Textures

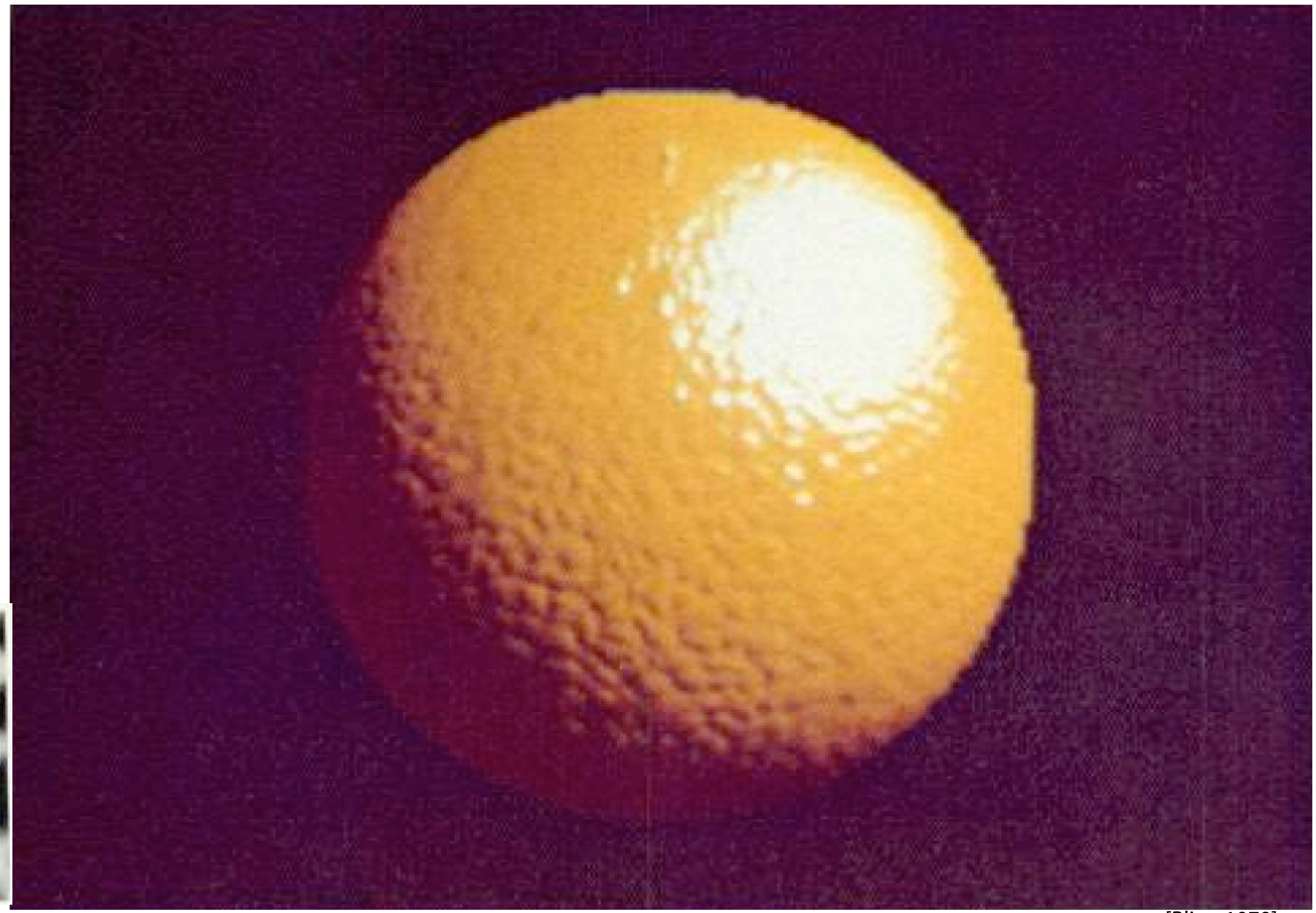
CS 4620 Lecture 22

Announcements

- Prelim grading this Friday
  - Will discuss it on Monday
  - Still few stragglers on taking the prelim

Bump mapping

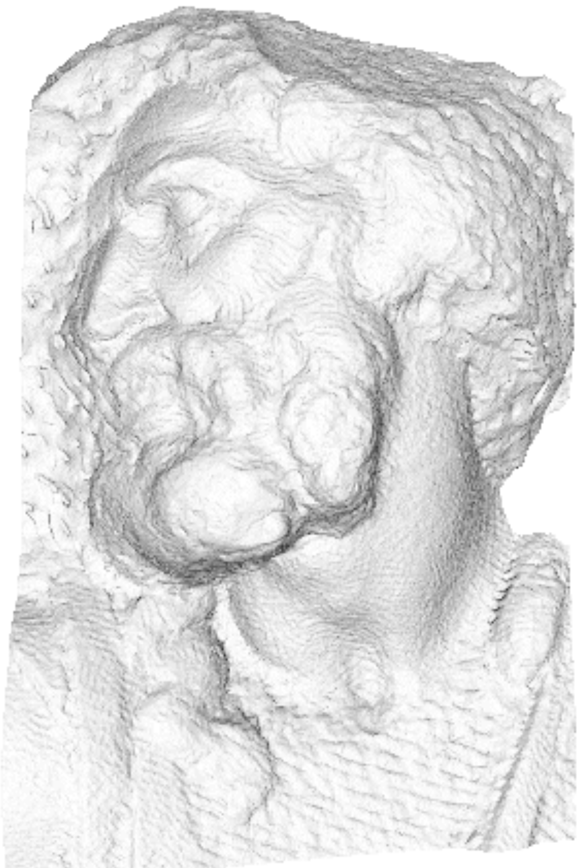
- Perturbs normal based on input grayscale height field



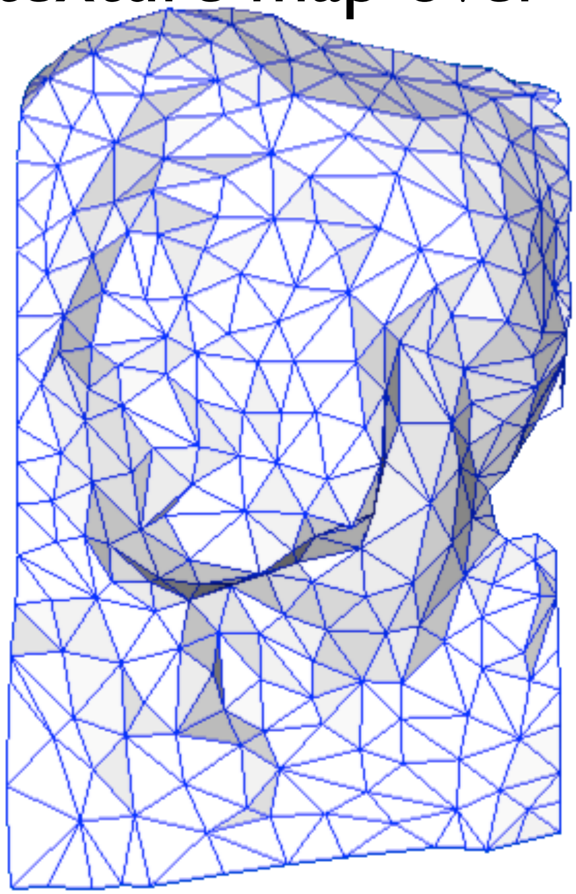
[Blinn 1978]

