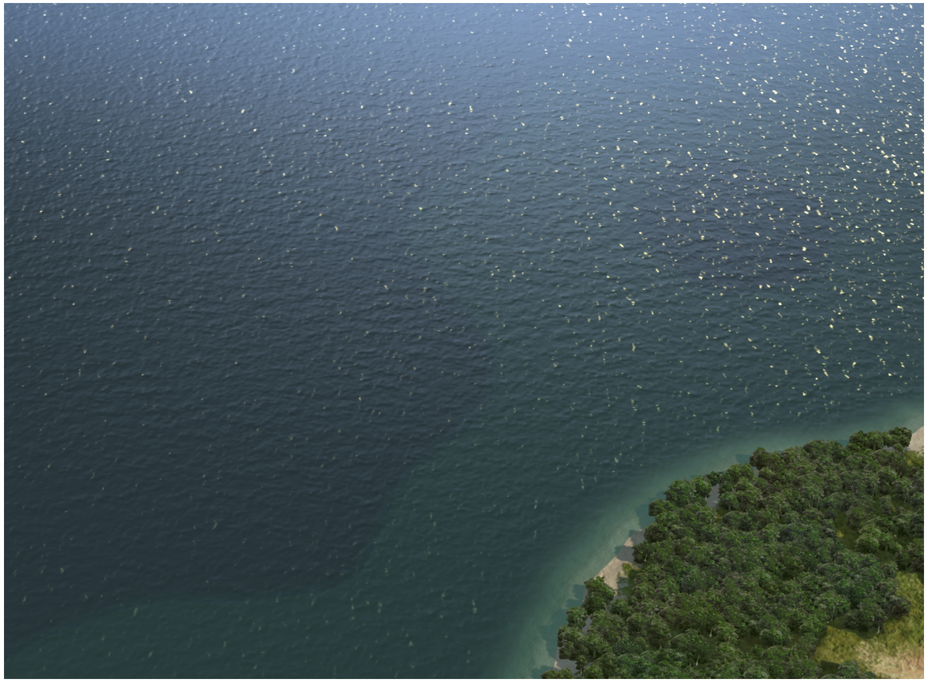
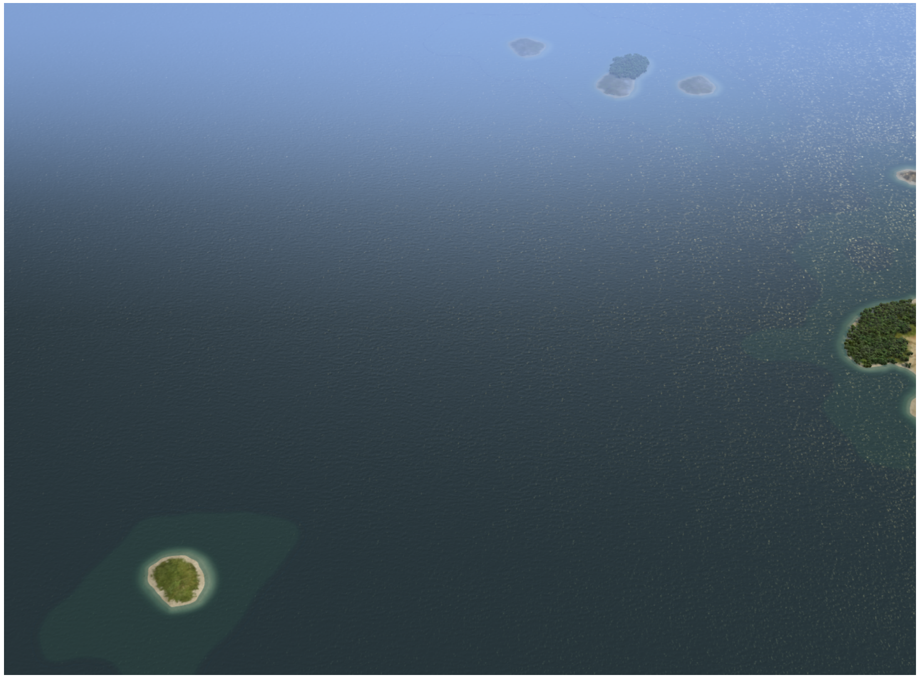
LEAN Mapping

