Real - Time Rendering

Lighting / Shading

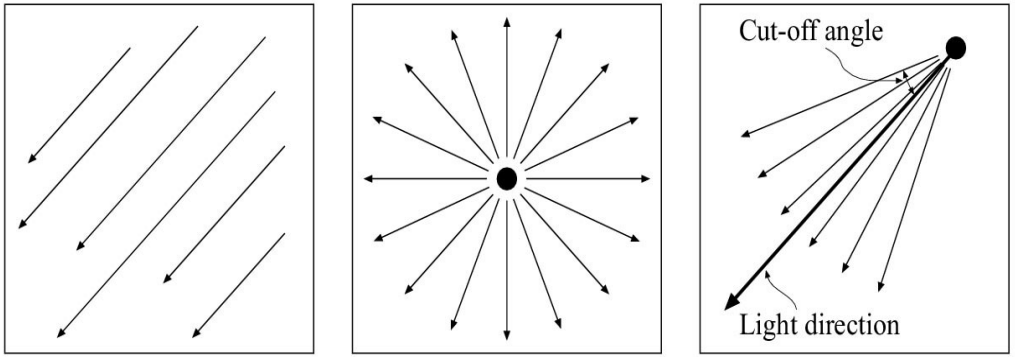
Michal Červeňanský
Juraj Starinský

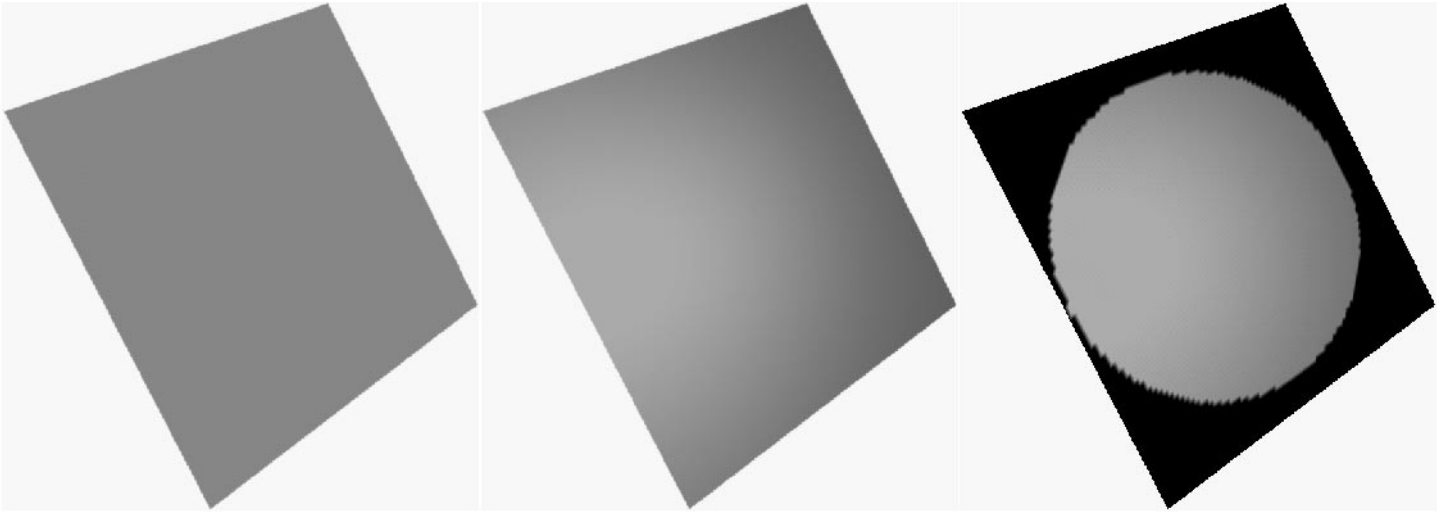
Overview

- Light Sources
- Lighting
- Shading
- Texturing
- Alpha Mapping
- Gloss Mapping
- Light Mapping
- Rough Surface

Light Sources

- Directional light
- Point light
- Spot light
- Colour, intensity





Lighting model

- Phong model
- Colour, Intensity
- Ambient, Diffuse, Specular

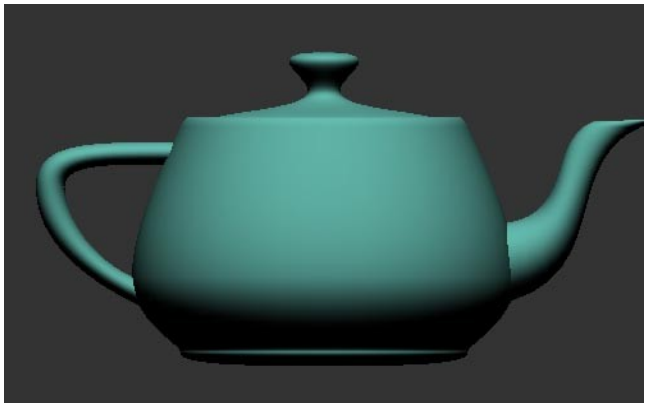
Lighting model - ambient

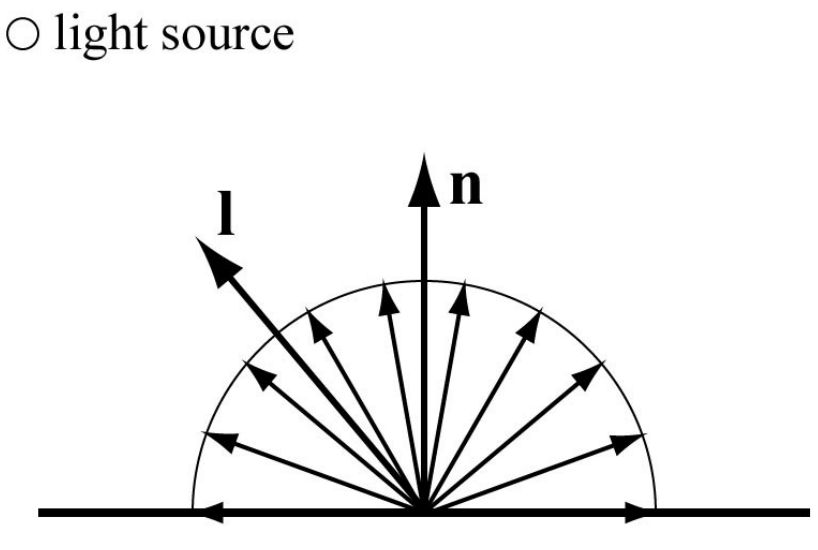
- Constant colour
- Light form other surfaces

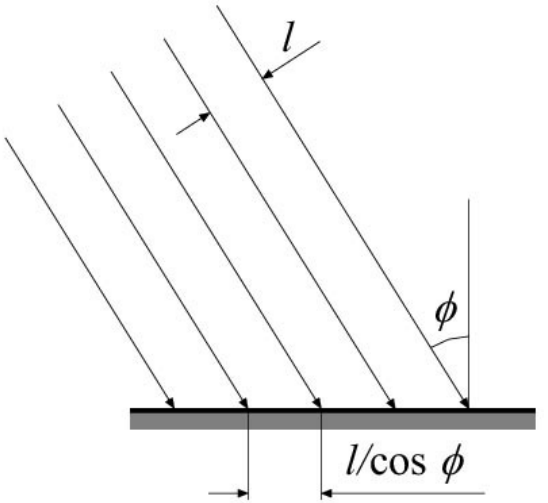
Lighting model - diffuse

- Lambert's law $i_{diff} = \mathbf{n} \cdot \mathbf{l} = \cos\phi$
- Photons are scattered equally

$$\mathbf{i}_{diff} = (\mathbf{n} \cdot \mathbf{l})\mathbf{m}_{diff} \otimes \mathbf{s}_{diff}$$







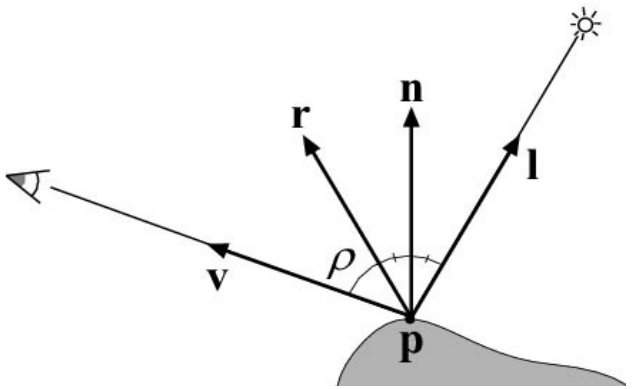
Lighting model - specular

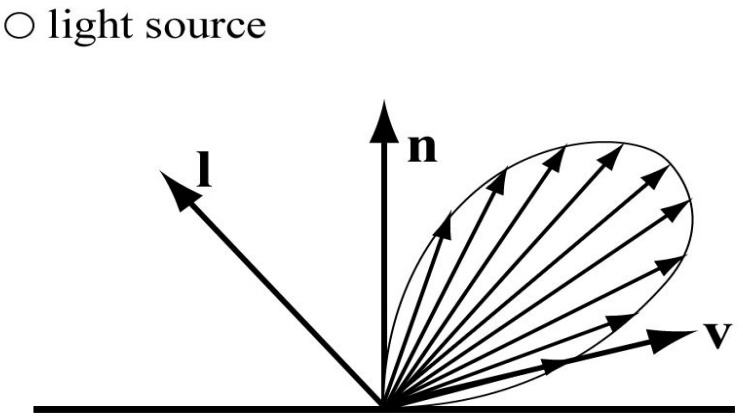
- The highlight simulation

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

$$i_{spec} = (\mathbf{r} \cdot \mathbf{v})^{m_{shi}} = (\cos \rho)^{m_{shi}}$$

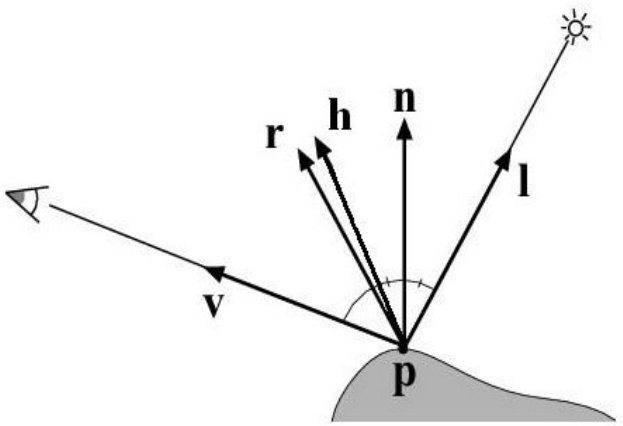
$$\mathbf{i}_{spec} = \max(0, (\mathbf{r} \cdot \mathbf{v})^{m_{shi}})\mathbf{m}_{spec} \otimes \mathbf{s}_{spec}$$





Lighting model (2)

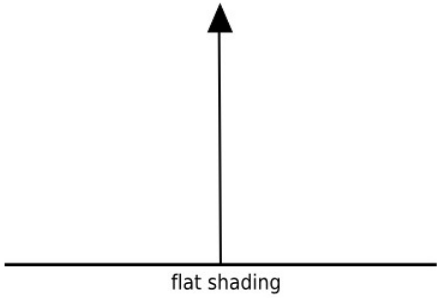
- Half vector $H = \frac{L+V}{|L+V|}$
- Speed-up
- $i = i_{amb} + i_{diff} + i_{spec}$