Normal mapping

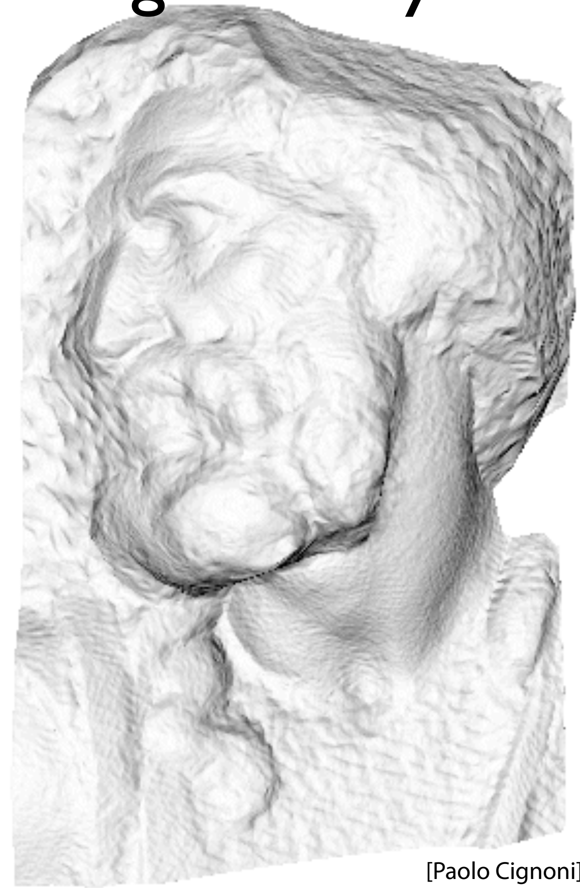
- Stores normals as texture map over coarse geometry



original mesh
4M triangles

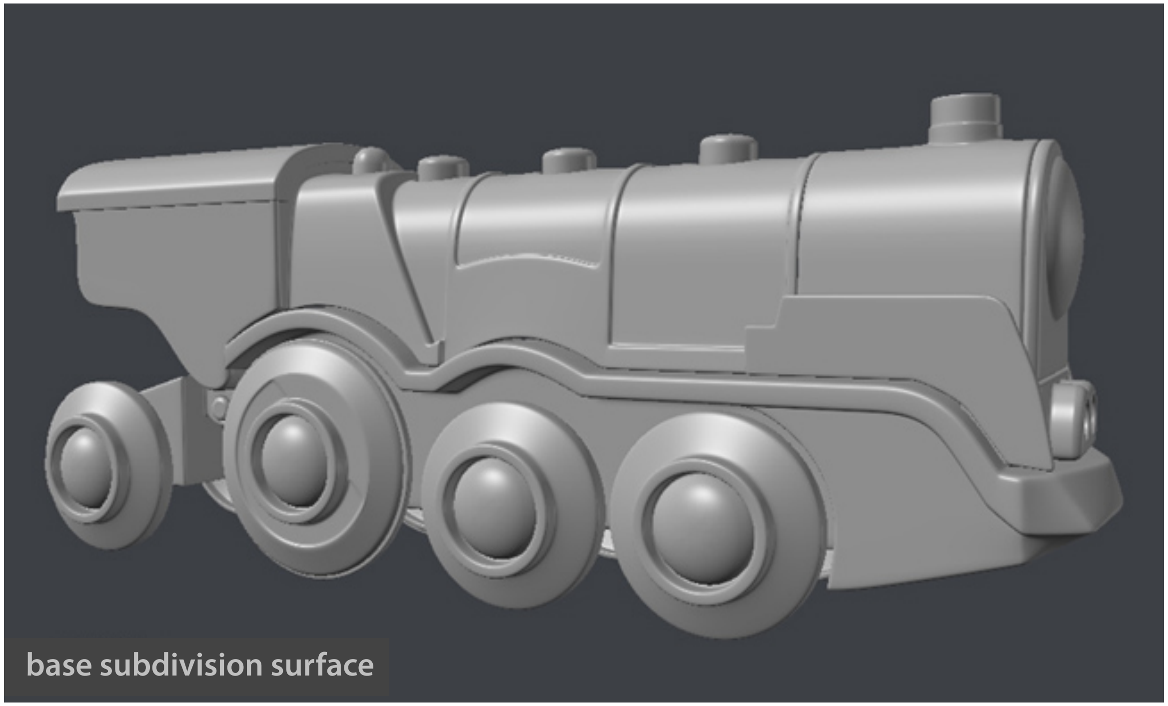


simplified mesh
500 triangles



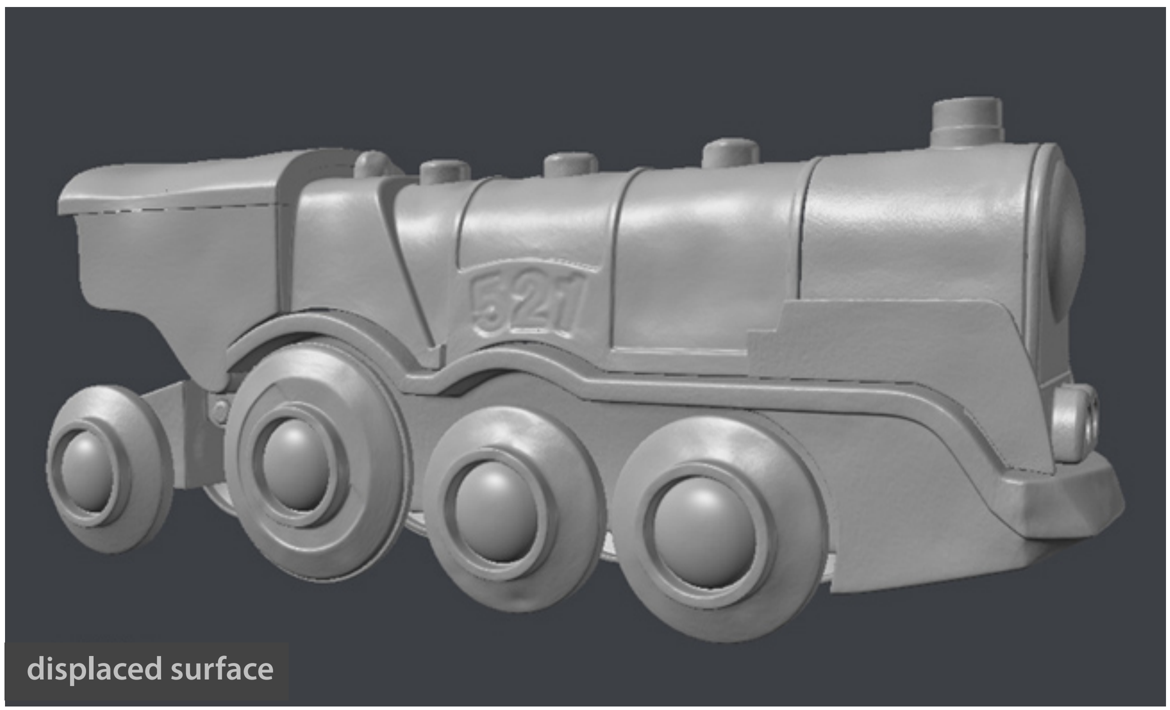
[Paolo Cignoni]