Marc Olano & Dan Baker
Firaxis Games

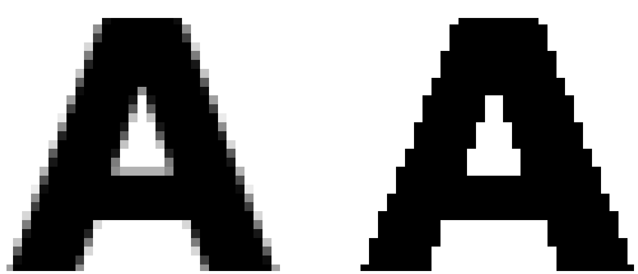


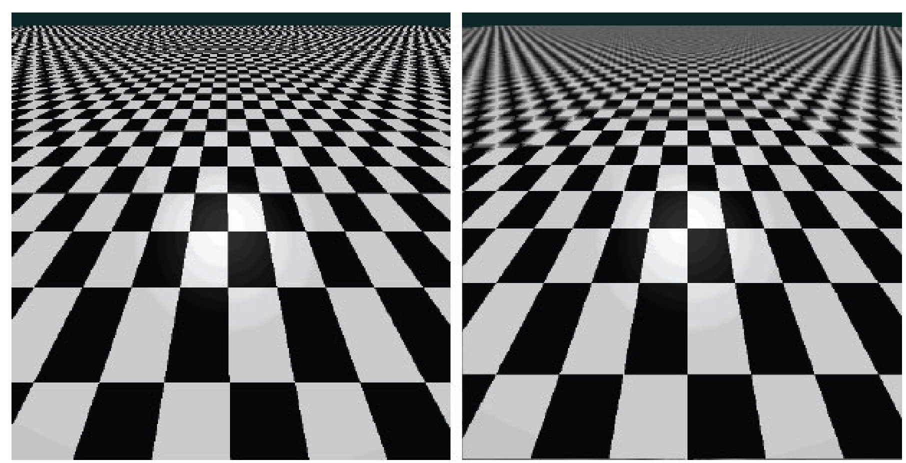


Presented by
Marcus Antonsson & Mattias Johnsson

Aliasing

- Spatial aliasing
- Prevalent for textures
- Solution is filtering techniques

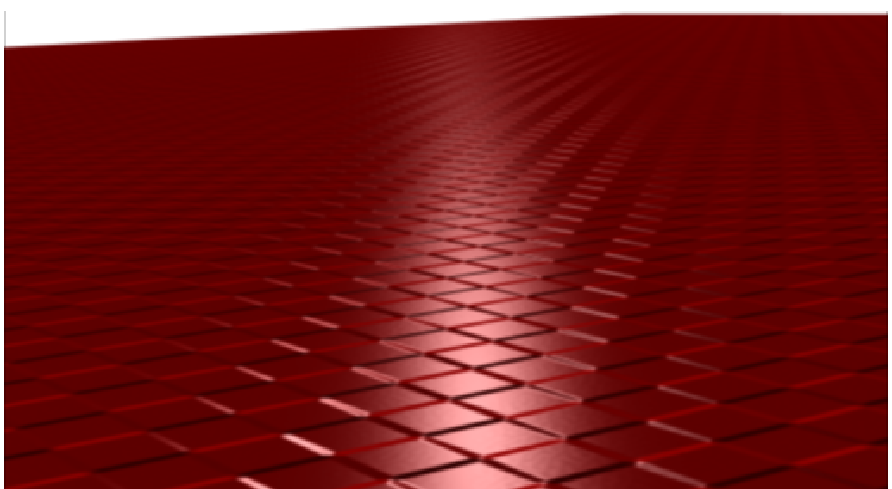


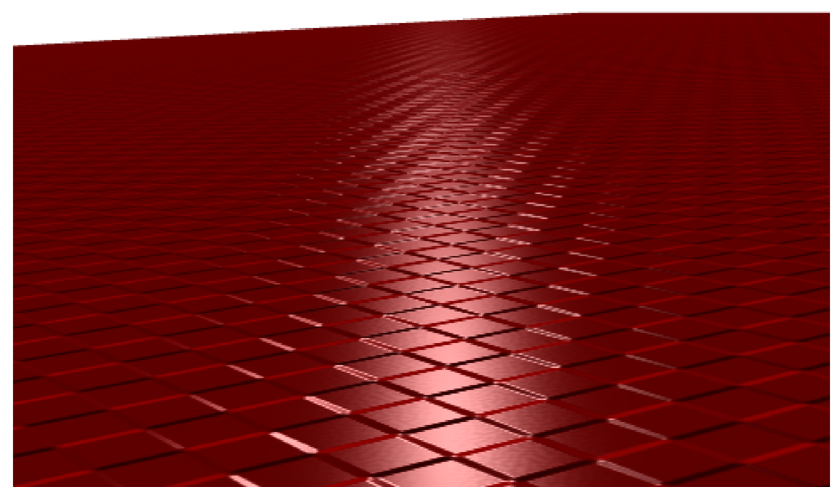


aliasing effects   anti-aliasing by over-sampling

Aliasing in bump mapping

