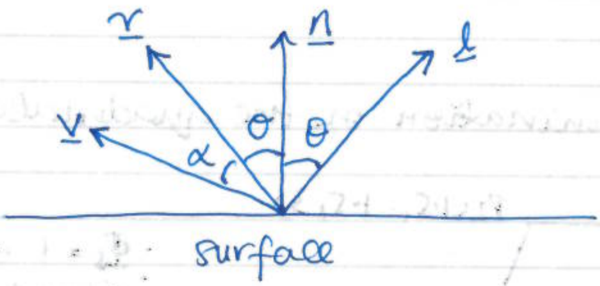
5:55am

2014

1) Phong shading

$$L = K_a + \left[K_d (n \cdot \ell) + K_s (v \cdot r)^q \right] \frac{\Phi}{4\pi d^2}$$

a)



$$\cos\theta = \frac{n \cdot \ell}{|n||\ell|}$$

$$\cos\alpha = \frac{v \cdot r}{|v||r|}$$

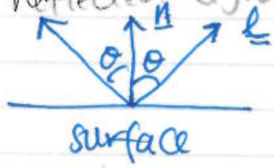
b) K_a: ambient coefficient

Ambient illumination represents the reflection of all indirect illumination

K_d: diffuse coeff.

Diffuse illumination assumes the surface reflects equally in all directions. smaller angle = higher energy of reflected light

$$L(\omega_r) = K_d (n \cdot \ell) \frac{\Phi_s}{4\pi d^2}$$

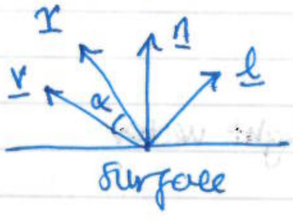


q: specular exponent

K_s: specular coeff.

Specular illumination assume reflection is only at a mirror angle and is view dependent.

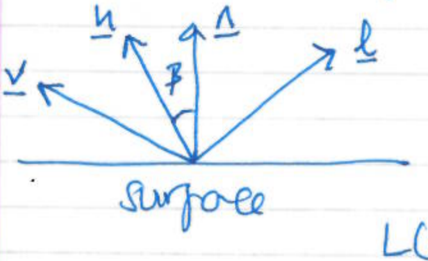
$$L(\omega_r) = K_s (v \cdot r)^q \frac{\Phi}{4\pi d^2}$$



The closer v is to r, gives brighter looking secondary rays.

c) Blinn-Phong shading

We use the halfway vector h between ℓ and v

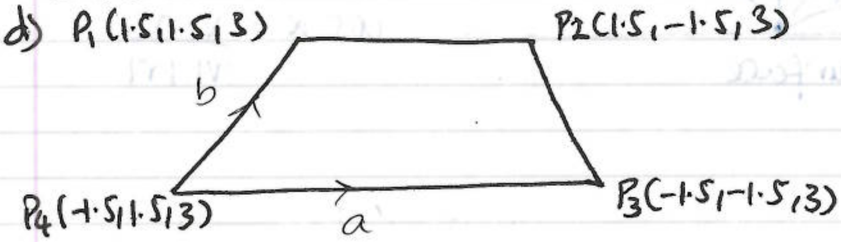


$$h = \frac{\ell + v}{\| \ell + v \|}$$

$$L(\omega_r) = K_s (\cos\beta)^q \frac{\Phi_s}{4\pi d^2} = K_s (n \cdot h)^q \frac{\Phi_s}{4\pi d^2}$$

Advs of Blinn Phong compared to Phong.
Blinn-Phong requires a lower exponent value (q) for the same shading effect as Phong.
+ much quicker to compute when viewer and light are are remote as $\underline{h}$ can be treated as a constant.