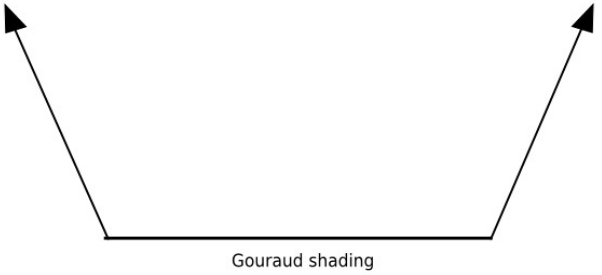


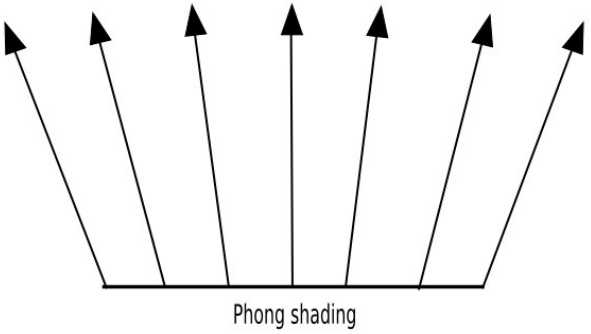

```
float4 color = tex2D(TexMaterial, IN.texcoord1);
float3 normal = (tex2D(TexNormal, IN.texcoord1)-0.5).rgb;

float3 vecToEye   = normalize(posEye   - IN.FragPos);
float3 vecToLight = normalize(posLight - IN.FragPos);

float3 vecHalf = normalize(vecToLight+vecToEye);

float difLight = max(dot(normalize(normal), vecToLight),0);
float specLight= pow(max(dot(normalize(normal),vecHalf),0),15);

if(difLight <= 0) specLight = 0;

float3 difColor  = difLight*color.rgb;
float3 specColor = specLight*color.rgb;
float3 ambColor  = float3(0.1,0.1,0.1);

OUT.color.xyz = difColor + specColor + ambColor;
```

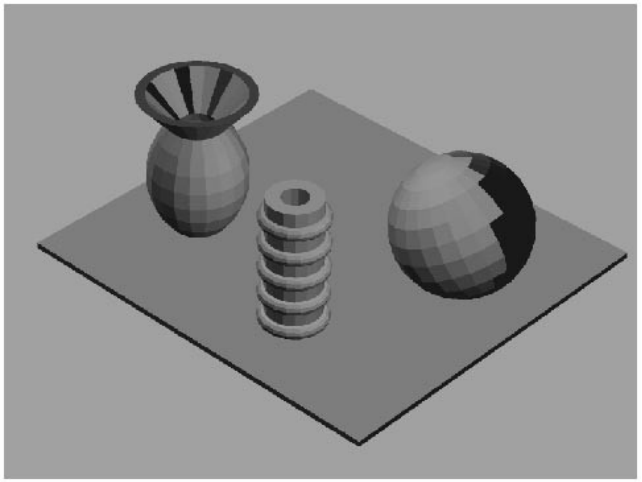
Shading

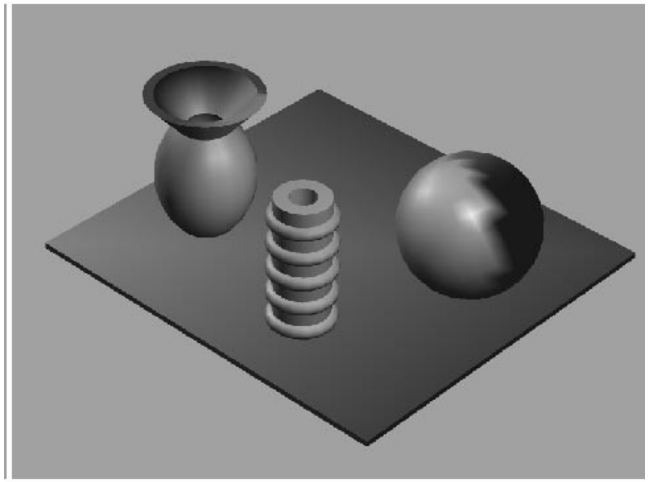
- Flat, Gouraud, Phong
- Per - primitive, vertex, fragment

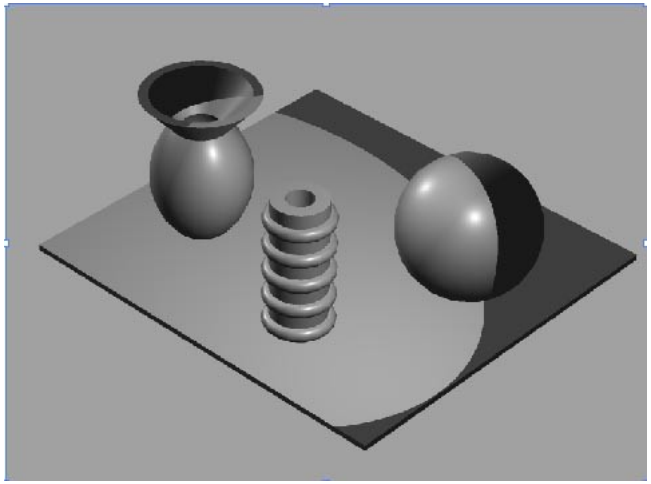






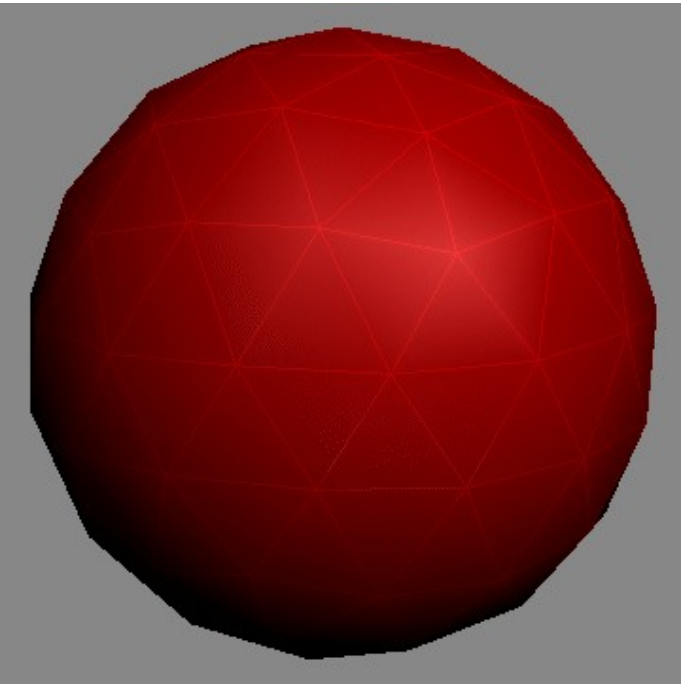
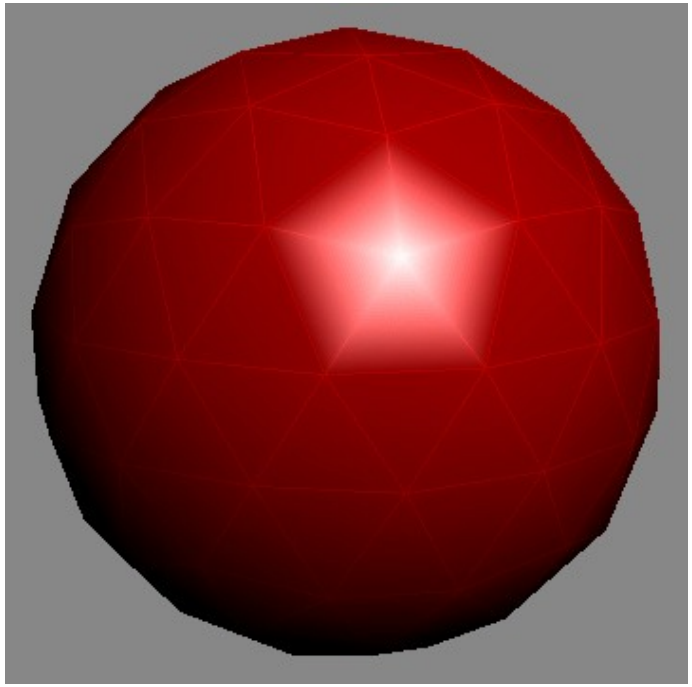
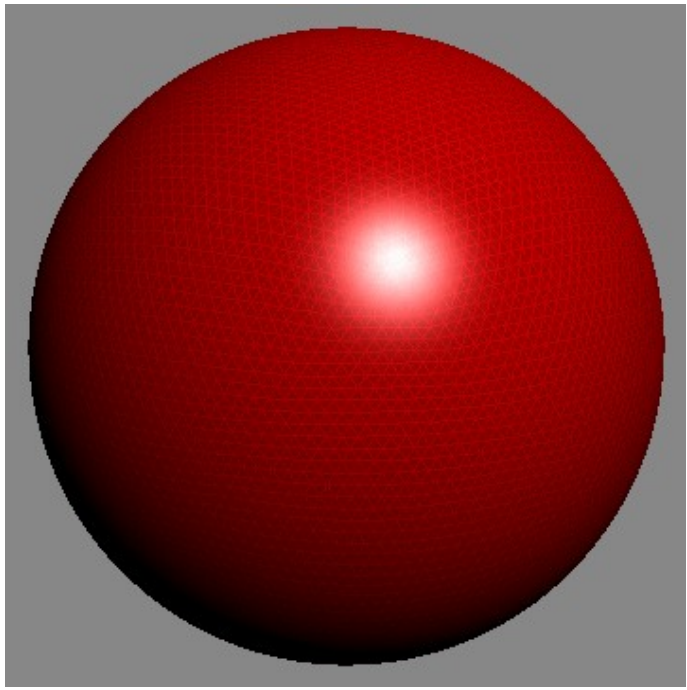






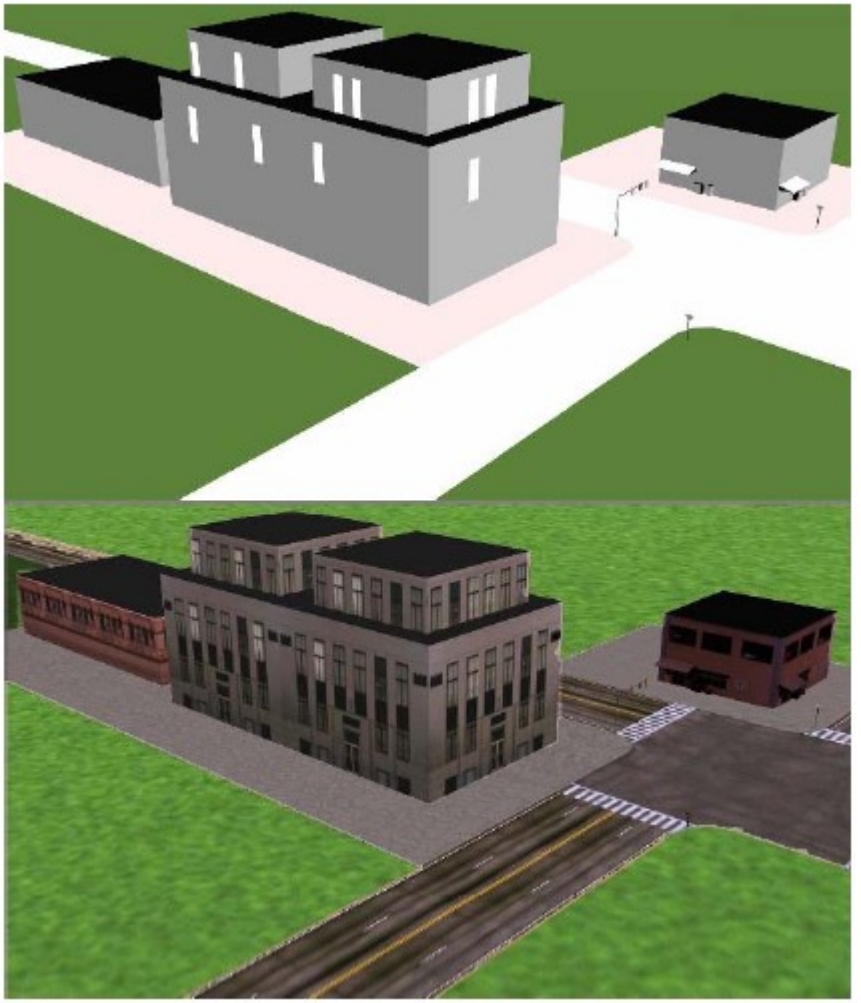
Shading (2)