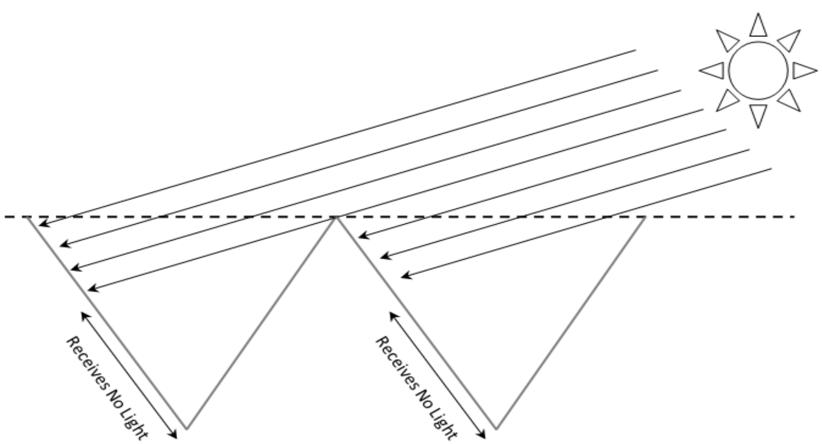
- Usual solutions do not work with MIP maps
- Diffuse lighting OK
- Cause aliasing for specular lighting
  - Bump normals appear correct
  - Shininess should appear more dull

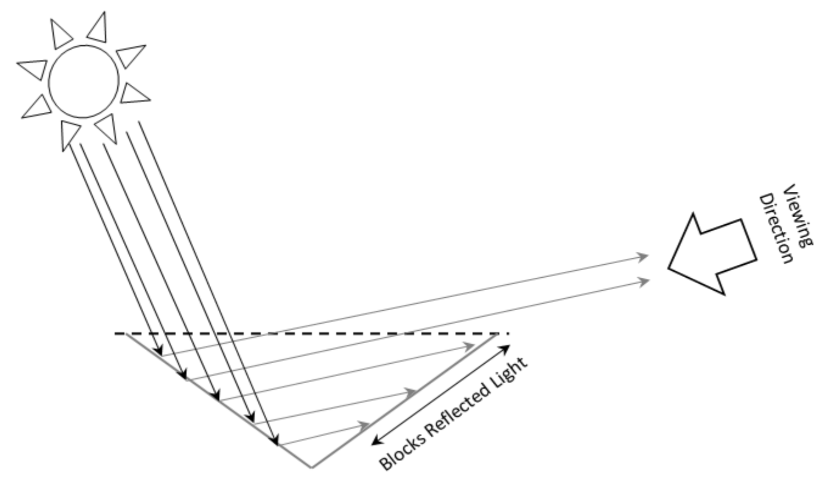




Shading model

- Uses modified Ward shading model
  - Based on Beckman Distribution
  - Added fresnel term
  - Cook-Torrance without shadowing & masking



