TEXTURE FILTERING AND MIPMAPPING

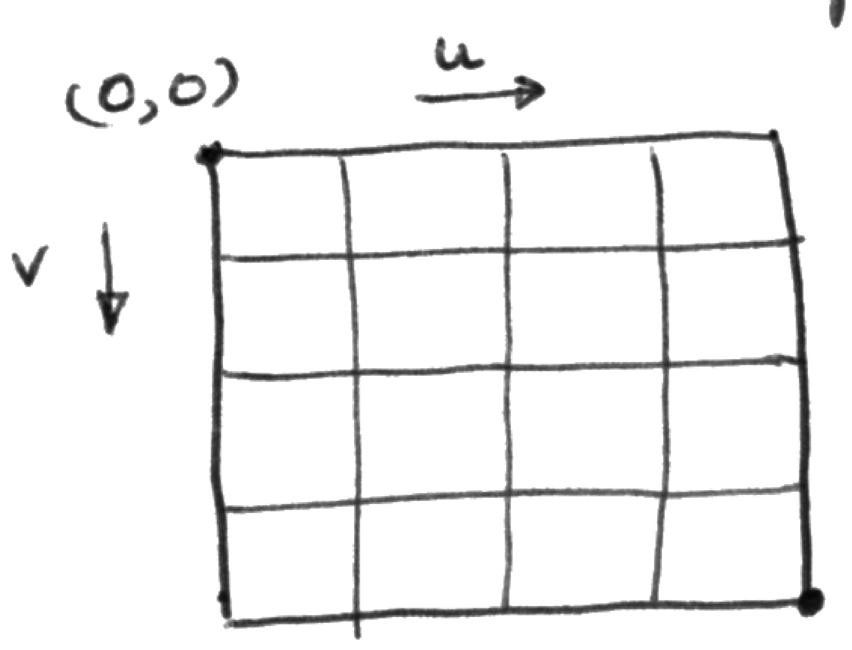
→ A mipmap is a pyramid of progressively averaged and downsampled versions of a base texture.

→ For eg. Given a base texture size of 256X256, the associated mipmap may contain a series of 8 images, each $1/4^{th}$ the total area of the previous one: 128X128 pixels, 64X64 pixels, 32X32 pixels 16X16 pixels, 8X8 pixels, 4X4 pixels, 2X2 pixels, 1 pixel.

→ To form every subsequent level, 4 pixels from the previous (higher) level in the pyramid are averaged and the averaged value is used as the value for a pixel at the next level.

→ Texture filtering strategies decide that given a (u, v, d) how is/are texture samples queried and combined to get the information for the pixel on screen.

→ Given a texture of size 4X4 it is overlain with a (u, v) coordinate space as shown in the figure.



Now given any (u,v) value, texture filtering allows us to figure out which texel(s) the texture coordinates map to.

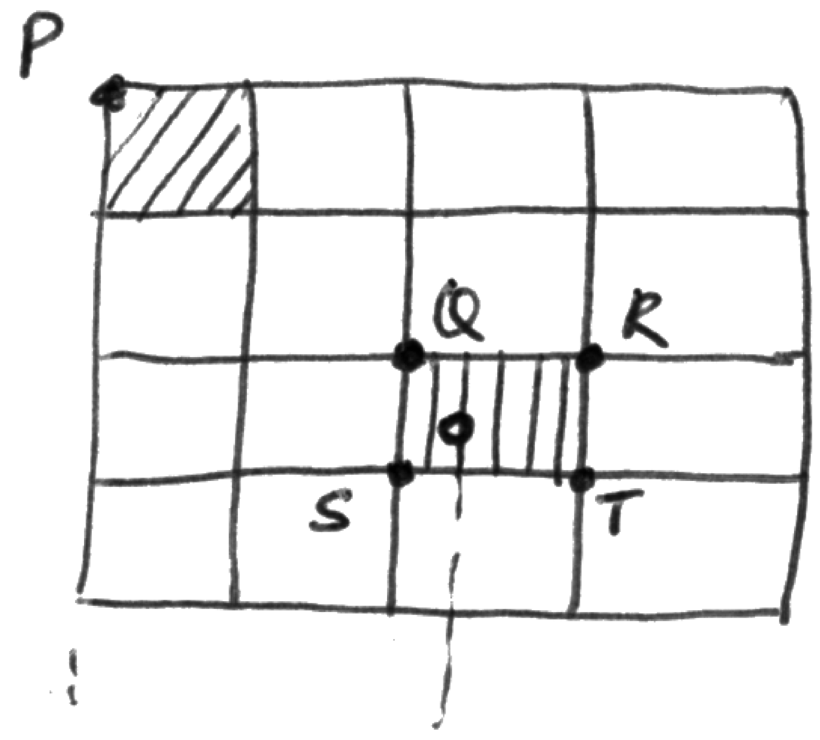
a) Nearest-neighbour filtering → (Also called "Point Sampling/Filtering")

In this scheme, the texel that is nearest to the given (u,v) values is used. The intensity of this texel is mapped back to the onscreen pixel.

b) Bilinear filtering

Here a <u>weighted average</u> of the 4 closest texels to the given (u,v) is computed.