Compute the illumination of the quadrilateral at its centre.

d) 

- $\Phi_s = 1$ at $p(4, 0, 0)$
- viewed from origin
- $Ka = 1$
- $Kd = 2$
- $Ks = 3$
- $q = 2$

→ $\underline{n} = a \times b$

$$a = \begin{pmatrix} 0 \\ -3 \\ 0 \end{pmatrix} \quad b = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \quad \underline{n} = \begin{pmatrix} 0 \\ 0 \\ 9 \end{pmatrix} \quad \underline{\hat{n}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$L = Ka + \left[Kd(n \cdot l) + Ks(v \cdot r)^q \right] \frac{\Phi_s}{d^2}$$

→ centre point

$$= \frac{1}{4} \begin{pmatrix} 0 \\ 0 \\ 12 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}$$

Normalisation

→ light vector

$$l_{origin} - \text{Point} = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ -3 \end{pmatrix} \qquad \underline{\hat{l}} = \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix}$$

→ view vector

$$V_{origin} - \text{Point} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} -? \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -3 \end{pmatrix} \qquad \underline{\hat{v}} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

→ reflection vector

$$\underline{r} = 2(n \cdot l)n - l$$

$$= 2\left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix} \right] \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1.2 \end{pmatrix} - \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix}$$

distance $= |\underline{l}| = \sqrt{16+9} = 5$

$$L = 1 + \left\{ 2\left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix} \right] + 3\left[\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 0.8 \\ 0 \\ -0.6 \end{pmatrix} \right]^2 \right\} \frac{1}{5^2}$$

$$L = 1 + \{-1.2 + 1.08\} \frac{1}{5^2}$$

$$L = 0.0752$$

e) vertex shader

· Transforms the input vertex stream into a stream of vertices mapped onto the screen. Attributes of vertices can then transfomed eg color, texture etc

Geometry shader

· This is an optional stage between the vertex and fragment shader

· The geometry shader, for each input primitive has access to all the vertices that make up the primitive, including adjacency info

· can generate primitives dynamically

Fragment shader

· Given the interpolated vertex attributes (vertex shader output) the fragment shader computes color values for each fragment.

f) Gouraud shading: Implemented in the vertex shader because it faces vertices and interpolates color between them and doesn't interpolate normals.

Phong shading: Implemented in the fragment shader because it interpolates normals of the points inside the triangle

Q2. Color

a) ~~CIE diagram~~ x and y coordinates in CIE diagram and plot the point as P.

$$x = r / r+g+b = 150/(150+99+51) = 0.5$$

$$y = g / r+g+b = 99/(150+99+51) = 0.33$$

b) Estimate wavelength of the pure color. (fully saturated color)
The line intersects with the complement color at 620nm

c) Estimate the saturation of the color in a)