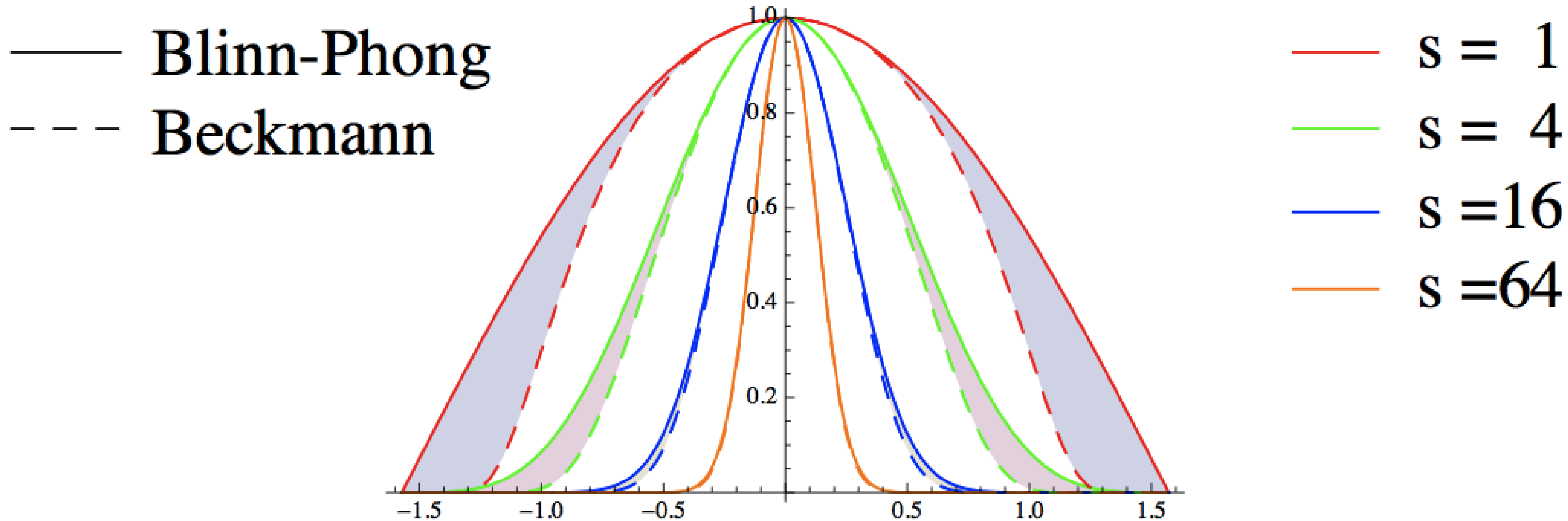


Outline of solution

- Blinn-Phong/Beckmann equivalence
- Surface Beckmann
- MIP techniques
  - Linearity of mip techniques
  - Equivalence
- Adding the roughness term
- Combining several maps