simplified mesh
and normal mapping
500 triangles



base subdivision surface

hand-painted displacement map (detail)



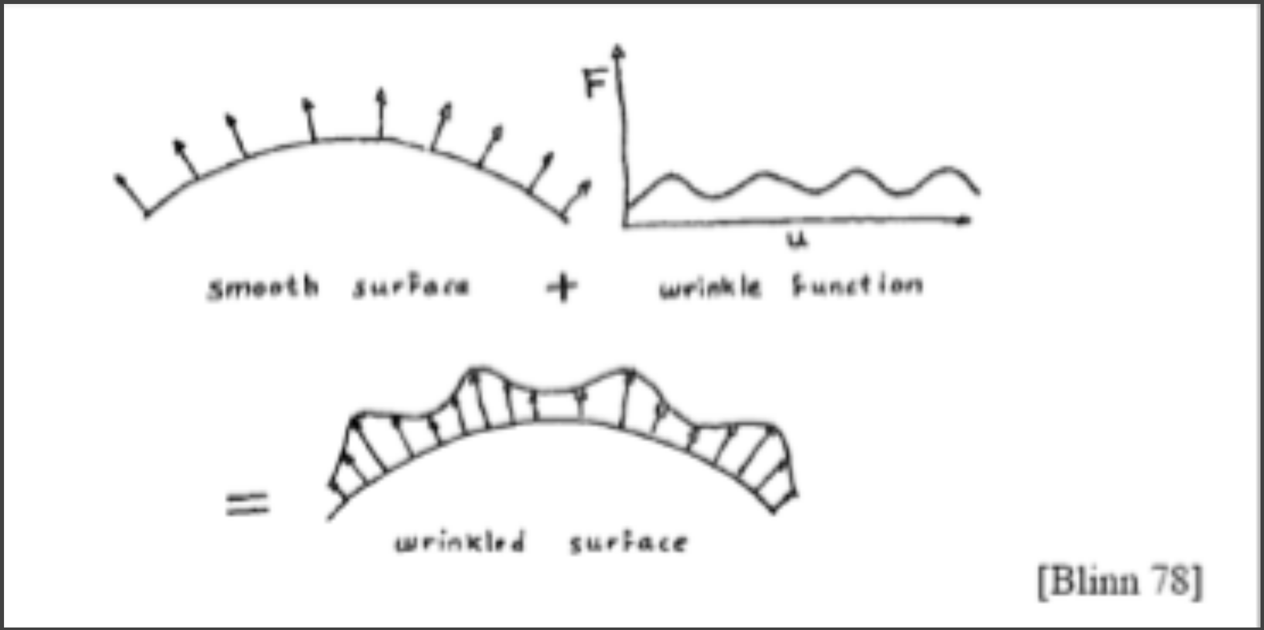
displaced surface

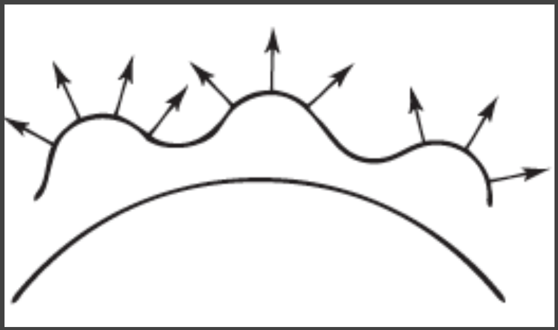
Paweł Filip
tolas.wordpress.com


521
fryrender
physically-based render engine
©2007 Paweł Filip

Bump mapping

- "Simulation of Wrinkled Surfaces" Blinn 78





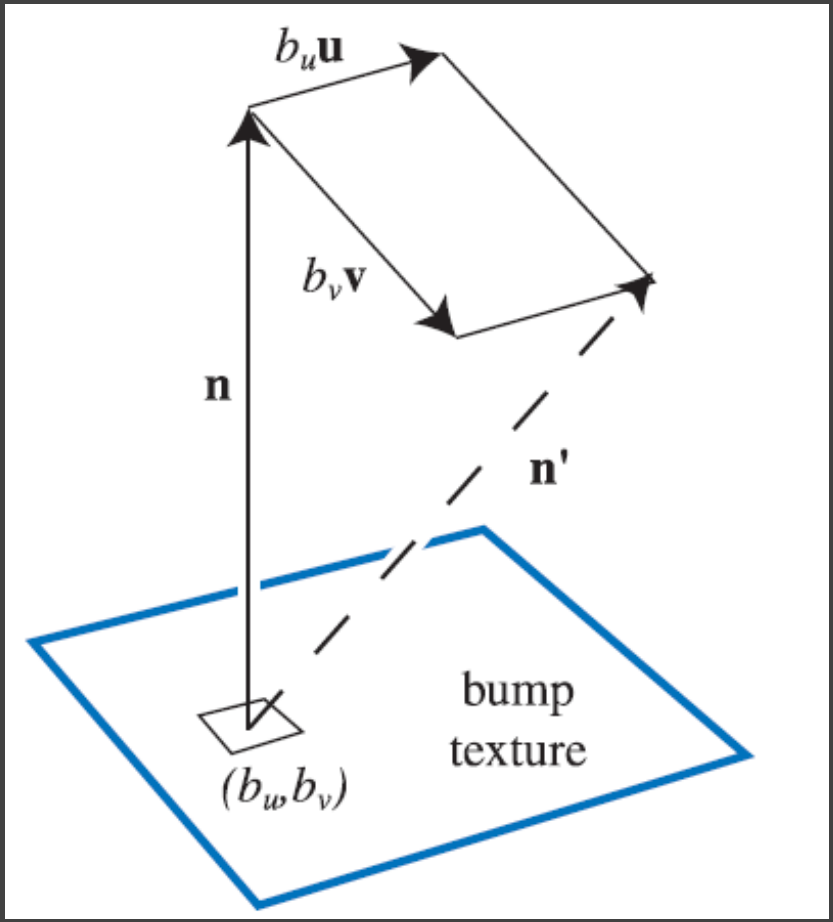
- Blinn: keep surface, use new normals

Bump Mapping

- Alter normals of surface
  - Only affects shading normals
- Also, mimics effect of small scale geometry (detail)
  - Except at silhouette
  - Adds perceived bumps, wrinkles