$$= \frac{|P - W|}{|W - E|}$$

P: point P
W: white point
E: edge intersection point at 620nm

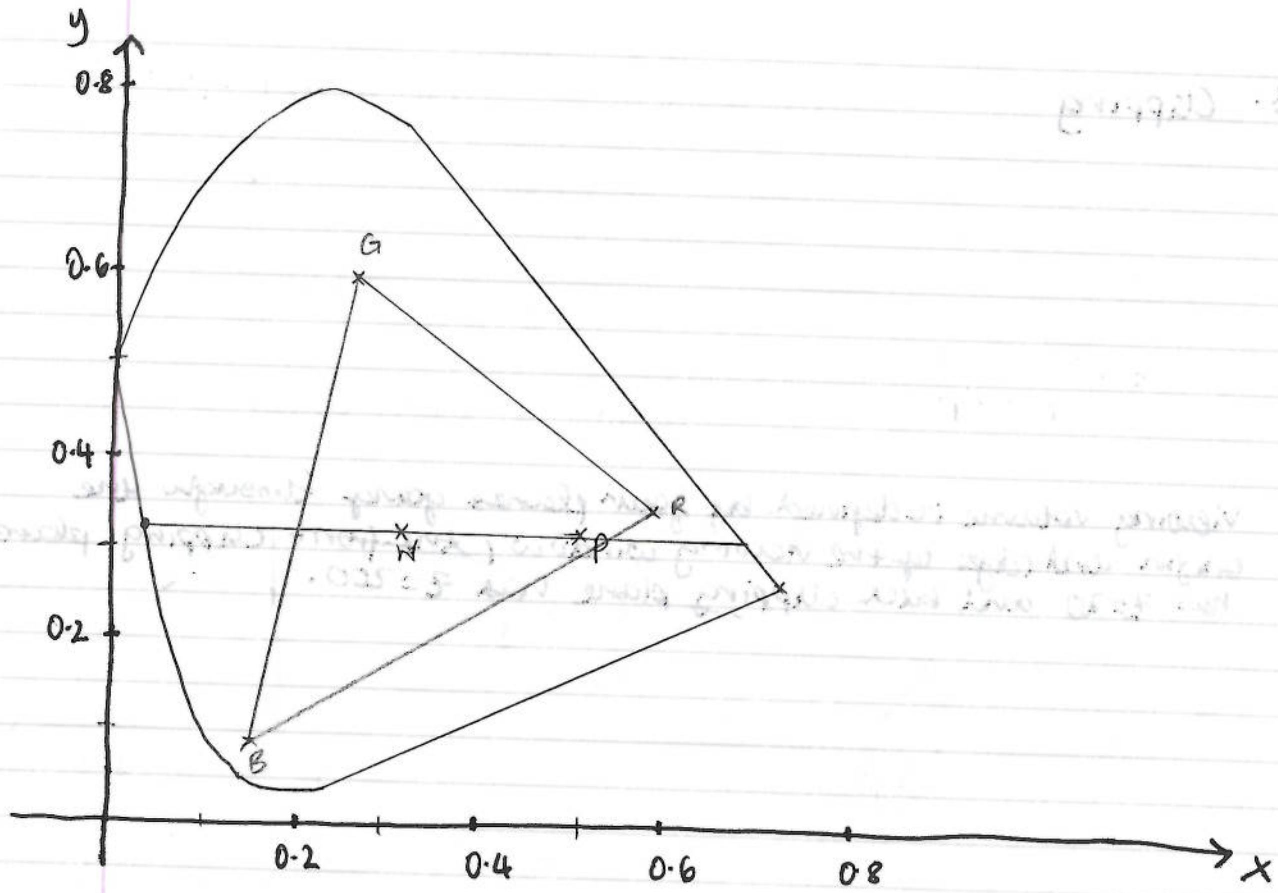
$$= \frac{|0.5 - 0.33|}{|0.33 - 0.66|} = 0.52$$

d) Find wavelength of complement of color in b)

Estimate from line ≈ 490 nm

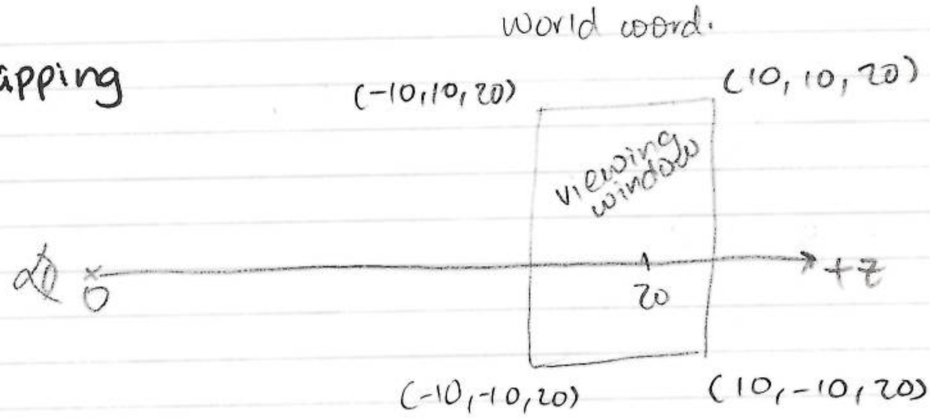
e) See diagram

f) CIE must be convex because visible light is a mixture of pure hues. The pure hues are round the edge and any blend of them must be inside the horseshoe.



y
0.8
0.6
0.4
0.2
G
R
W
P
B
0.2
0.4
0.6
0.8
x

Q3. Clipping



Viewing volume is defined by four planes going through the origin and edges of the viewing window, ~~the front~~ clipping plane has $z=20$ and back clipping plane has $z=200$.

a) Find the inner normals

Near plane: $\underline{n} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ (pointed towards positive z / away from origin)

Far plane: $\underline{n} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$ since the normal had to point towards origin $n = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$ not $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