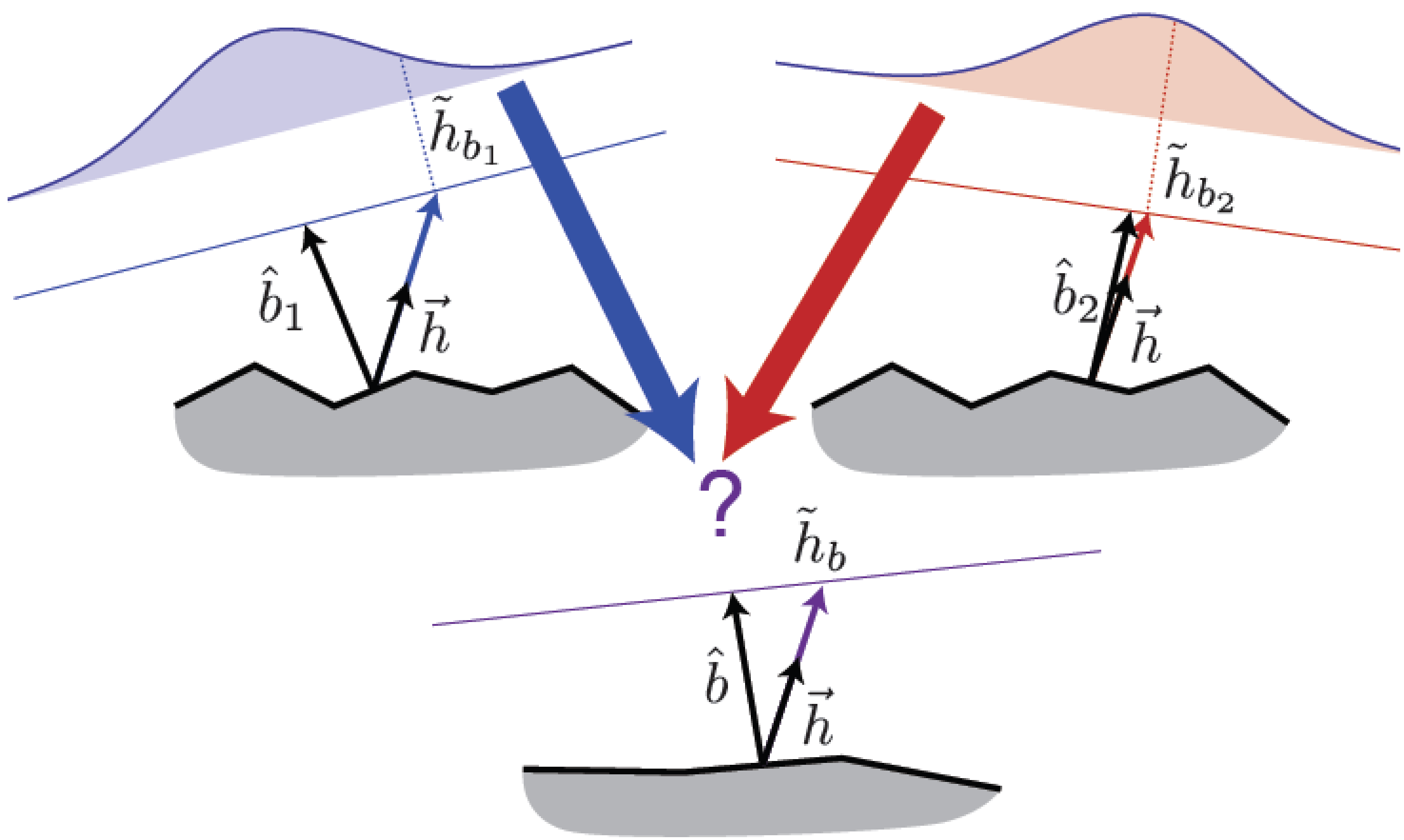
Blinn-Phong/Beckmann equivalence

- Blinn-Phong approaches Beckman distribution with variance 1/s as s increases
- Technique can be used with standard Blinn-Phong