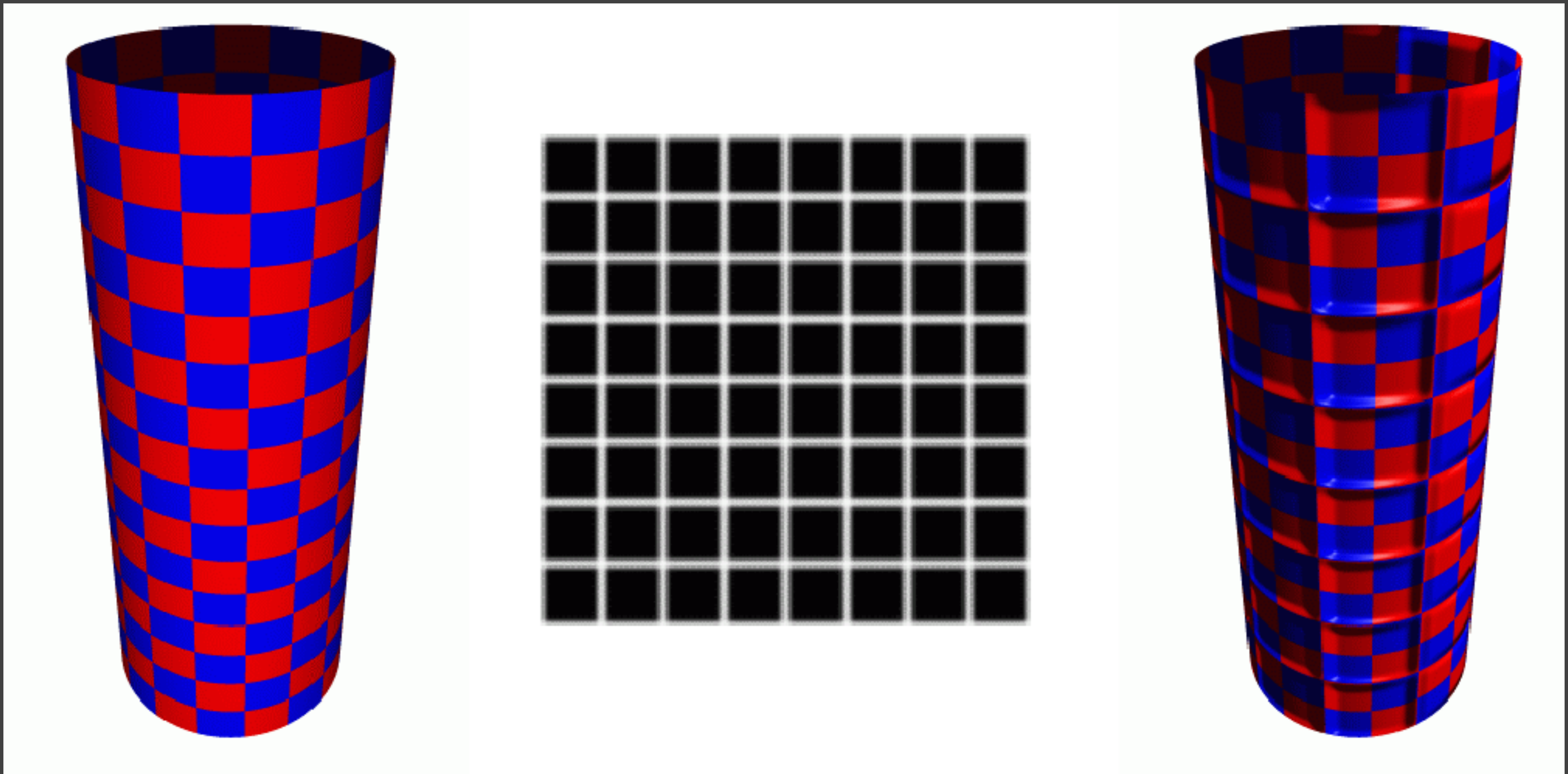
Blinn’s Original Method

- Look up bu and bv
- N’ is not normalized



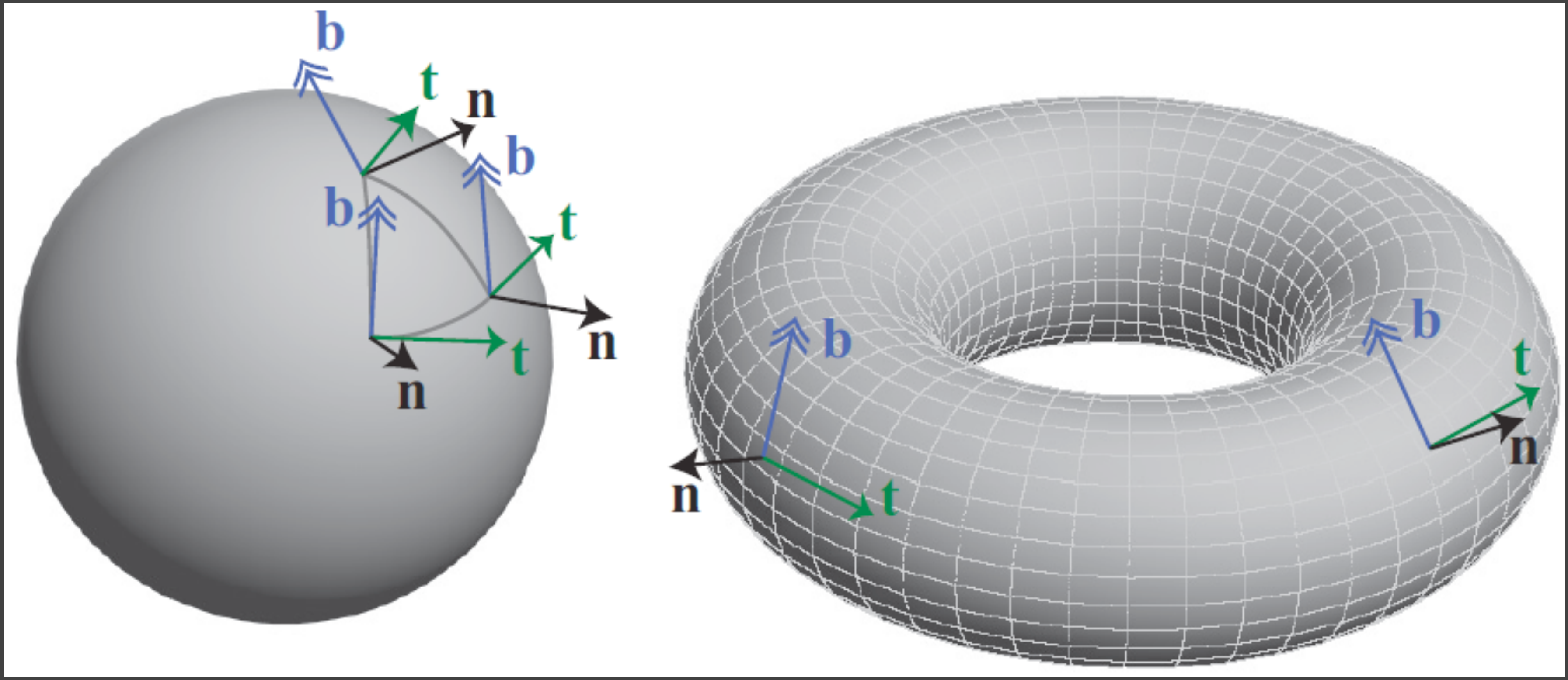
Bump Mapping

Bump Mapping



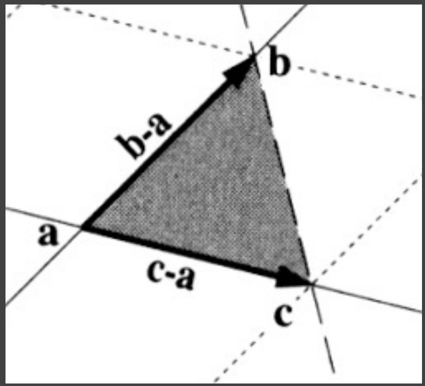
Before Bump Mapping

- First, need some frame of reference
  - Normal is modified with respect to that
  - Have tangent space basis: t and b
  - Normal, tangent and bitangent vectors



Tangent Space Basis Vectors for an Arbitrary Mesh

- http://www.terathon.com/code/tangent.html
- Find T and B for a triangle (a,b,c)
  - so that $Q - a = (u - u_0)T + (v - v_0)B$
  - T and B are tangent vectors aligned to the TM

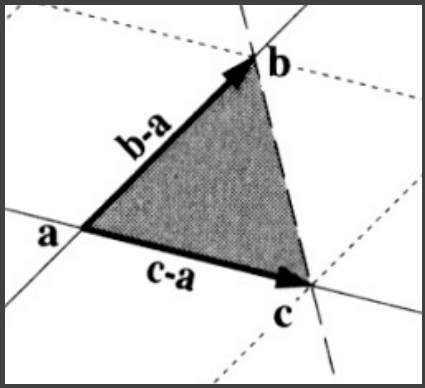