[x val should be -ve] Left plane: $\begin{pmatrix} l \\ b \\ n \end{pmatrix} \times \begin{pmatrix} l \\ t \\ n \end{pmatrix} = \begin{pmatrix} bn-nt \\ nl-ln \\ lt-bl \end{pmatrix} = \begin{pmatrix} bn-nt \\ 0 \\ lt-bl \end{pmatrix} = (b-t)\begin{pmatrix} n \\ 0 \\ -l \end{pmatrix}$

x val should be -ve so $(-n, 0, l)$

[x val should be +ve] Right plane: $\begin{pmatrix} r \\ b \\ n \end{pmatrix} \times \begin{pmatrix} r \\ t \\ n \end{pmatrix} = \begin{pmatrix} bn-nt \\ rn-nr \\ rt-br \end{pmatrix} = (b-t)\begin{pmatrix} n \\ 0 \\ -r \end{pmatrix}$

Top plane: $\begin{pmatrix} l \\ t \\ n \end{pmatrix} \times \begin{pmatrix} r \\ t \\ n \end{pmatrix} = \begin{pmatrix} tn-nt \\ nr-ln \\ lt-tr \end{pmatrix} = (r-l)\begin{pmatrix} 0 \\ n \\ -t \end{pmatrix}$

→ y val should be -ve so $(0, -n, t)$

Bottom Plane: $\begin{pmatrix} l \\ b \\ n \end{pmatrix} \times \begin{pmatrix} r \\ b \\ n \end{pmatrix} = \begin{pmatrix} bn - nb \\ nr - ln \\ lb - br \end{pmatrix} = (r-n)\begin{pmatrix} 0 \\ n \\ -b \end{pmatrix}$

Plane equations

$$\left.\begin{aligned}
H_{near} &= (0, 0, 1, -near)^T \\
H_{far} &= (0, 0, -1, far)^T \\
H_{left} &= (-near, 0, left, 0)^T \\
H_{right} &= (near, 0, -right, 0)^T \\
H_{top} &= (0, -near, top, 0)^T \\
H_{bottom} &= (0, near, -bottom, 0)^T
\end{aligned}\right] \text{Lecture}$$

Derived

$H_{near} = (0, 0, 1, -20)^T$

$H_{far} = (0, 0, -1, 200)^T$

$H_{left} = (-n, 0, l, 0)^T$

$H_{right} = (n, 0, -r, 0)^T$

$H_{top} = (0, -n, t, 0)^T$

$H_{bottom} = (0, n, -b, 0)^T$

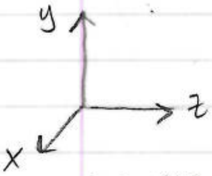
b) Determine which vertices are inside the viewing volume.

	World coordinates
$P_1(100, 100, 210)$	$(-10, -10, 20)$
$P_2(95, 85, 180)$	$(-10, 10, 20)$
	$(10, -10, 20)$
$P_3(70, 70, 170)$	$(10, 10, 20)$
	$\downarrow \quad \downarrow$
$P_4(70, 90, 160)$	l/b

P_1: $z=210$ which is >200 therefore this point sits outside the volume.

→ If $20 < z < 200$ its inside the near and far plane

→ since all x values are (+)ve and +ve x-values are toward the right we need to check the right plane.



→ since all y-values are (+)ve and +ve y-values are toward the top we need to check top plane.

	Top $H_{top}=(0,-n,t,0)^T$	Right $H_{right}=(n,0,-r,0)^T$
* $\Rightarrow$?	Plane equation: $H(0,-10,10,0)$ $-10y+10z=0$	Plane equation $H(10,0,-10,0)$ $10x-10z=0$
P_2:	$d=H\cdot p$ $=\begin{pmatrix}0\\-10\\10\end{pmatrix}\cdot\begin{pmatrix}95\\85\\180\end{pmatrix}$ $=950>0$ ∴ inside	$d=H\cdot p$ $=\begin{pmatrix}10\\0\\-10\end{pmatrix}\cdot\begin{pmatrix}95\\85\\180\end{pmatrix}$ $=-850<0$ ∴ outside
P_3:	$d=H\cdot p$ $=\begin{pmatrix}0\\-10\\10\end{pmatrix}\cdot\begin{pmatrix}70\\70\\170\end{pmatrix}$ $=1000>0$ ∴ inside	$d=H\cdot p$ $=\begin{pmatrix}10\\0\\-10\end{pmatrix}\begin{pmatrix}70\\70\\170\end{pmatrix}$ $=-1000<0$ ∴ outside