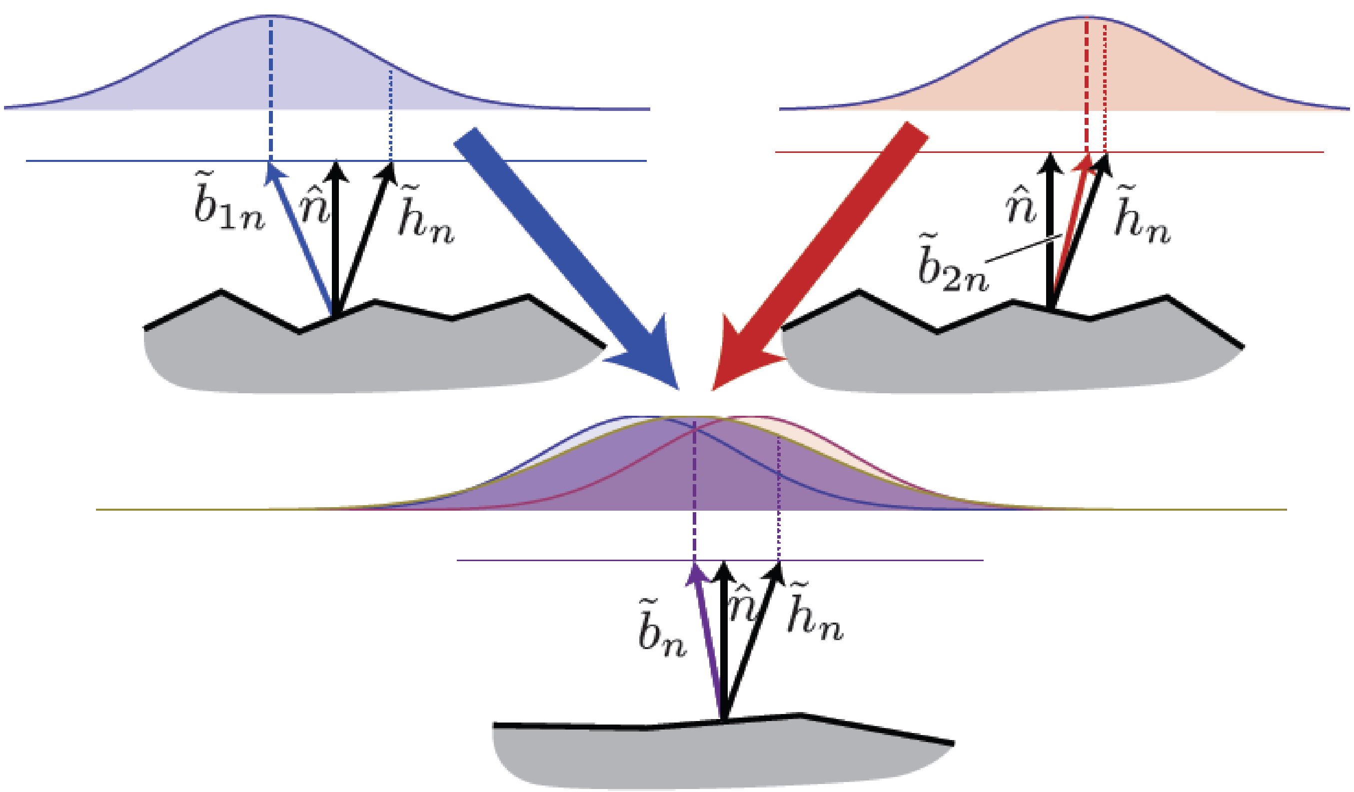
Surface Beckmann

- Standard Beckmann projects to the microfacet normal plane



Surface Beckmann cont.

- Project to underlying surface (normal plane) instead



Beckmann distribution

- Standard Beckmann:

$$\frac{1}{\sqrt{2\pi}|\Sigma|} e^{-\frac{1}{2}\tilde{h}_b^T \Sigma^{-1} \tilde{h}_b}$$

- They use:

$$\frac{1}{\sqrt{2\pi}|\Sigma|} e^{-\frac{1}{2}(\tilde{h}_n - \tilde{b}_n)^T \Sigma^{-1} (\tilde{h}_n - \tilde{b}_n)}$$