- Tessellation is important

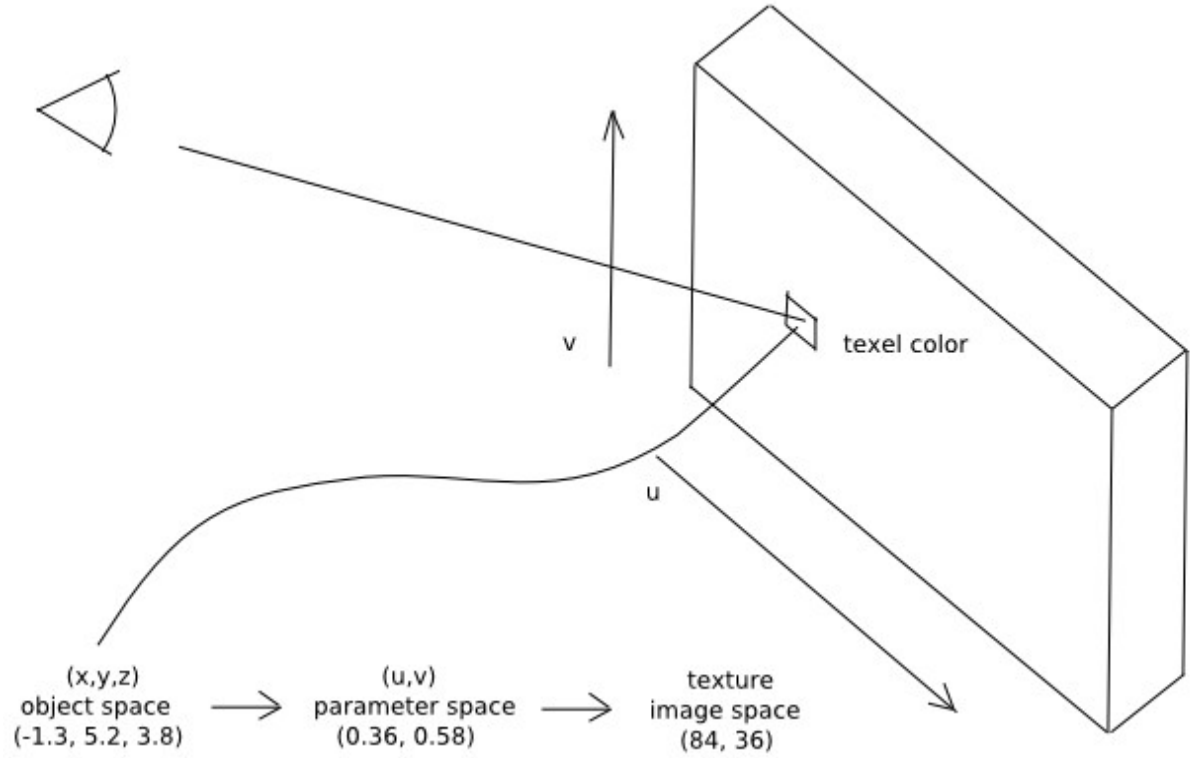
Texturing

- Light Sources
- Lighting
- Shading
- Texturing
  - Texture mapping
  - Texture settings
  - Texture filtering
- Alpha Mapping
- Environment Mapping
- Gloss Mapping



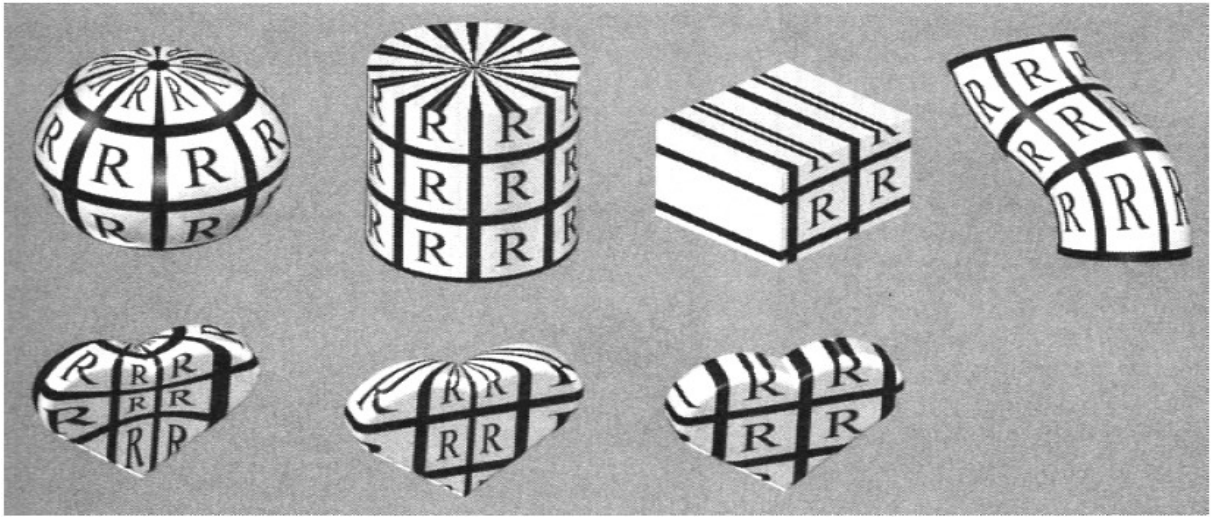
Texture mapping

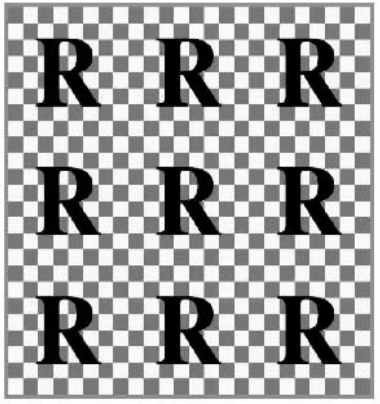
- 1D, 2D, 3D, Cube
- Texture transformations
- Texture projections (spherical, cylindrical, planar, natural)

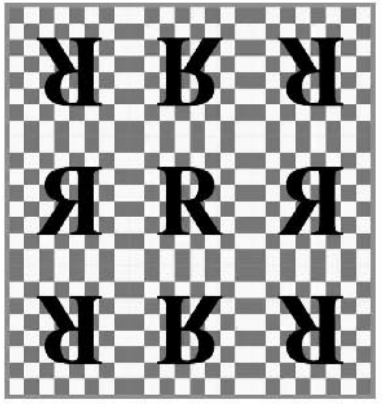


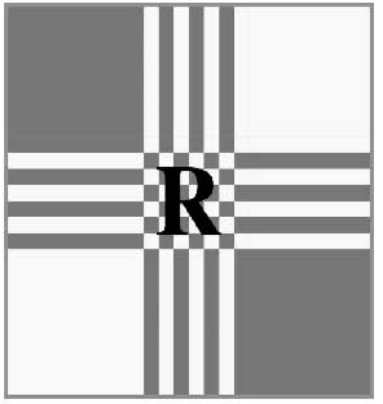
Texture settings

- Wrap modes (repeat, mirror, clamp, border)
- Compression (DXT, S3C, …)
- Multi-texturing