P_3:

$d = H \cdot P$
$= \begin{pmatrix} 0 \\ -10 \\ 10 \end{pmatrix} \cdot \begin{pmatrix} 70 \\ 90 \\ 160 \end{pmatrix}$
$= 100 > 0$
$\therefore$ inside

$d = H \cdot P$
$= \begin{pmatrix} 10 \\ 0 \\ -10 \end{pmatrix} \cdot \begin{pmatrix} 70 \\ 90 \\ 160 \end{pmatrix}$
$= -900$
$\therefore$ outside

P_0, P_1, P_2, P_3 are all outside the volume

c) Write mx that projects 3D points of scene onto viewing plane $z=20$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/20 & 0 \end{pmatrix}$$

Homogeneous $\begin{pmatrix} x \\ y \\ z \\ z/20 \end{pmatrix}$

Cartesian $\begin{pmatrix} 20x/z \\ 20y/z \\ 20 \\ 1 \end{pmatrix}$

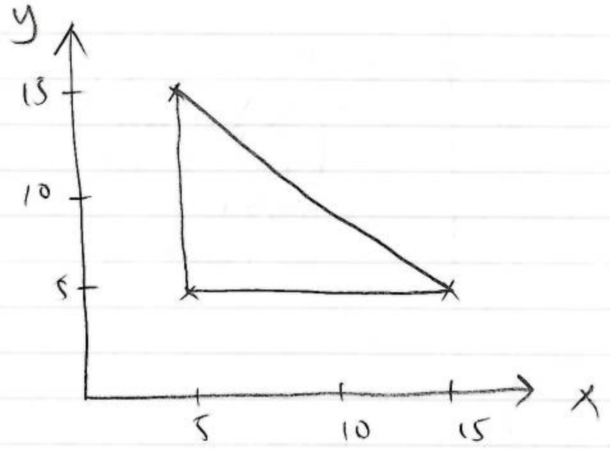
d) Find the 2D coordinates of the projected vertices.

Cartesian (2D)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/20 & 0 \end{pmatrix} \begin{pmatrix} 10 \\ 10 \\ 40 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 40 \\ 2 \end{pmatrix} \rightarrow \begin{pmatrix} 20(10)/40 \\ 20(10)/40 \\ 20 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 20 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/20 & 0 \end{pmatrix} \begin{pmatrix} 10 \\ 30 \\ 40 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 40 \\ 2 \end{pmatrix} \rightarrow \begin{pmatrix} 20(10)/40 \\ 20(30)/40 \\ 20 \end{pmatrix} = \begin{pmatrix} 5 \\ 15 \\ 20 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/20 & 0 \end{pmatrix} \begin{pmatrix} 30 \\ 10 \\ 40 \\ 1 \end{pmatrix} = \begin{pmatrix} 30 \\ 10 \\ 40 \\ 2 \end{pmatrix} \rightarrow \begin{pmatrix} 20(30)/40 \\ 20(10)/40 \\ 20 \end{pmatrix} = \begin{pmatrix} 15 \\ 5 \\ 20 \end{pmatrix}$$



e) Advantages and disadvantages of clipping in 2D as opposed to 3D.

2D

· Clipping done at last stage of rendering ∴ complex to ensure not to draw outside.

3D

· use frustrum to clip within 6 planes

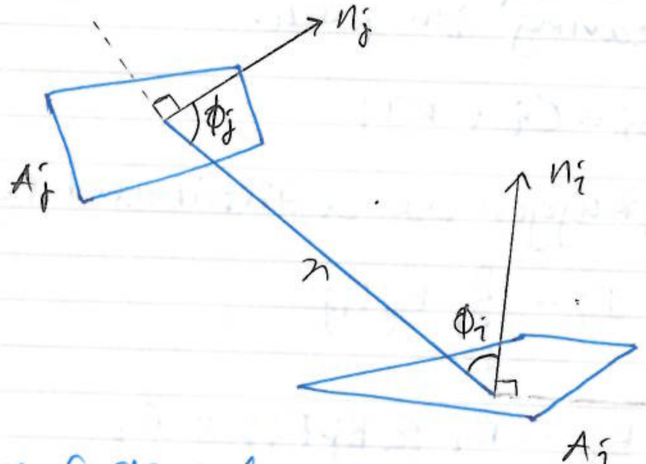
· simple; calculate clip plane equations – constraints if viewpoint is zero.