Tangent Space Basis Vectors for an Arbitrary Mesh

- http://www.terathon.com/code/tangent.html
- Find T and B for a triangle (a,b,c)
  - so that $Q - a = (u - u_0)T + (v - v_0)B$
  - T and B are tangent vectors aligned to the TM

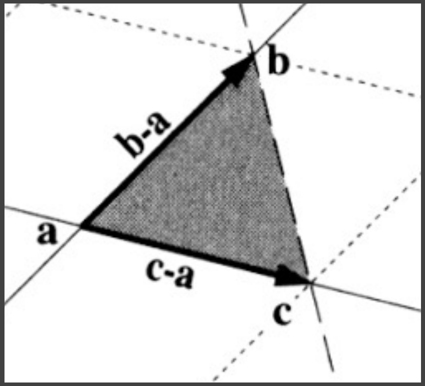
$$Q_1 = (c - a), Q_2 = (b - a)$$

$$(s_1, t_1) = (u_1 - u_0, v_1 - v_0)$$
$$(s_2, t_2) = (u_2 - u_0, v_2 - v_0)$$



Tangent Space Basis Vectors for an Arbitrary Mesh

$$Q_1 = s_1 T + t_1 B$$
$$Q_2 = s_2 T + t_2 B$$



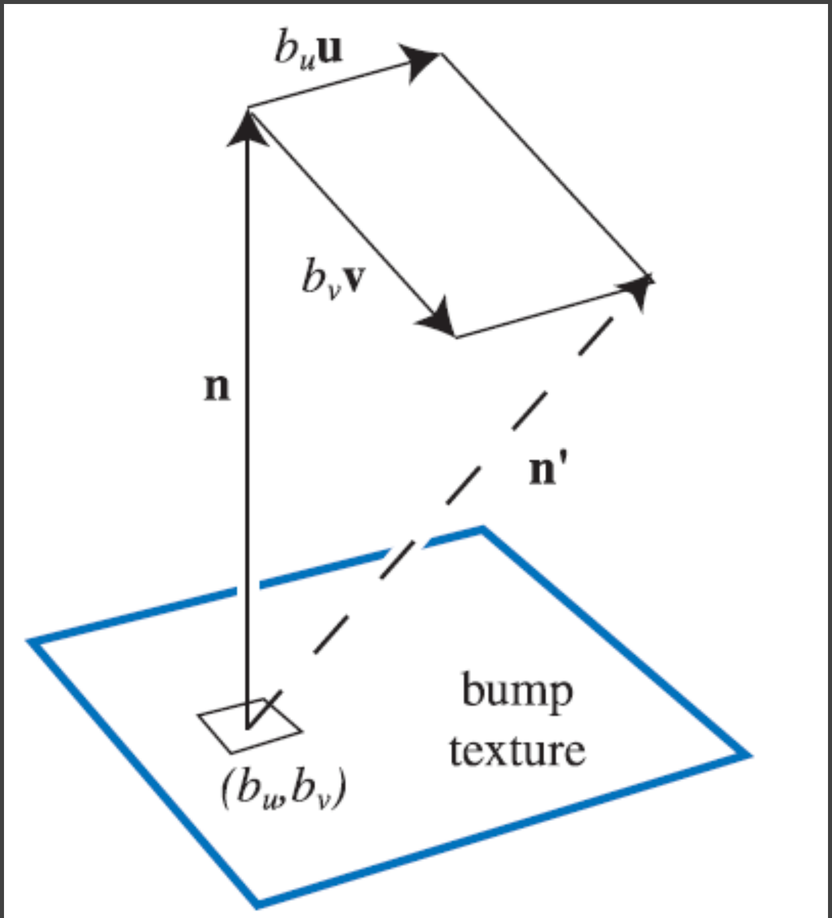
$$\begin{bmatrix} Q_1^x & Q_1^y & Q_1^z \\ Q_2^x & Q_2^y & Q_2^z \end{bmatrix} = \begin{bmatrix} s_1 & t_1 \\ s_2 & t_2 \end{bmatrix} \begin{bmatrix} T^x & T^y & T^z \\ B^x & B^y & B^z \end{bmatrix}$$