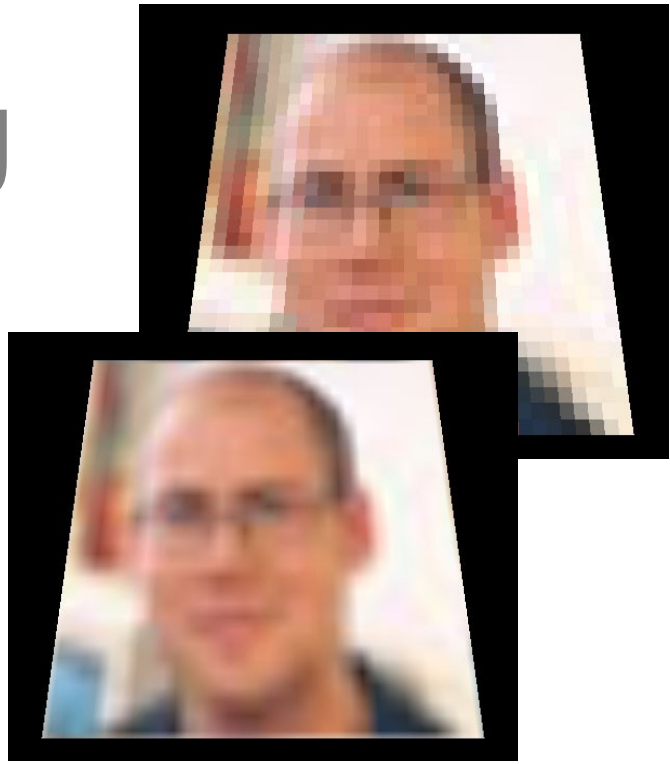


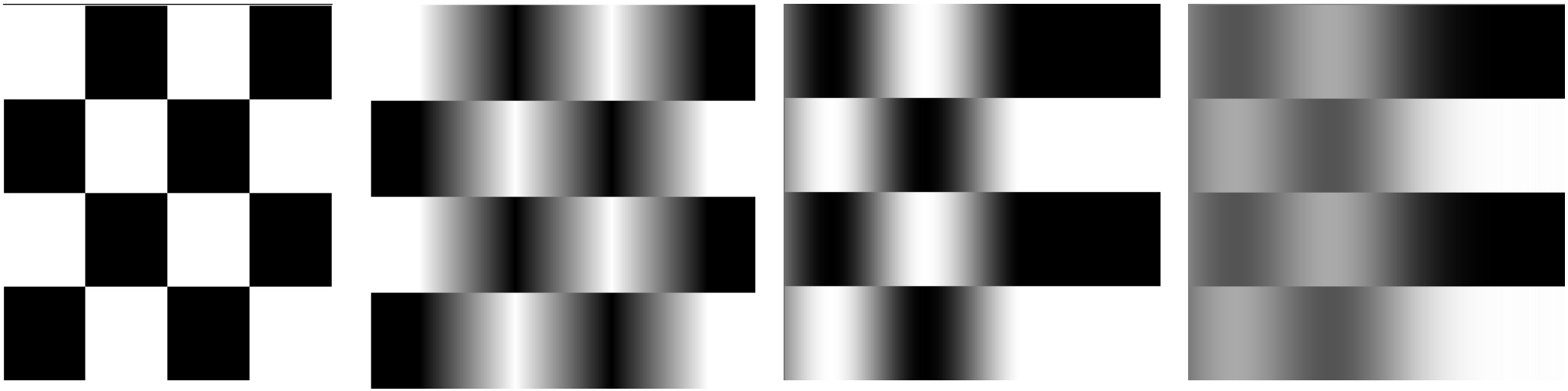




Texture filtering

- Nearest
- Linear – (tri/bi)linear interpolation
- Bi-cubic – ATI or shaders

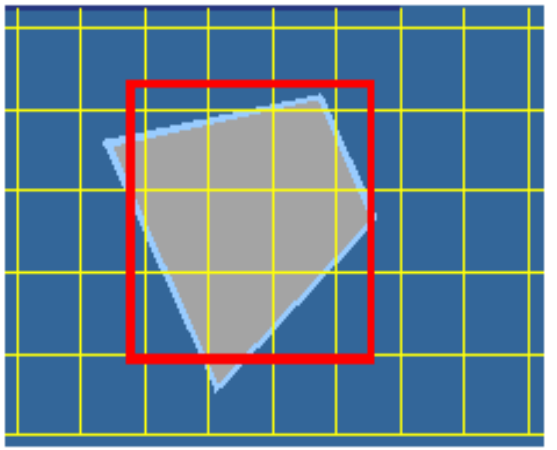


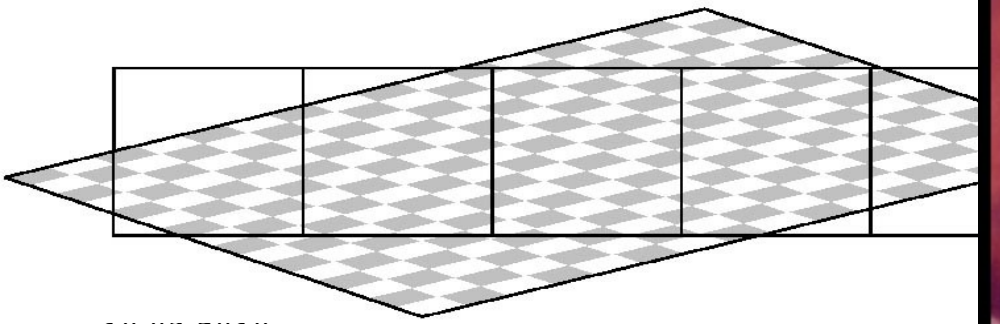


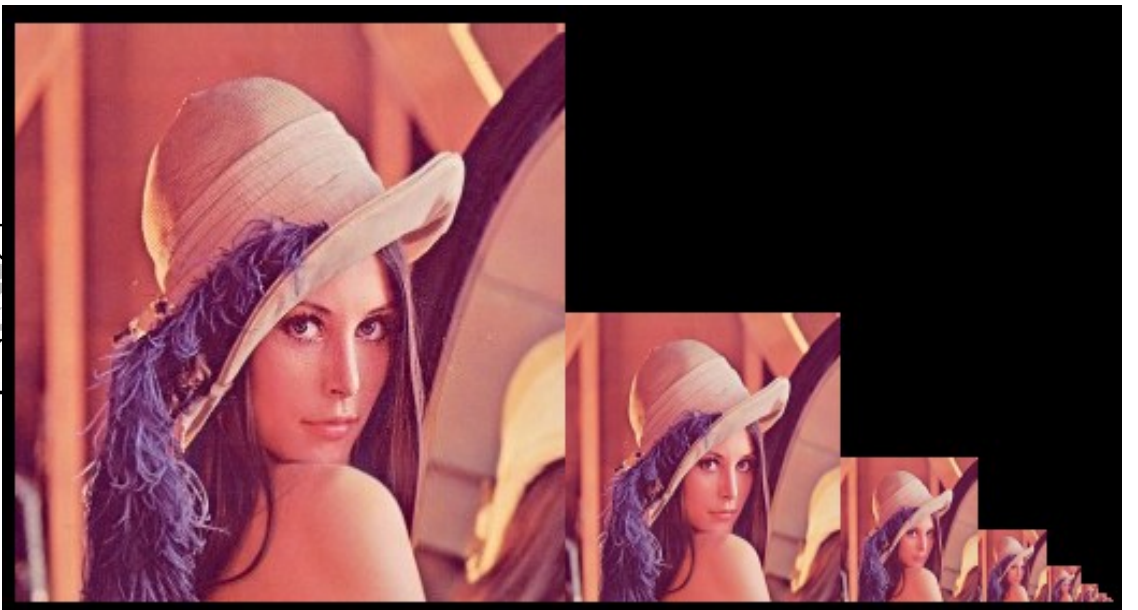
Nearest Linear Catmul-Rom CubicBSpline

Texture filtering (2)

- Mipmapping – minification
  - Down sampled textures to half
  - Memory requirements 133%
  - $d = \log_2(\mathrm{sqrt}(A))$
  - Trilinear filtering (interpolation)

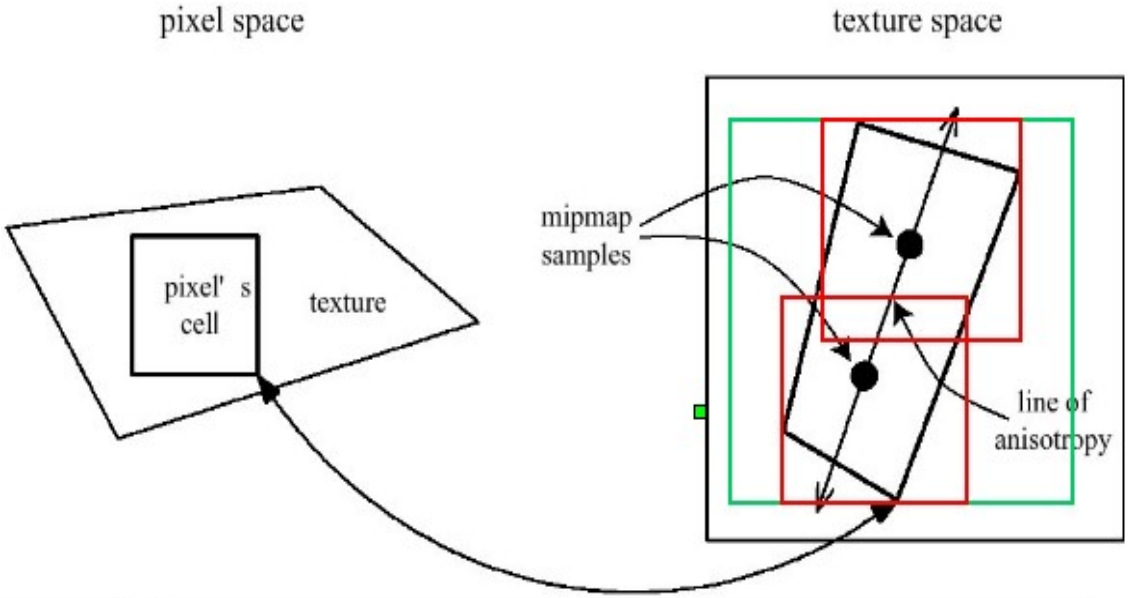


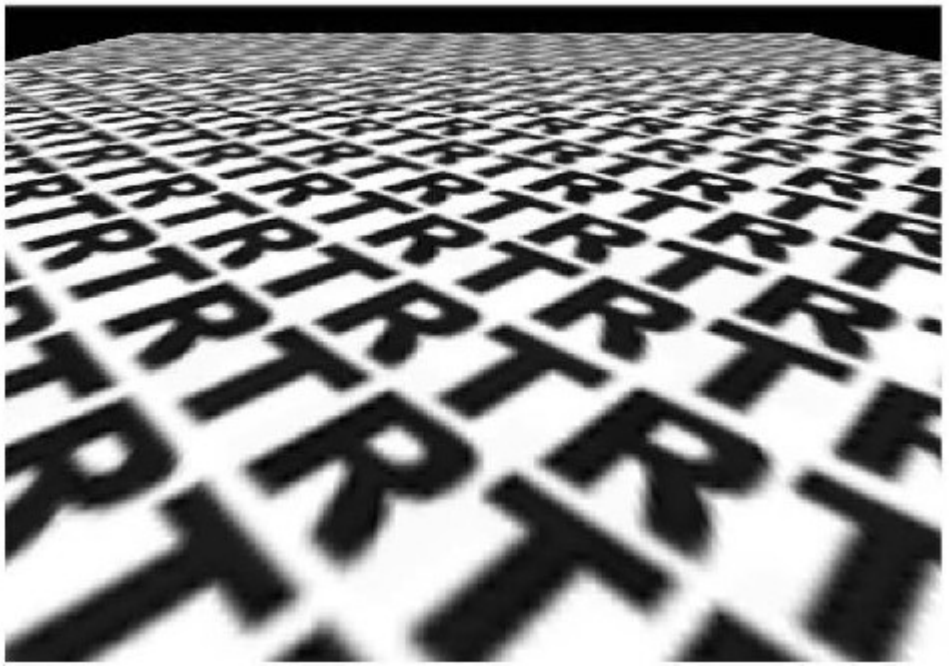


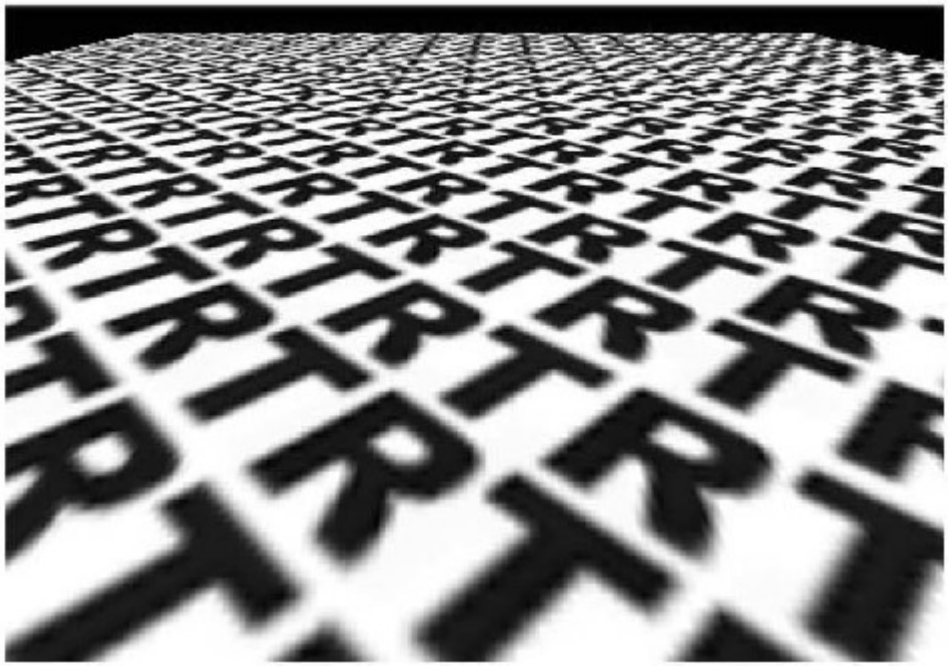


Texture filtering (3)

