1.

a

i)

the point that is directly under the light source.

ii)

the point where the light can reflect into the viewer’s eye

iii)

It won’t change

2.

i)

E = 0.4 \* A + 0.6 \* B

ii)

F = 0.6 \* C + 0.4 \* D

iii)

G = 0.2 \* E + 0.8 \* F

iv)

From i ii and iii we know that:

G = 0.12 \* B + 0.48 \* C + 0.32 \* D

so we just interpolate the color using the above coefficient which gives us

(30, 150, 0) ^ T

v)

0.12 \* n\_B + 0.48 \* n\_C + 0.32 \* n\_D

c

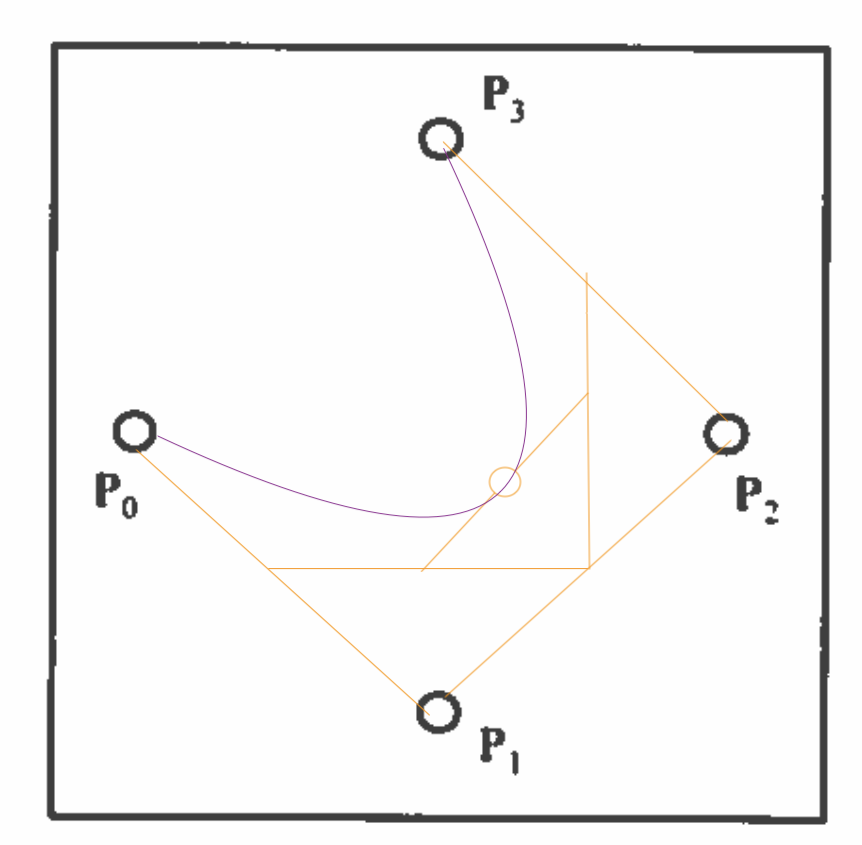
h = (l + v) / mod(l + v)

Blinn Phong lighting model is more efficient than the Phong since the half way vector calculated above can be considered as constant while Phone model needs to recalculate the reflection vector for every pixel due to the change of curvature.

2.

TODO

a



b

a\_0(\mu) = (1 - \mu) ^ 3

a\_1(\mu) = 3 \* \mu \* (1 - \mu) ^ 2

a\_2(\mu) = 3 \* \mu ^ {2} \* (1 - \mu) ^ 1

a\_3(\mu) = \mu ^ 3

c

Cannot ensure C^1 continuity in the point where it starts and also end

d

i) a\_0(\mu) > 0 because 0 <= (1 - \mu) <= 1 => 0 <= (1 - \mu) ^ 3 <= 1

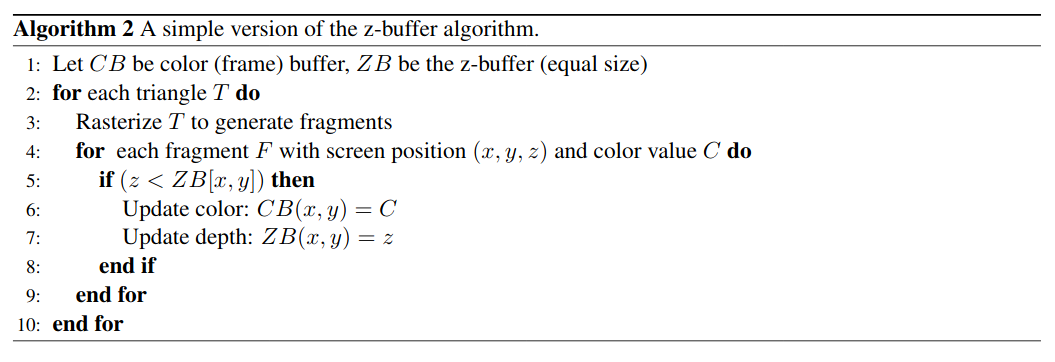
other follows similarly

ii)

\sigma^{3}\_{i = 0} a\_i(\mu) = ((1 - \mu) + \mu) ^ {3} = 1

3.

a)



Initialize the colour buffer CB

Initialize the depth buffer DB

For each triangle T:

Rasterize the triangle and obtains the Fragments

For each fragment F with coordinate (x, y, z) and colour c:

if DB(x, y) < z:

DB(x, y) = z

CB(x, y) = c

b)

TODO

c)

super-sampling: sample higher resolution

convolution filtering: use a convolution filter the smooth out the image

d)

super-sampling:

pros:

work well with filled polygons

cons:

slow

convolution filtering:

pros:

fast, can be implemented in hardware

cons:

degrade image

e

TODO: don’t know please help.

4.

a

a - uniform scaling

b - non-uniform scaling

c - rotation

d - orthographic projection

e - perspective projection

f identity

b

[[1, 0, 0, 0],

[0, -1, 0, -2y\_0],

[0, 0, 1, 0],

[0, 0, 0, 1]]

c

[[-1, -1, 0],

[ 1, 0, 0],

[0, 0, 1]]

d

[[1, 0, 0, 0],

[0, 1, 0, 0],

[0, 0, 1, 0],

[0, 0, 0.5, 0]]