Q4. Radiosity

a) $$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

$$F_{ij} = \frac{\cos\phi_i \cos\phi_j |A_j|}{\pi r^2}$$

Explain role of form-factors in the radiosity method.



The form factors link every pair of patches and determines the proportion of radiated energy from one that strikes the other.

b) Describe the full matrix soln and the progressive refinement solution for computing radiosity.

① full matrix soln:

$B_i = E_i + R_i I_i$

$I_i = \sum_{j=1}^{n} B_j F_{ij}$ $\quad \therefore B_i = E_i + R_i \sum_j B_j F_{ij}$

$$B_i - \sum_j R_i B_j F_{ij} = E_i$$

$$\begin{pmatrix} 1 & -R_1F_{12} & -R_1F_{13} & \cdots & -R_1F_{1n} \\ -R_2F_{21} & 1 & -R_2F_{23} & \cdots & -R_2F_{2n} \\ -R_3F_{31} & -R_3F_{32} & 1 & \cdots & -R_3F_{3n} \\ \vdots & \cdots & \cdots & \cdots & \cdots \\ -R_nF_{n1} & -R_nF_{n2} & -R_nF_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}$$

- Radiosity is the Σ of energy emitted by the surface itself plus any reflected energy due to light arriving from other surfaces.

- We assume that the emitted energy is constant over a given surface and for a small area of the surface dA.

$$B\,dA = E\,dA + RI$$

- We divide the scene up into a # of polygons and B_i is energy leaving ith patch.

$$B_i = E_i + RI_i$$

- We treat each polygon as a distributed light source.

$$I_i = \sum_{j=1}^{n} B_j F_{ij}$$

then $$B_i - R_i \sum_j B_j F_{ij} = E_i$$

② Progressive refinement rdn:

We render the image after each iteration.
Evaluating B_i row-wise: Gathering

$$B_i^k = E_i + R_i \sum_j B_j^{k-1} F_{ij}$$

" " column-wise: Shooting

$$B_j^k = B_j^{k-1} + R_j\, F_{ji}\, \Delta B_i^{k-1}$$

② Better if the scene is very large and we need to re-calculate the form factors every time we need them.

① Bad due to computational strategies.

② Progressive refinement ~~[illegible]~~ method:

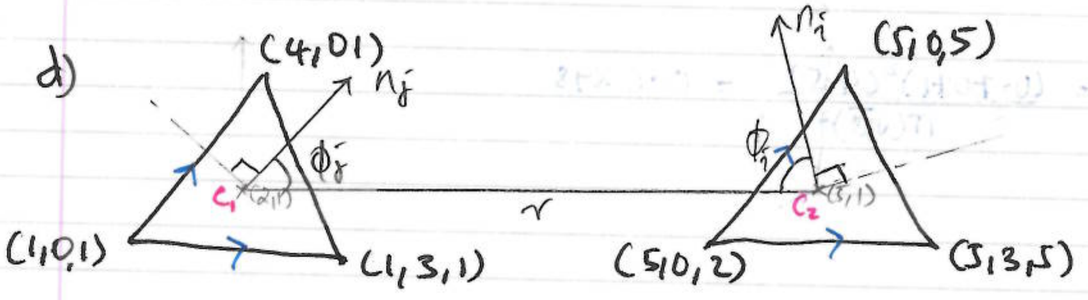
c) Why can't it be extended to include specular reflections?

- Specular reflections are direction dependent and involve the relative directions of the viewpoint and each light source.

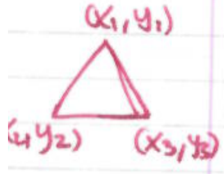
- This introduces problems because every patch is a light source and no longer points.

- Therefore we'd have to integrate incident light over a specular cone and computing specularities would be challenging.

d)



Calculate F_{ij} and F_{ji}. Use the centroids of the triangle to estimate distance.