- Anisotropic
  - < 16 samples
  - Vertical, horizontal

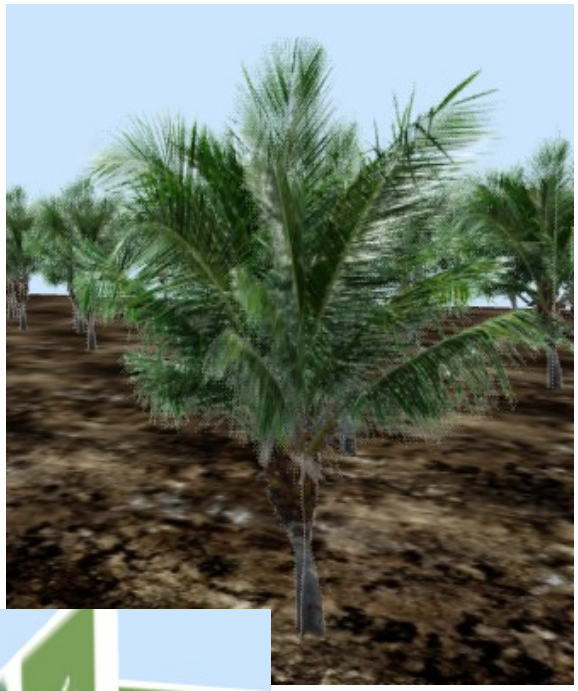






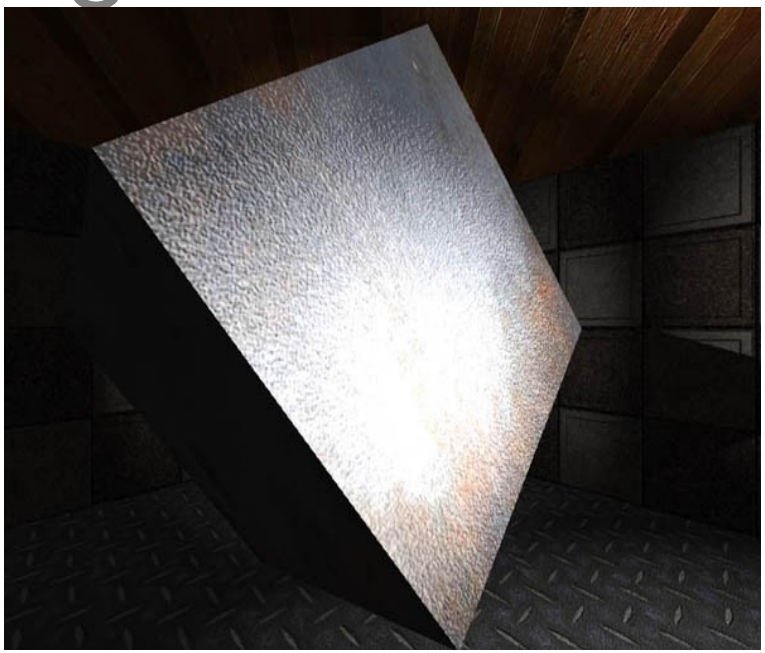
Alpha mapping

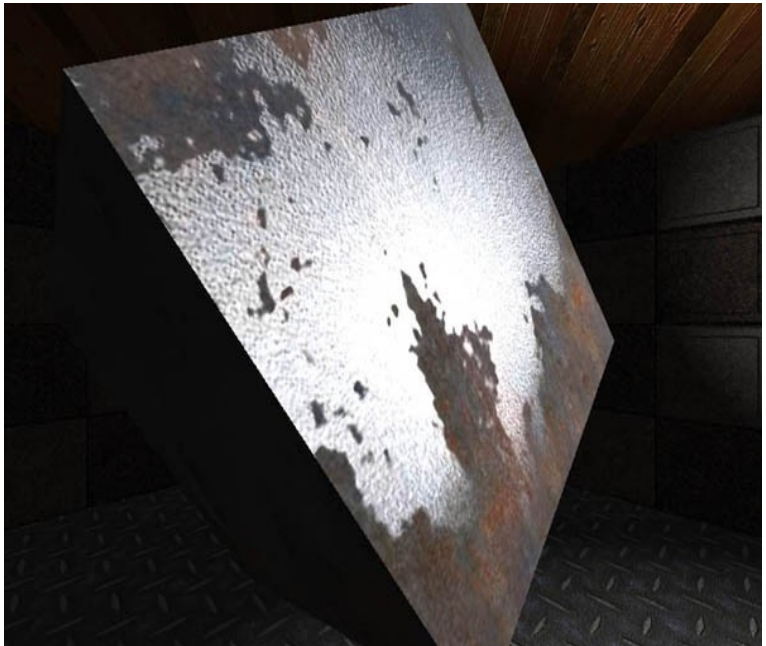
- Billboards
- Transp. Textures
  - trees,...
- Animated textures
  - fire, ...



Gloss mapping

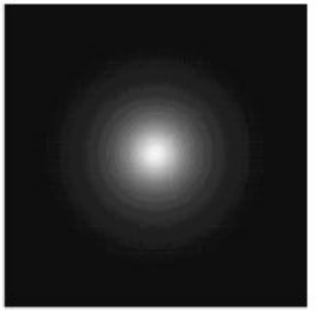
- Material properties in texture
- Used for specular

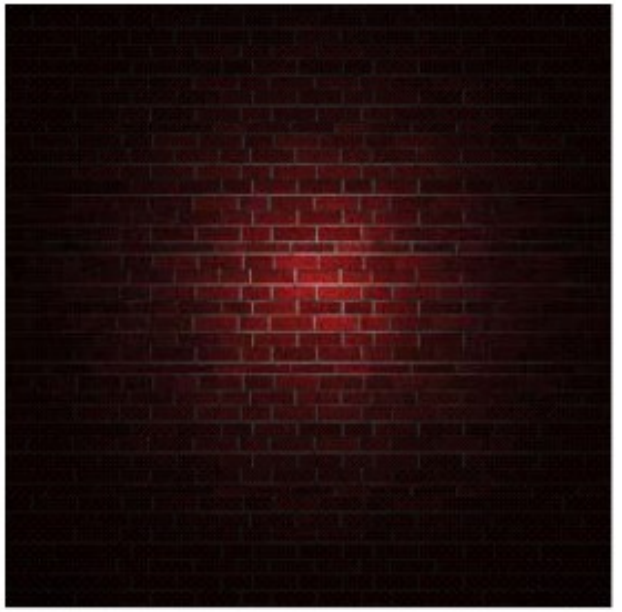




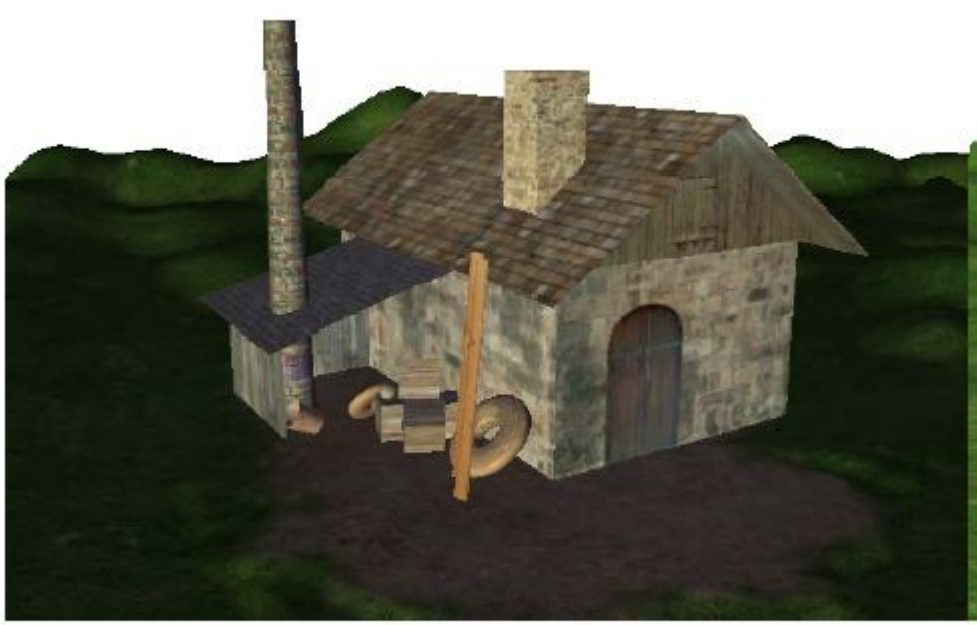
Light mapping

- Diffuse light view independent
- Precomputed light text.
  - Low resolution
  - Various approaches
- Combine with surface
- Shadows (static)
- Animated maps





Light mapping (2)

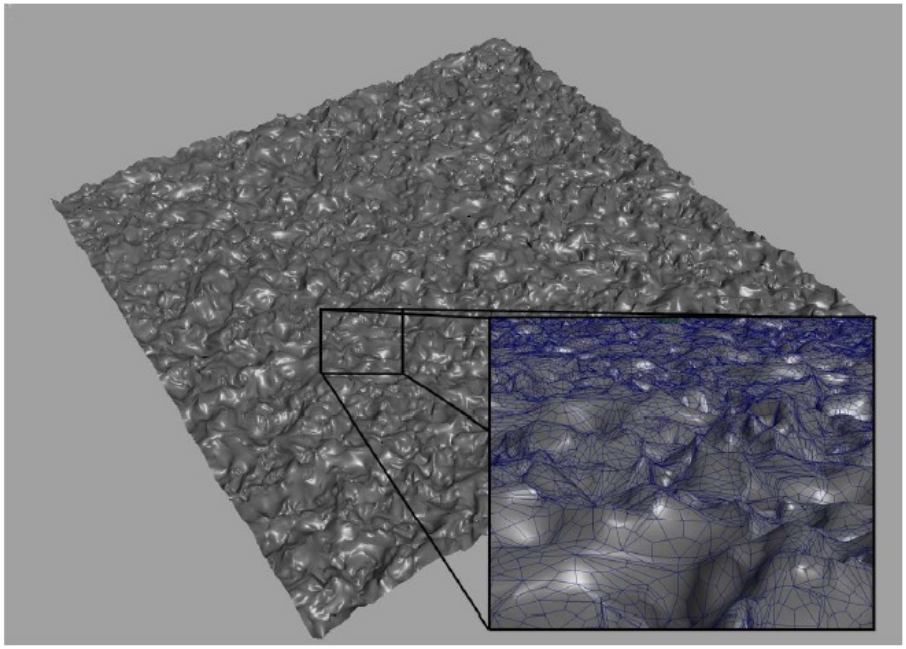


Original scene

Light-mapped

Rough surfaces

- ...
- Gloss Mapping
- Light Mapping
- Rough Surface
  - Bump mapping
  - Normal mapping
  - Displacement mapping
  - Parallax mapping
  - Relief mapping
  - Combinations
- Ambient Occlusion