Tangent Space Basis Vectors for an Arbitrary Mesh

- For tangent vectors of a single vertex
  - Average tangents for tris sharing the vertex
  - Same as normal
- B = N x T
- N’ = N + bu T + bv B = N + bu U + bv V

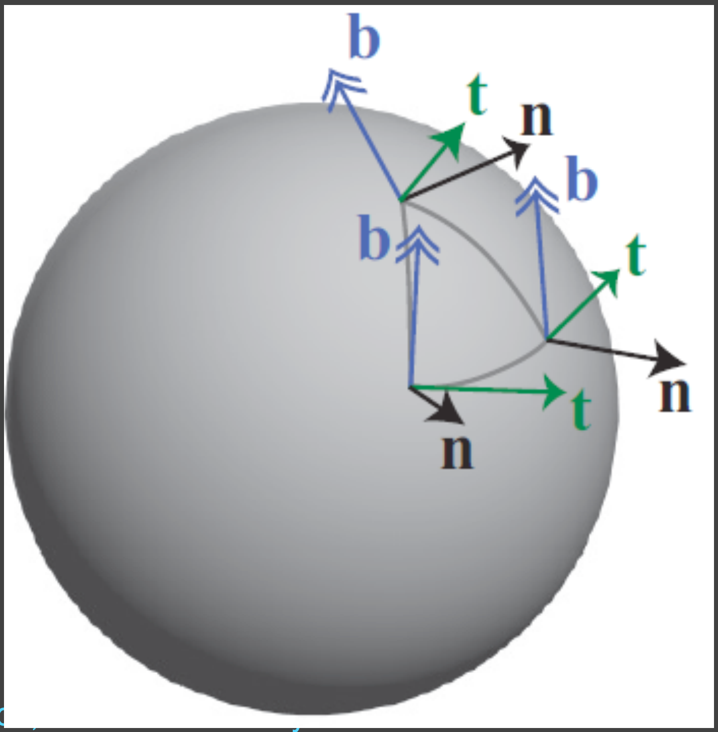
Blinn’s Original Method

- Look up bu and bv
- N’ is not normalized



Rendering with Bump Maps

- N'.L
- Perturb N to get N' using bump map
- Transform L to tangent space of surface
  - Have N, T (tangent), bitangent B = T x N



Transforming into this space

- Transform light vector into tangent space using following basis matrix

$$\begin{bmatrix} T_x & T_y & T_z & 0 \\ B_x & B_y & B_z & 0 \\ N_x & N_y & N_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Normal Maps

- Preferred technique for bump mapping for modern graphics cards
- Store new normals in texture map
  - Encodes (x, y, z) mapped to [-1, 1]
- More memory but lower computation

Normal Map

Height Map

- Store

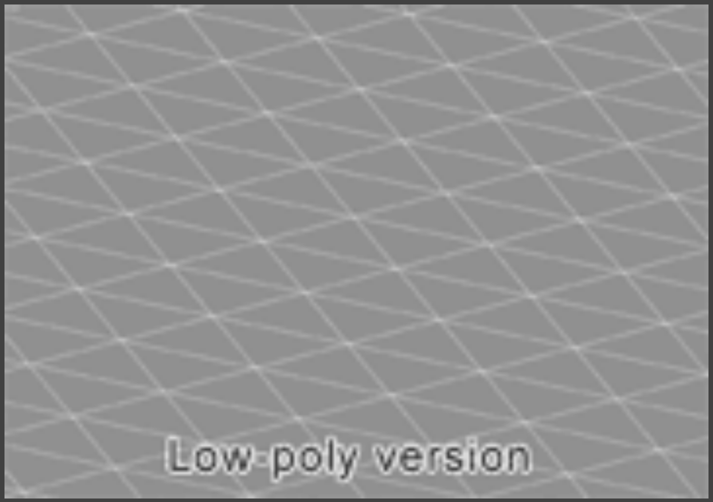
```
colorComponent = 0.5 * normalComponent + 0.5
```

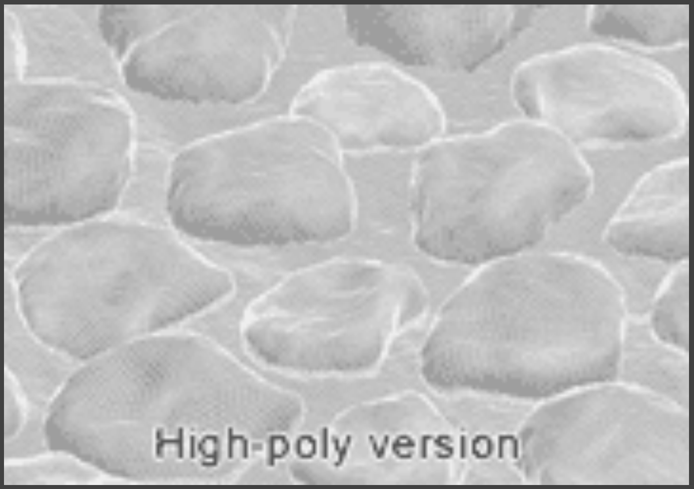
-

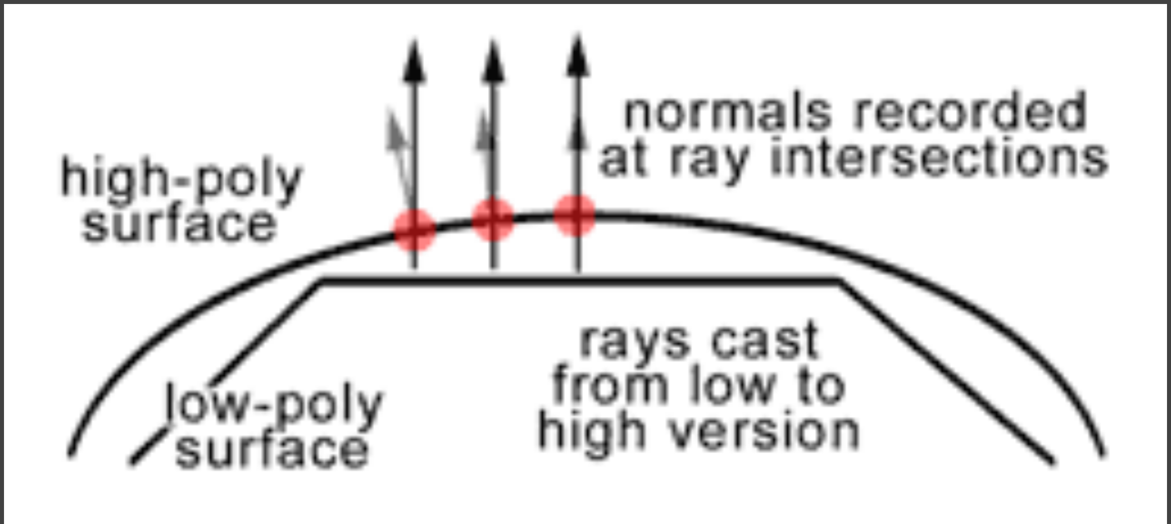
- Use

```
normalComponent = 2* colorComponent -1
```

Creating Normal Maps







Creating Normal Maps

- First create complex geometry
- Simplify (in modeling time) to simple mesh with normal map

Normal Map

Which space is normal map in?