If we assume a 4x4 texture where every texel is referenced by a integral value that represents it top-left corner.



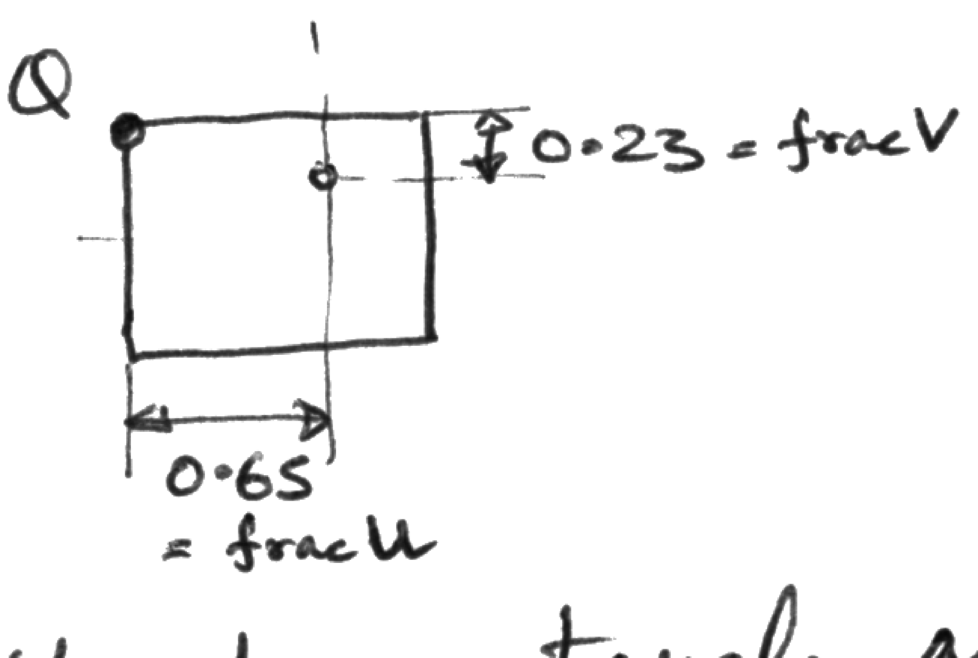
For eg. $P \equiv (0,0)$ represents the texel shaded

$Q \equiv (2,2)$ represents the texel shaded

So now if get a $(u,v) \equiv (2.65, 2.23)$

then the four closest texels to this are Q, R, S and T

Inside the texel Q, the fractional components of u and v tell us the position of the actual point being queried on the texture



This gives us the weights for the four texels as.

(a) $fracU * fracV$ for Q
(b) $(1-fracU) * fracV$ for R
(c) $fracU * (1-fracV)$ for S
(d) $(1-fracU) * (1-fracV)$ for T

So the bilinearly filtered value returned is

$$I = fracU . fracV . Q + (1-fracU)\, fracV . R + fracU . (1-fracV) S + (1-fracU)(1-fracV) T$$

where Q, R, S, T represent the intensity of these texels in the above equation.

c) Trilinear Filtering

When using mipmapping, a third option possible is to combine the bilinearly filtered intensities returned from two successive mipmaps by linearly interpolating between them using the 'd' coordinate.

- Note that when the value of d is such that only the first or last mipmap level is used trilinear filtering reduces to bilinear filtering
- The cost of trilinear filtering is approximately double that of bilinear filtering.

d) Anisotropic filtering

- In anisotropic filtering, more than one trilinear filterings are carried out.
- For eg. for 8 × anisotropic filtering – 8 sample locations are identified on <u>each</u> mipmap as per some sampling pattern. Then they (i.e., each pair) are combined using trilinear filtering. Then the 8 intensity values obtained after trilinear filtering are combined to get the final value.
- As a measure of relative cost, 8 × anisotropic filtering is 8 times more expensive than trilinear filtering.