Centroid formula $\quad C = \left(\frac{x_1, x_2, x_3}{3}, \frac{y_1, y_2, y_3}{3} \right)$

~~$C_1 = (3, 1)$~~ $\quad C_1 = (2, 1, 1)$

~~$C_2 = ($~~ $\quad C_2 = (5, 1, 4)$

$n_i = \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 9 \\ 0 \\ 0 \end{pmatrix} \qquad \hat{n}_i = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

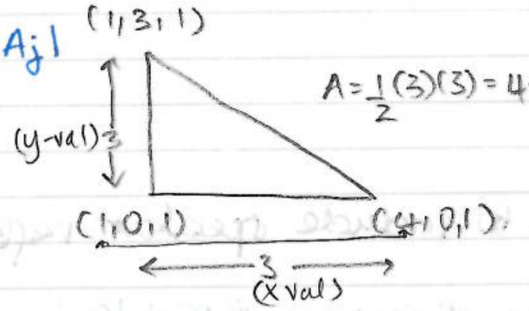
$n_j = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} \times \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -9 \end{pmatrix} \qquad \hat{n}_j = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$

$$F_{ij} = \frac{\cos\phi_i \cos\phi_j |A_j|}{\pi r^2} \qquad F_{ji} = \frac{\cos\phi_i \cos\phi_j |A_i|}{\pi r^2}$$

$i \longrightarrow j$

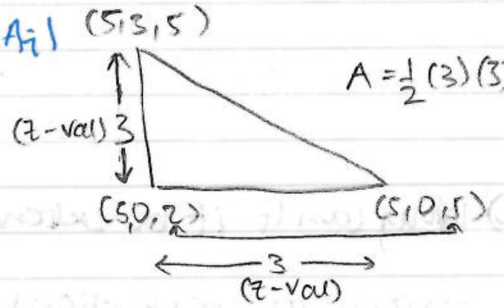
$\rightarrow$ $$\underline{r} = \underset{(i)}{\begin{pmatrix}5\\1\\4\end{pmatrix}} - \underset{(j)}{\begin{pmatrix}2\\1\\1\end{pmatrix}} = \begin{pmatrix}3\\0\\3\end{pmatrix} \qquad \hat{\underline{r}} = \begin{pmatrix}0.7071\\0\\0.7071\end{pmatrix} \qquad |r| = \sqrt{18}$$

$\rightarrow$ $|A_j|$



$A = \frac{1}{2}(3)(3) = 4.5$

$|A_i|$



$A = \frac{1}{2}(3)(3) = 4.5$

$\rightarrow$ $$\phi_i = \hat{\underline{n}}_i \cdot \hat{\underline{r}} = \begin{pmatrix}1\\0\\0\end{pmatrix} \cdot \begin{pmatrix}0.7071\\0\\0.7071\end{pmatrix} = 0.7071$$

$\rightarrow$ $$\phi_j = \hat{\underline{n}}_j \cdot \hat{\underline{r}} = \begin{pmatrix}0\\0\\-1\end{pmatrix} \cdot \begin{pmatrix}-0.7071\\0\\-0.7071\end{pmatrix} = 0.7071$$

$i \longleftarrow j$

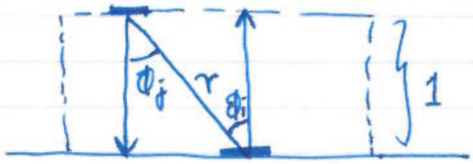
$$F_{ij} = \frac{(0.7071)^2(4.5)}{\pi(\sqrt{18})^2} = 0.0398$$

$$F_{ji} = \frac{(0.7071)^2(4.5)}{\pi(\sqrt{18})^2} = 0.0398$$

e) Hemicube method: How it's used to calculate form factors

- All patches that project onto the same area of the hemisphere have the same form factor.

- If the centre of a hemicube pixel is at coordinate $(x_p, y_p, 1)$



- The unit vector from the patch towards the origin:

$$\frac{1}{r}(-x_p, -y_p, -1)^T$$

where $r = \sqrt{x_p^2 + y_p^2 + 1}$

- The unit surface normal to the hemicube pixel is $(0, 0, -1)^T$

$$\cos\phi_i = 1/r$$

$$\cos\phi_j = 1/r$$

ΔA: Area of a hemicube pixel

- then $$\frac{\cos\phi_i \cos\phi_j \Delta A}{\pi r^2} = \frac{\Delta A}{\pi r^4}$$