- World space
  - Easy computation
    - Get normal
    - Get light vector
    - Compute shading
  - Can we use the same normal map for…
    - two walls
    - A rotating object

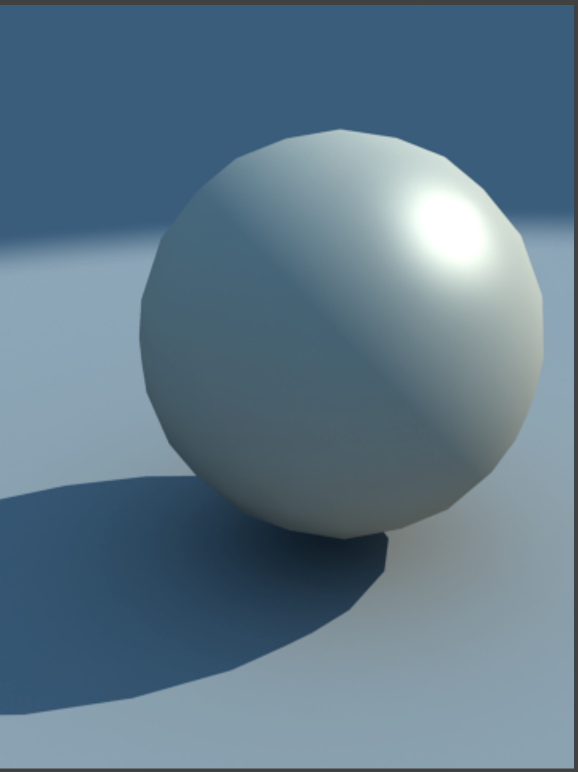
- Object space
  - Better, but cannot be reused for symmetric parts of object

Which space is normal?

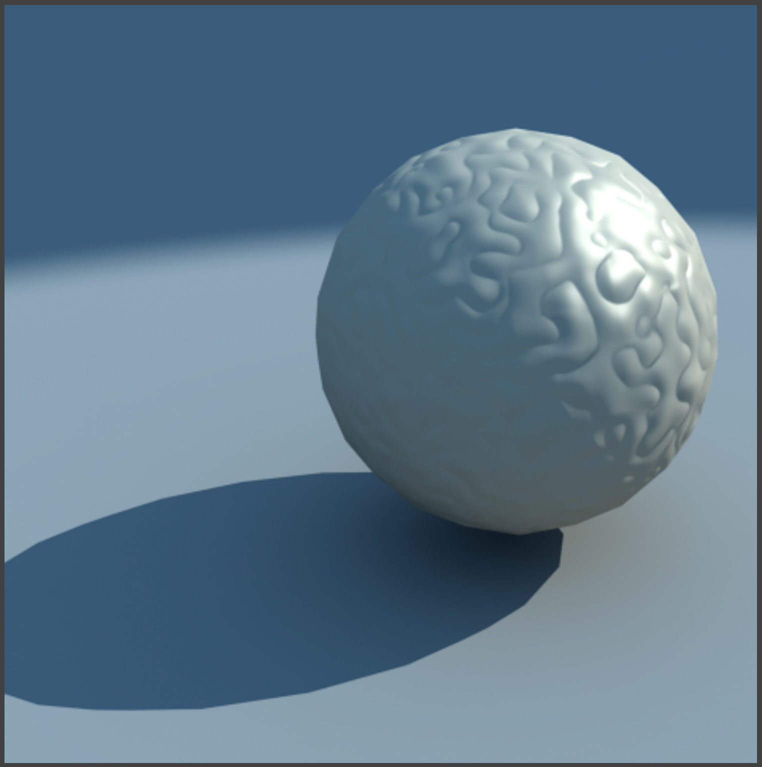
- Tangent space normals
  - Can reuse for deforming surfaces
  - Transform lighting to this space and shade

Displacement and Bump/Normal Mapping

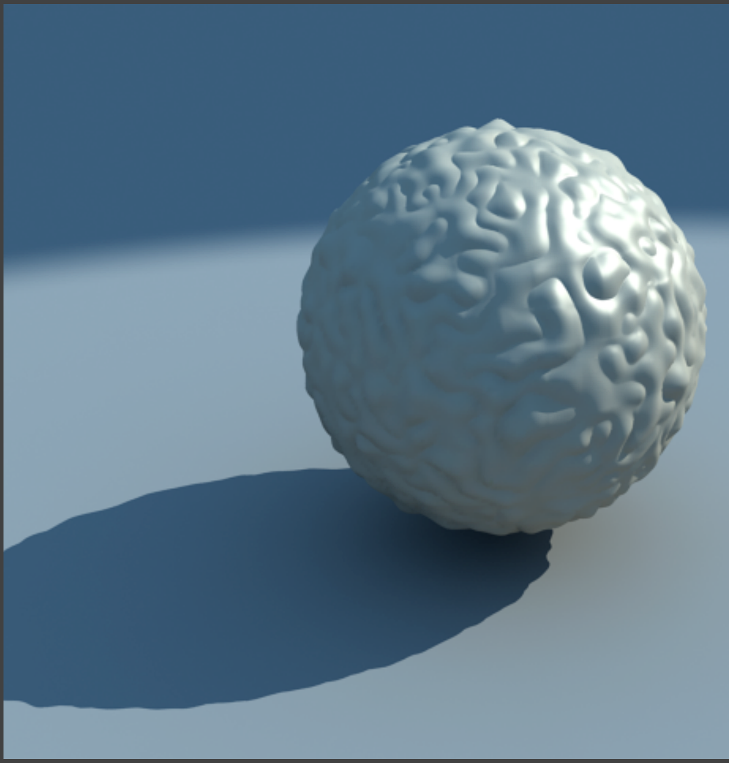
- Mimic effect of geometric detail/meso geometry
  - Also detail mapping