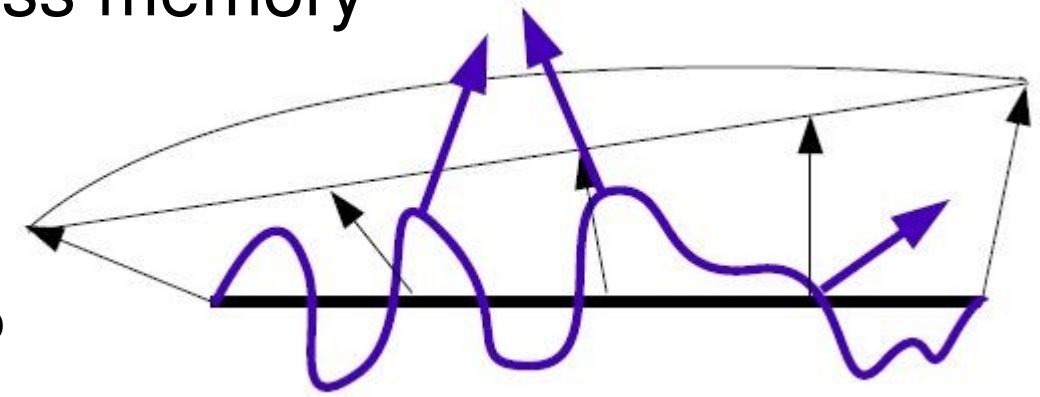


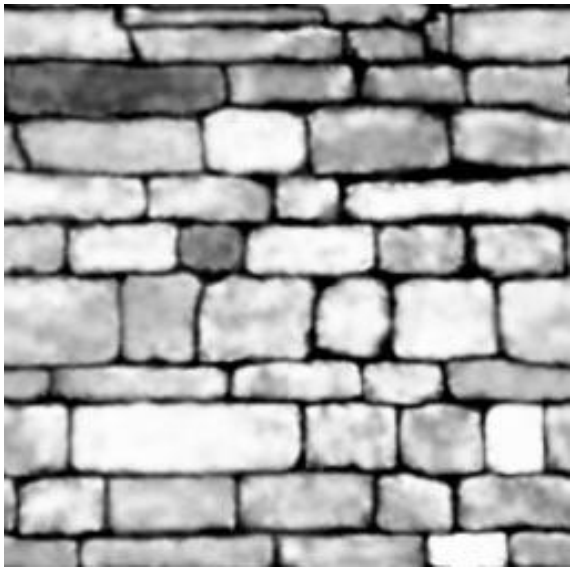
Bump mapping

- Rendering of structured surfaces
- Visual improvement without additional geometry
- Silhuette without change
- No shadows

Bump mapping (2)

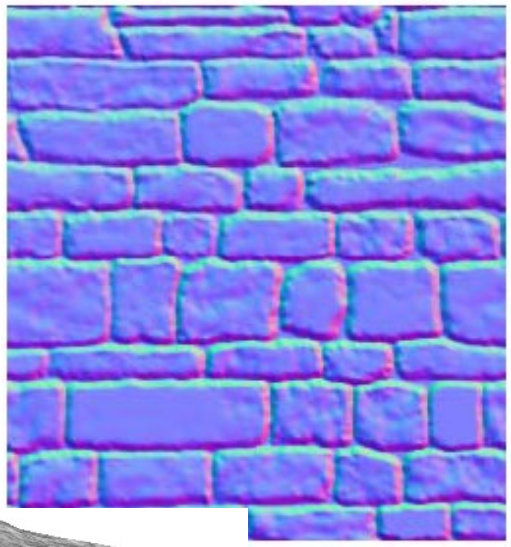
- Height map
  - Calculated normal
  - Central differences
  - uv space
- Perturbing surface normal by height map normal
- More computation
- Less memory



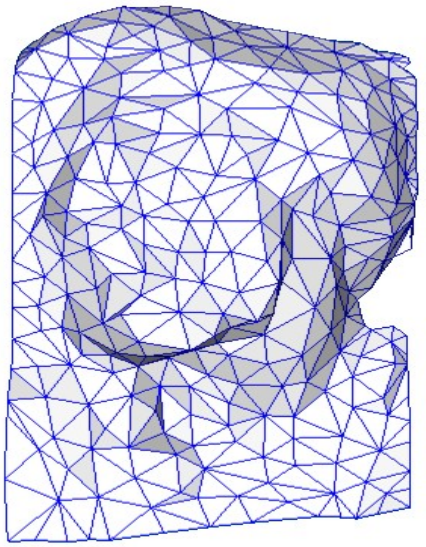


Normal mapping

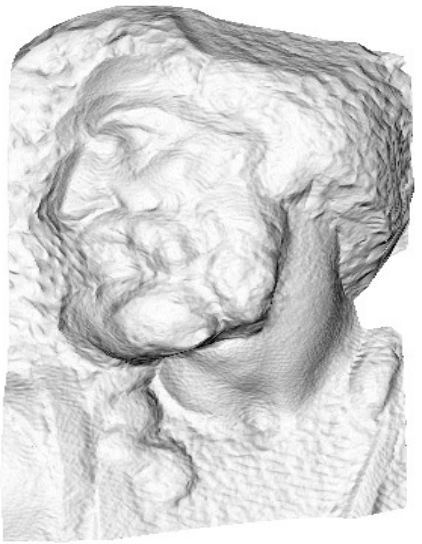
- Replaces surface normals from texture
- Normals in texture
  - Generated from high res. model
  - High resolution
  - Compressed [-1,1] → [0,1]
  - uv space



original mesh
4M triangles



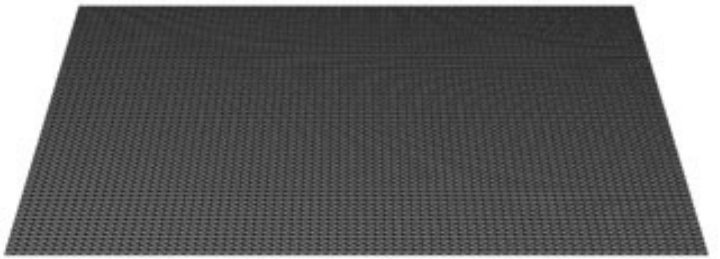
simplified mesh
500 triangles



simplified mesh
and normal mapping
500 triangles

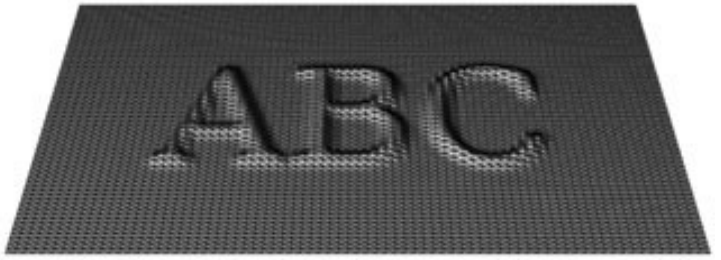
Displacement mapping

- Displacement along surface normal
- Applied to vertexes
  - Adaptive tessellation
- Costly technique
- New GPU (DX11)
  - Tesselator shader
  - Automatic LOD



ORIGINAL MESH

DISPLACEMENT MAP



MESH WITH DISPLACEMENT

Parallax mapping

- Approximation of original geometry
- More apparent depth
- Trace rays against height fields

Normal

Paralax

Relief

Parallax mapping

- Calculate offset
  - s,b (scale, bias) based on material
  - s=00.1, b=s*(-0.5)
  - $h_n=h*s-b$
  - $T_n = T_0+(h_n*V_{xy}/V_z)$
  - $V_z \rightarrow 0$
- Limited offset
- $T_n = T_0+(h_n*V_{xy})$

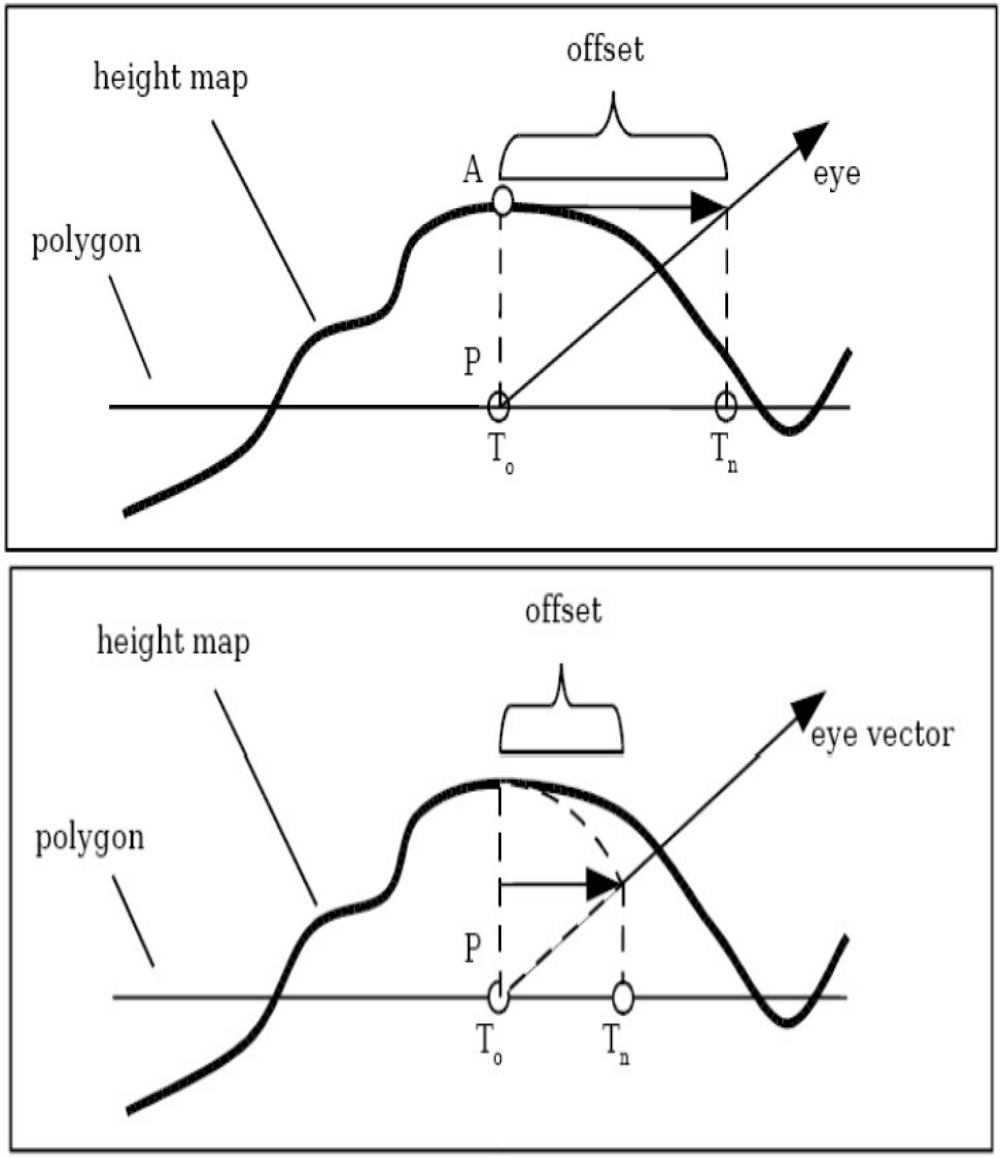
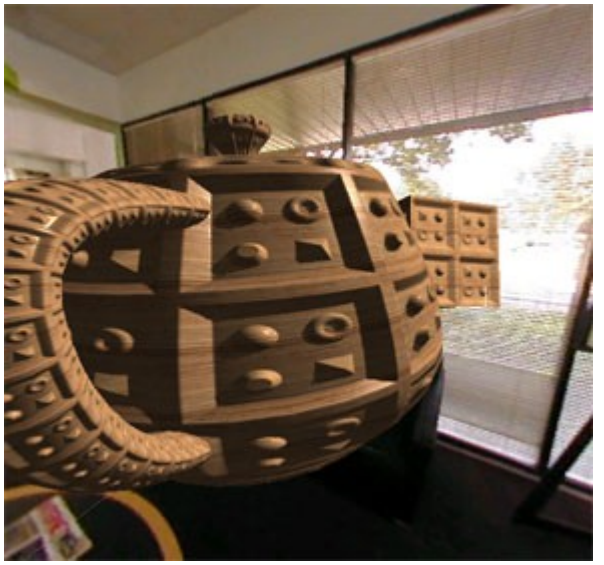
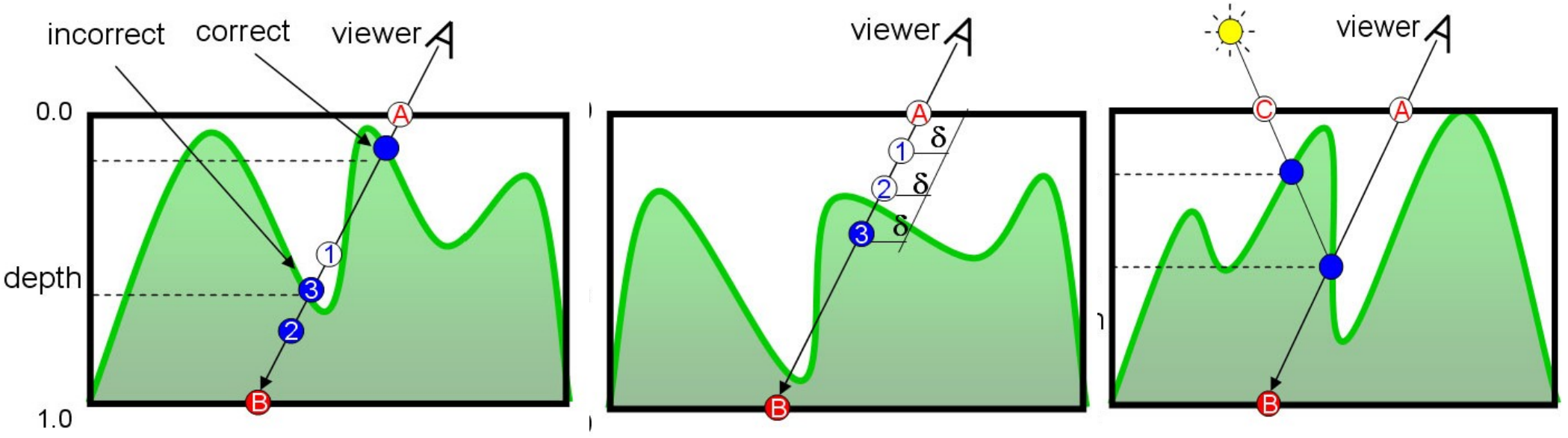


