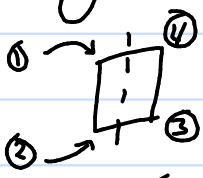
(3)

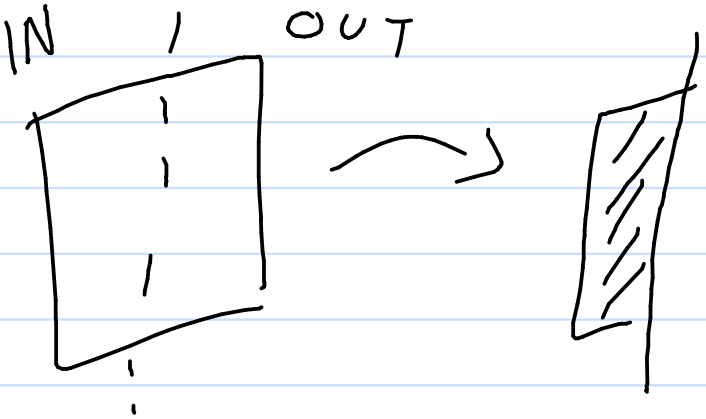
Always Compute intersection with the boundary represented by the Non-zero bit.

☆ Inefficient if most points are outside the clip window

Polygon Clipping

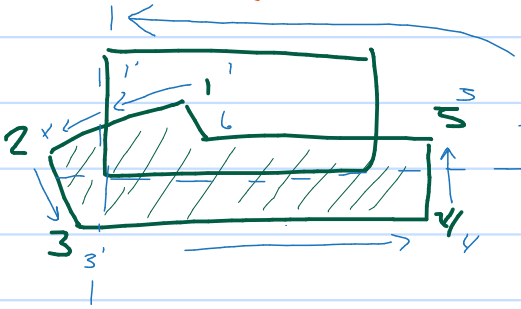
Hodgeman - Sutherland





- both ① & ② are INSIDE, so "output" ②
- ② & ③ ; compute intersection [ ] output intersection

* EXAM



Left 1', 3', 4, 5, 6, 1

Bottom: 1'', 4', 5, 6, 1, 1'

Right: 1', 1'', 1''', 5', 6

Top 1', 1'', 1''', 5', 6

Final

Nov 14

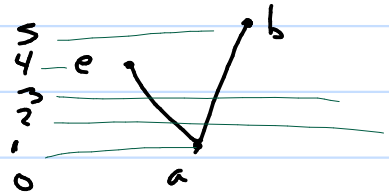
Polygon Scanline Alg.

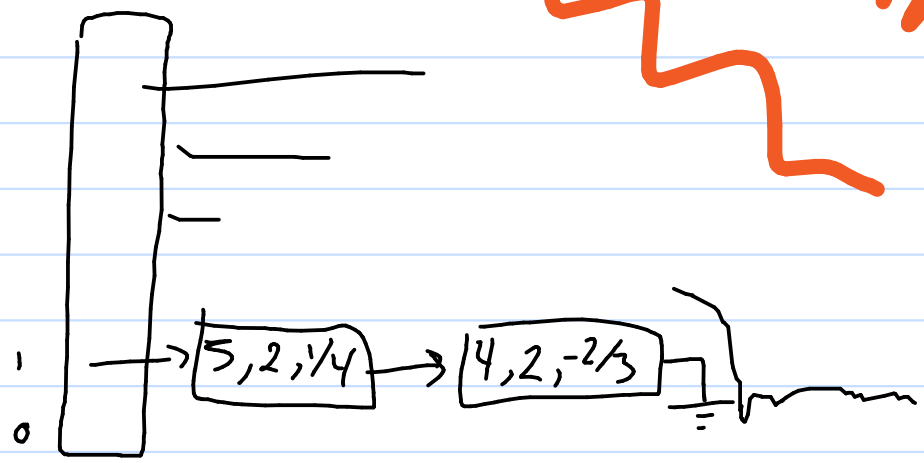
Global Edge Table

GET Entries (ymax, X@ymin, 1/slope)

EXAM

1





2 Active Edge Table

- uses the GET → gets x by adding 1/slope to X@ymin.

Aliasing / Anti..

Software

#2 - turn on neighbouring pixels with lighter, blended colour value.

Increase resolution

#1 **Hardware**

- more pixels!

Nov 20/13

Visible Surface Determination ✓

Shaders

Vertex + Fragment Shaders
- replace the earlier usual pipeline stuff.

PhongLighting → Eye Coords !!

Nov 27

- parallelized by vertex } V.Sh
  " " pixel } F.Sh

Phong Lighting → V. Shader
Phong Shading → Fr. Shader [perpixel; looks better]

"Uniform" keyword → global variable

- Fr. shaders output a color

"Varying" keyword → pass from V.Sh to Fr.Sh

GLULookAt = Rotation × Translation

Transformations - Views and/or Projection
(opposite of midterm)

1

- given GLULookat(params), write the matrices

- maybe another Transform on a house/barn etc like mid.
- know camera analogy.
- object representation

3

- parametric vs implicit ← volumetric
  ↖ typically surface

| don't need to memorize spline eq etc

- list & describe
  X spline
- subdivision surface

2

- Every Year: Lights, Shading => PHONG
  - desc components
  = "Adding up the Components"

=> Gouraud!

=> Phong Shading

Shading is how to apply a lighting model
- local lights
- Global: Raytracing, Radiosity

Texture Mapping

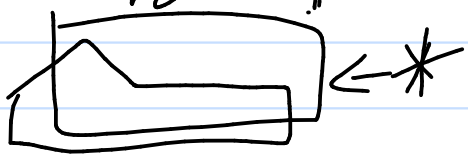
NO { X environment mapping
   { X bump

- how to find texture coordinates of mesh
  - packages, Two-Part Mapping technique

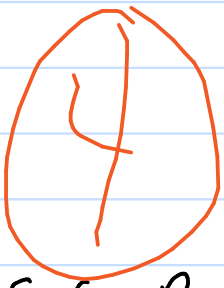
So, manual or automatic mapping.
- mag/minification

Definitely 1 or 2 Qs on:

- clipping - Cohen Sutherland line clipping
  - X Liang Barsky
- Poly: Sutherland-Hodgman → show how a poly is clipped



- Issue: Extended Sutherland-Hodgman Correction



Hid Surface Removal

- Painter's alg
- Z-buffer alg

DDA - elim f.p. mult.

Bresenham - just the basics

Poly Rasterization

- even odd
- non-zero wind } *
- scan line alg. ⇐ show G.E.T. & A.E.T. **

Shaders.

- Variable keywords
  - uniform (Global)
  - varying (interpolate in the rasterizer so the frag sh. has an interpolated value.)

X - built-in variables