- Where

Σ = 2D covariance matrix

The LEAN Map

- The matrix is given by

$$\Sigma = \begin{bmatrix} (\frac{1}{n}\sum \tilde{b}.x^2) - \bar{b}.x^2 & (\frac{1}{n}\sum \tilde{b}.x\,\tilde{b}.y) - \bar{b}.x\,\bar{b}.y \\ (\frac{1}{n}\sum \tilde{b}.x\,\tilde{b}.y) - \bar{b}.x\,\bar{b}.y & (\frac{1}{n}\sum \tilde{b}.y^2) - \bar{b}.y^2 \end{bmatrix}$$

- Does not combine linearly
  - But the individual terms do

$$N = (\vec{b}_n.x,\ \vec{b}_n.y,\ \vec{b}_n.z)$$

$$B = (\tilde{b}_n.x,\ \tilde{b}_n.y)$$

$$M = (\tilde{b}_n.x^2,\ \tilde{b}_n.x\,\tilde{b}_n.y,\ \tilde{b}_n.y^2)$$

Restoring Σ

$$N = (\vec{b}_n.x,\ \vec{b}_n.y,\ \vec{b}_n.z)$$

$$B = (\tilde{b}_n.x,\ \tilde{b}_n.y)$$

$$M = (\tilde{b}_n.x^2,\ \tilde{b}_n.x\,\tilde{b}_n.y,\ \tilde{b}_n.y^2)$$

$$\Sigma = \begin{bmatrix} M.x - B.x * B.x & M.y - B.x * B.y \\ M.y - B.x * B.y & M.z - B.y * B.y \end{bmatrix}$$

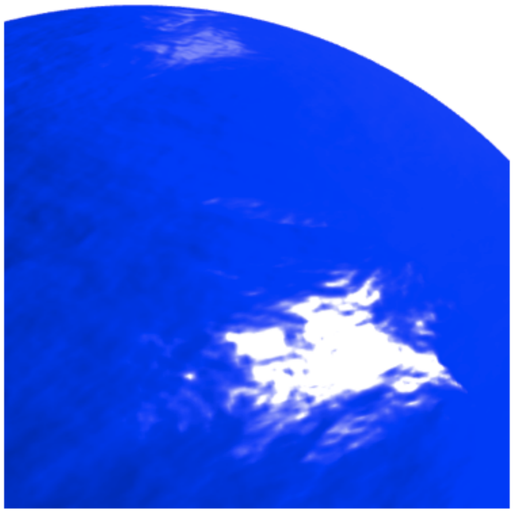
Adding the roughness term

- Perfectly reflective microfacets assumed
- 1/s can be added directly to the maps

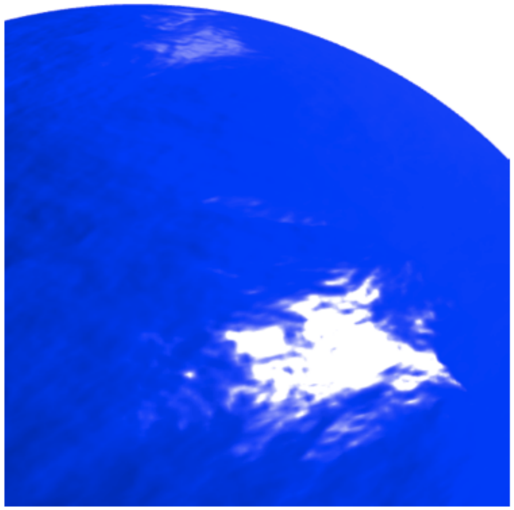
$$M = (\tilde{b}_n.x^2 + 1/s,\ \tilde{b}_n.x\,\tilde{b}_n.y,\ \tilde{b}_n.y^2 + 1/s)$$

- Alternately we can add 1/s during shading.
  - This is what they do in their implementation