Image by Terry Welsh

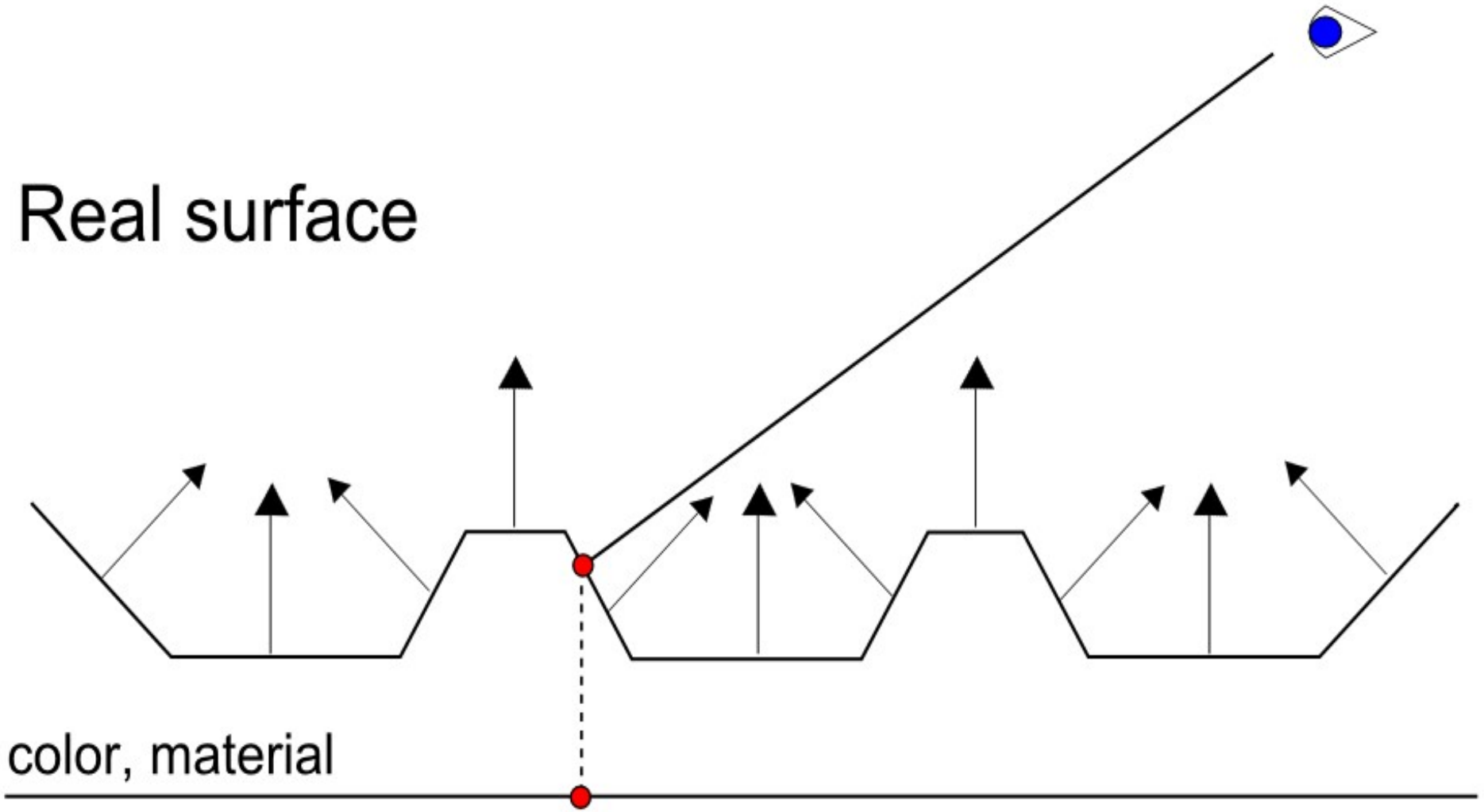
Relief mapping

- More accurate than parallax m.
- Self-shadowing, self-occlusion, silhouettes
- Based on ray-tracing
  - Various speed-up techniques