Geometry

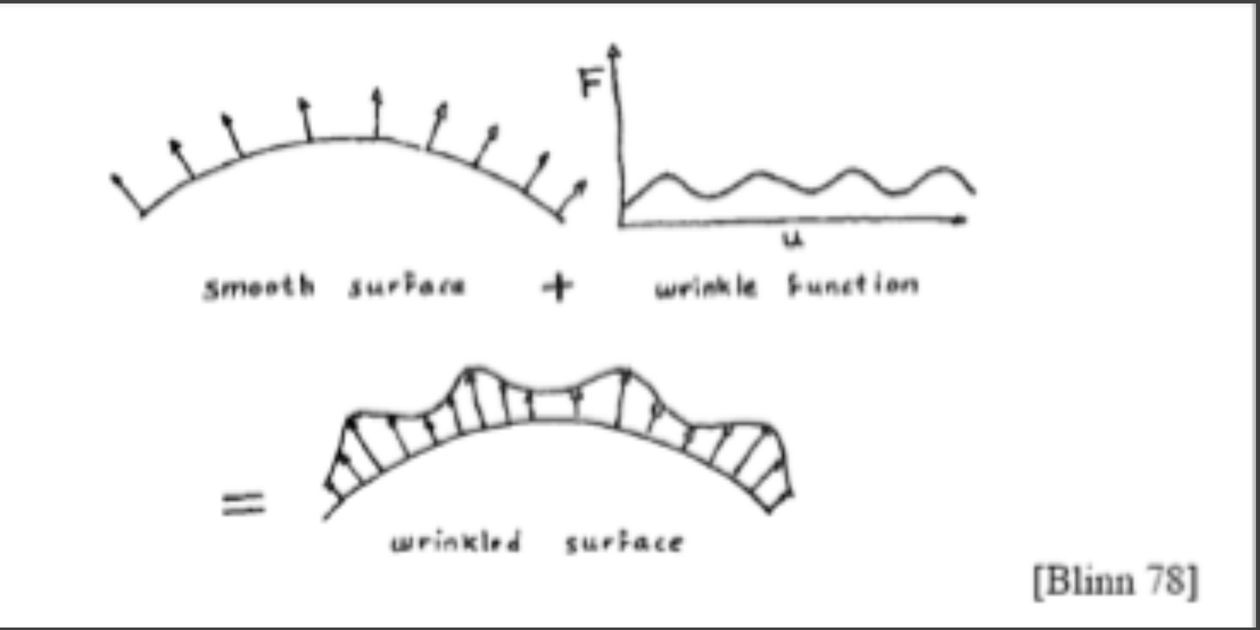


Bump mapping



Displacement mapping

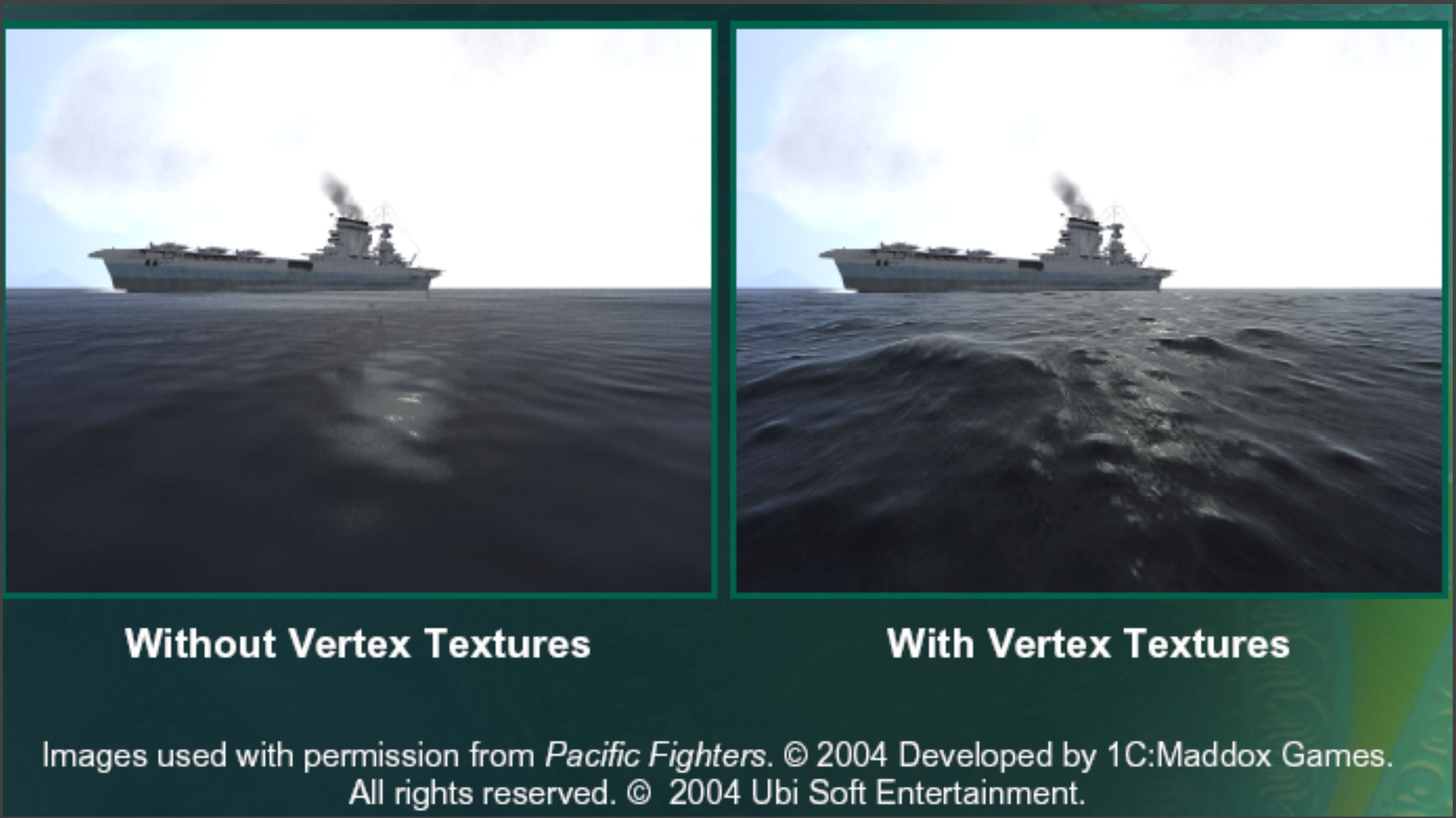
Displacement Mapping



$$P_{new} = P_{old} + DM(u) * \hat{N}$$

Displacement Maps

Displacement Maps: where?



Texture Maps

- Most flexible part of graphics hardware
- Textures can modulate
  - Material
    - Diffuse, Specular/roughness (gloss maps)
  - Geometry
    - Positions
      - displacement mapping
    - Normals
      - bump mapping, normal mapping
  - Lighting
    - Environment mapping
    - Reflection mapping
    - Shadow mapping