

Rendering - Part III

→ (Texture / Bump) Mapping: ways to add surface detail

→ Texture: Image √ Procedural

defined by
image and
modifies pixels

compute pattern based on algot
rules

↳ texels: pixels when using image as texture

↳ why is texture defined in its own coordinate system?

→ Texture mapping ⊂ Rasterisation

→ Blend texel and pixel colour] important

→ Mesh texturing: risk of seams (segmenting shape up)

→ (pixels and texture image)'s resolutions almost
NEVER match

→ moving camera always ensures the reso's stay different

→ Fixing Res. Mismatches: (pixel > texture) resolution

⊗ Higher pixel reso: n pixels mapped to 1 texel

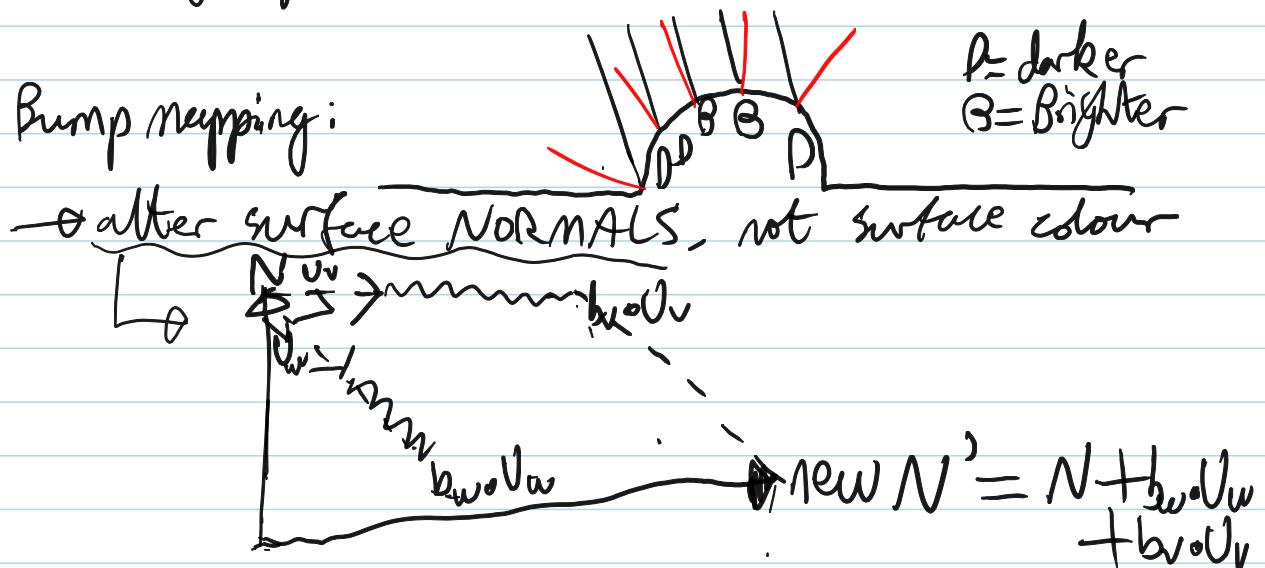
→ Bilinear interpolation filter: compute texel colour
through averaging stuff

⊗ Higher texel reso: 1 texels to 1 pixel

→ mipmaping: have set of texture maps, select
which map to use according to DISTANCE of pixel from viewer

- Repeatedly create smaller versions of a texture by downsampling each time → UNTIL the texture reached
- pre-processed, store each texture version in memory
- Textures: nOpen GL
 - add illumination to real-time scene
 - Lightings = rendered textures

- Bump mapping:



- how to derive BUMP values b_w and b_v ?