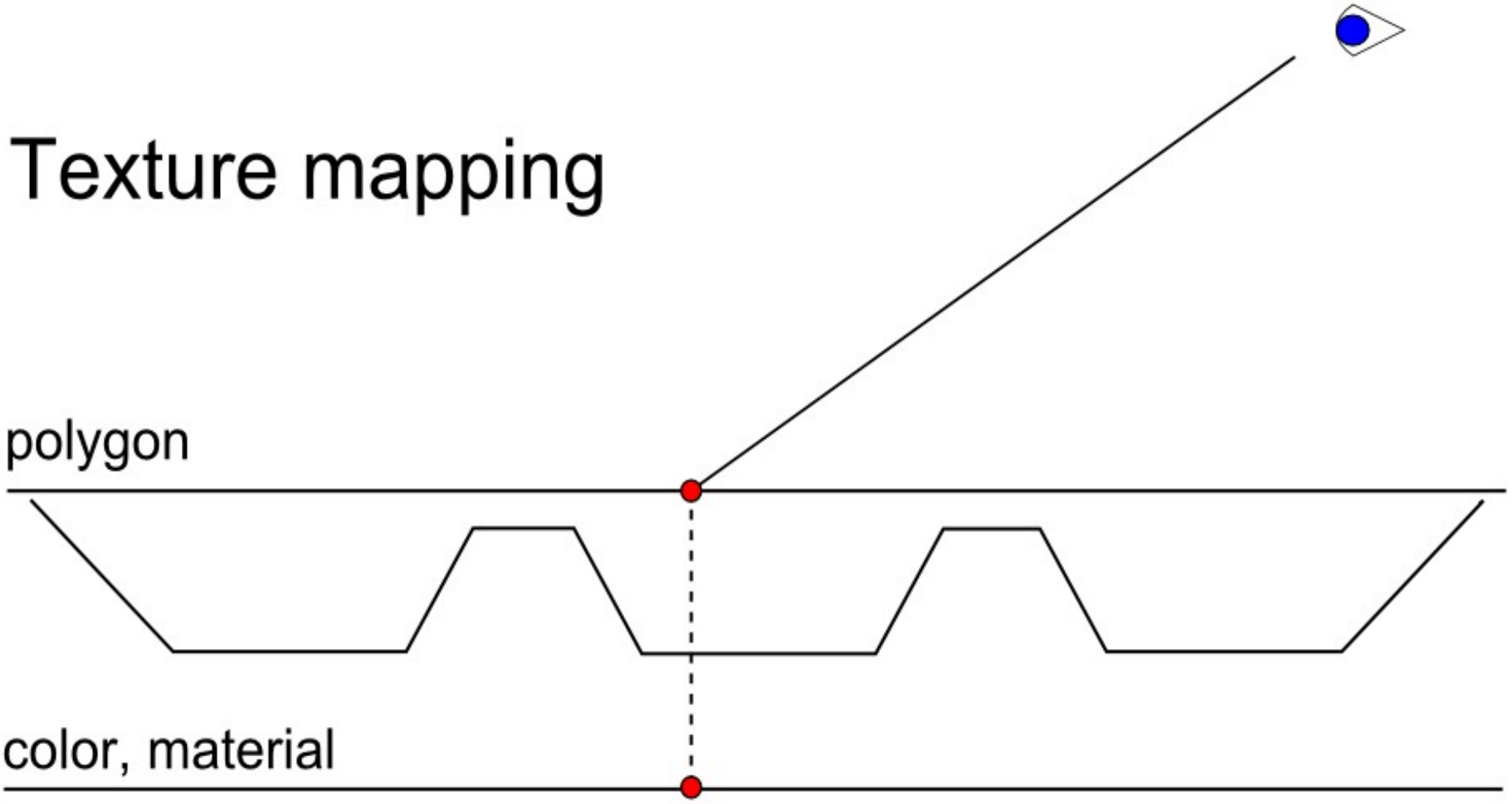




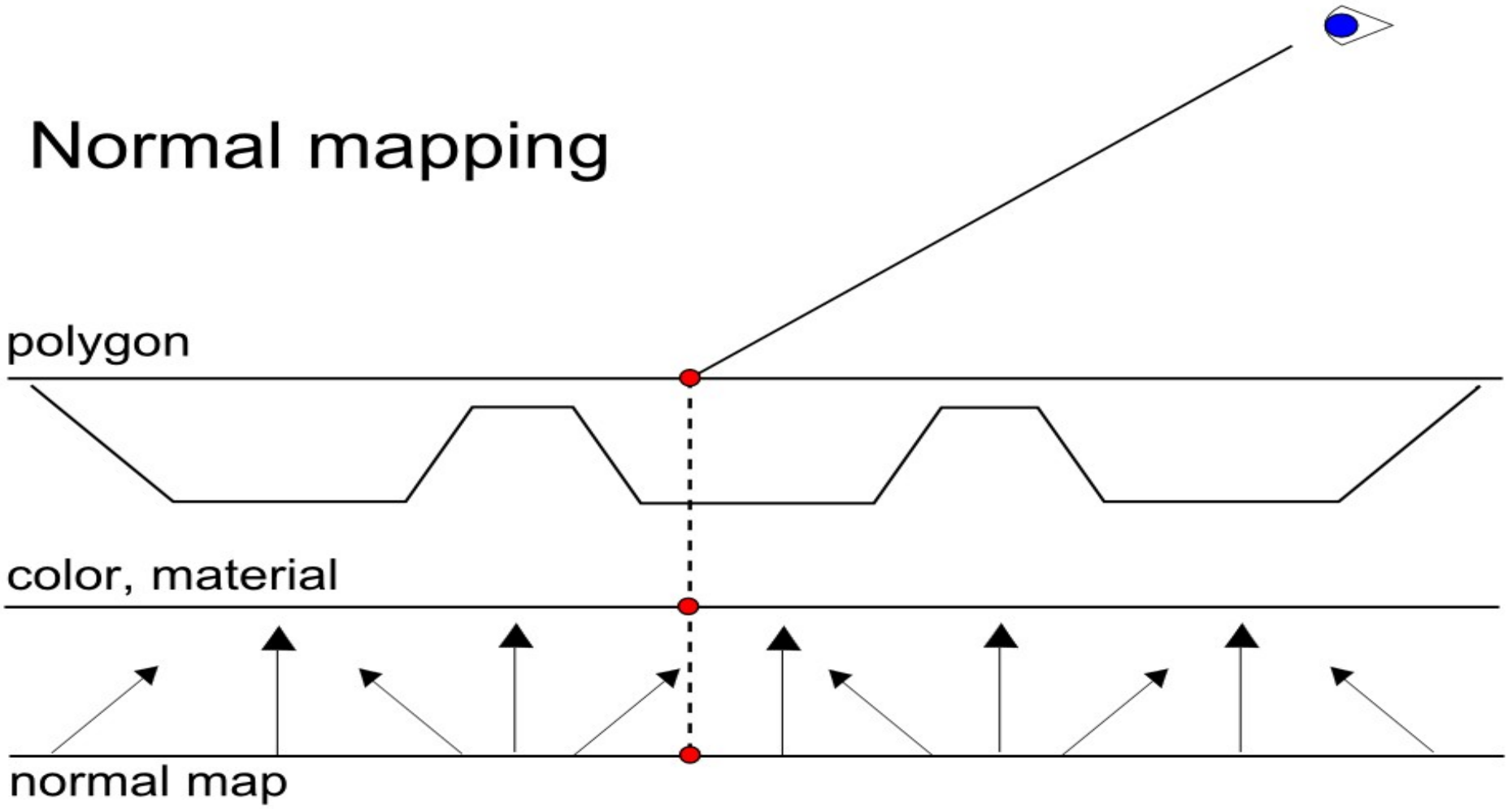
Comparison



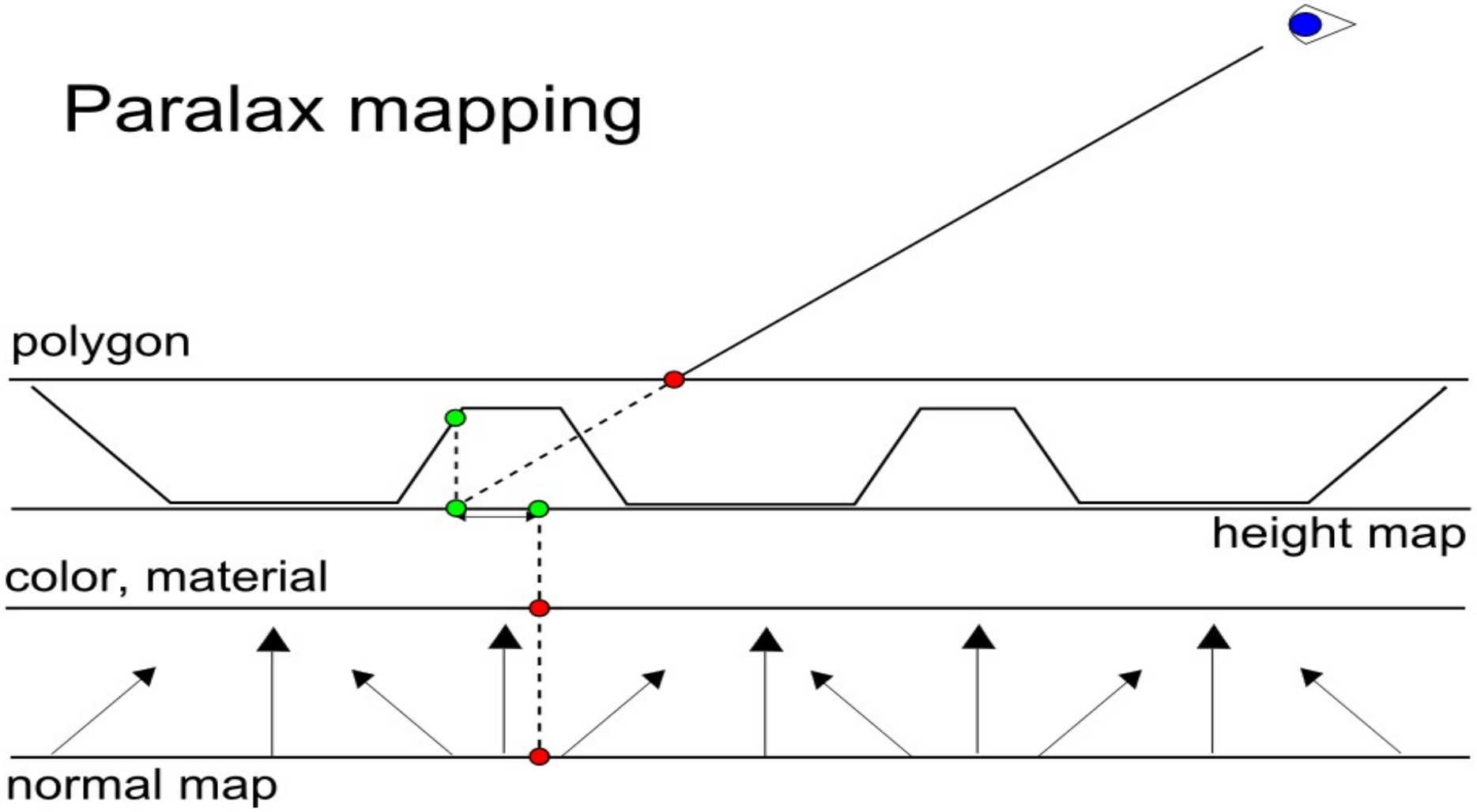
Comparison



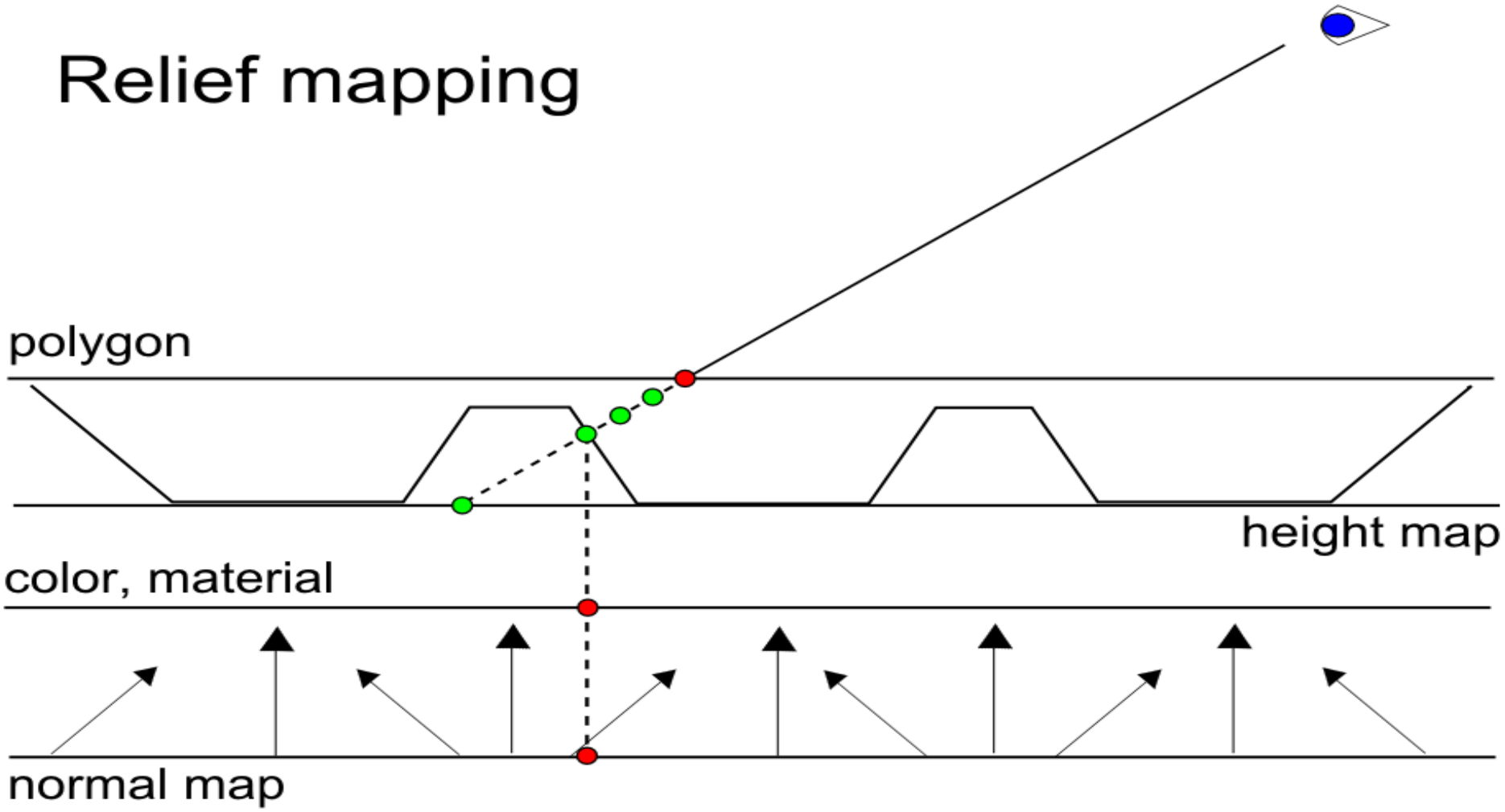
Comparison



Comparison


Paralax mapping
polygon
height map
color, material
normal map

Comparison


Relief mapping
polygon
height map
color, material
normal map

Comparison (2)

Texture mapping

Parallax mapping

Relief mapping

Combinations

- Steep Parallax Mapping
- Interval Mapping
- …

Texture Mapped

Normal Mapped

Parallax Mapped

Steep Parallax Mapped

References

- Real-time rendering slides
- GeForce 8 and 9 Series GPU Programming
- http://en.wikipedia.org/wiki/
- http://flickr.com/photos/buou/2126755136/
- http://vr.kaist.ac.kr/_r011.htm
- http://www.vis.uni-stuttgart.de/eng/teaching/lecture/ws04/seminar_spiele/bsp/
- http://www.cg.tuwien.ac.at/courses/Realtime/
- http://graphics.cs.ucf.edu/IntervalMapping/
- http://graphics.cs.brown.edu/games/SteepParallax/index.html
- http://www.inf.ufrgs.br/~oliveira/RTM.html
- http://ati.amd.com/developer/SIGGRAPH05/Tatarchuk-ParallaxOcclusionMapping-Sketch-print.pdf

Questions ?