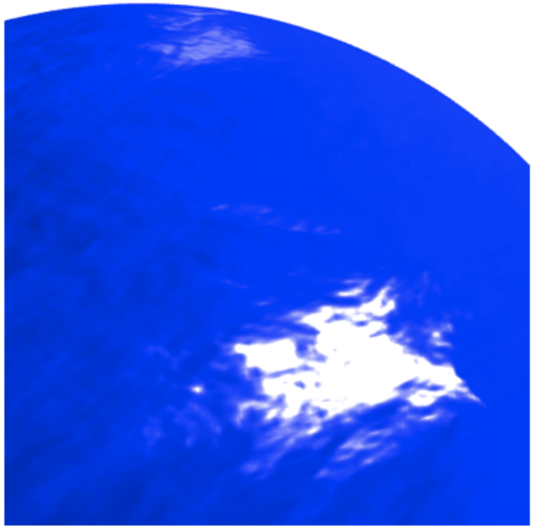
Everything is equivalent!



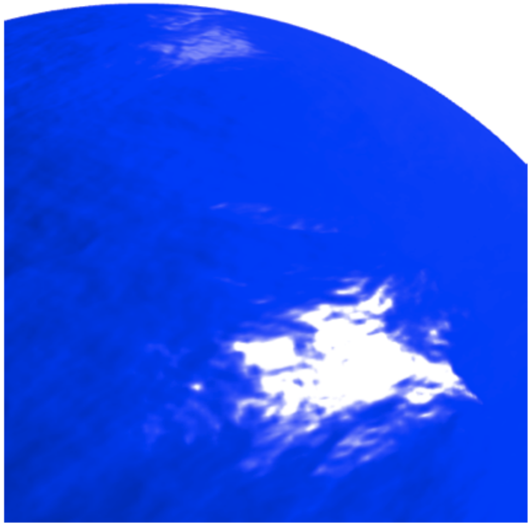
Blinn-Phong applied to bump normal



Beckmann in the bump tangent frame



Beckmann in the surface tangent frame with bumps as off-center distributions



LEAN mapping

Combining layers

- Several normal maps on the same surface
  - Good for waves, streams etc.
- Three methods explored
  - Combine at texture generation time
  - Generate mixture textures
  - Approximation

Combine at texture generation time

- Advantages
  - Least run-time overhead
  - No artifacts
- Disadvantages
  - Requires generation of textures at run-time
  - Or static textures

Mixture textures

- Given a height field f(x, y), normals can be computed

$$\widetilde{bf} = \left(-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}\right)$$

- Linear combination of two height fields

$$\begin{aligned}\tilde{b} &= \left(-\frac{\partial(tf+ug)}{\partial x}, -\frac{\partial(tf+ug)}{\partial y}\right)\\ &= t\,\widetilde{bf} + u\,\widetilde{bg}\end{aligned}$$

t,u > 0, t + u = 1

Mixture textures cont.

- Blue terms already saved in LEAN map
- Red terms need to be stored
- All terms are independent of t & u

$$
\begin{aligned}
\tilde{b}.x^2 &= (t\,\widetilde{bf}.x + u\,\widetilde{bg}.x)^2 \\
&= t^2(\widetilde{bf}.x)^2 + u^2(\widetilde{bg}.x)^2 + 2\,t\,u\,(\widetilde{bf}.x\,\widetilde{bg}.x) \\
\tilde{b}.y^2 &= (t\,\widetilde{bf}.y + u\,\widetilde{bg}.y)^2 \\
&= t^2(\widetilde{bf}.y)^2 + u^2(\widetilde{bg}.y)^2 + 2\,t\,u\,(\widetilde{bf}.y\,\widetilde{bg}.y) \\
\tilde{b}.x\,\tilde{b}.y &= (t\,\widetilde{bf}.x + u\,\widetilde{bg}.x)(t\,\widetilde{bf}.y + u\,\widetilde{bg}.y) \\
&= t^2\,(\widetilde{bf}.x\,\widetilde{bf}.y) + u^2\,(\widetilde{bg}.x\,\widetilde{bg}.y) \\
&\quad + t\,u\,(\widetilde{bf}.x\,\widetilde{bg}.y) + t\,u\,(\widetilde{bg}.x\,\widetilde{bf}.y),
\end{aligned}
$$

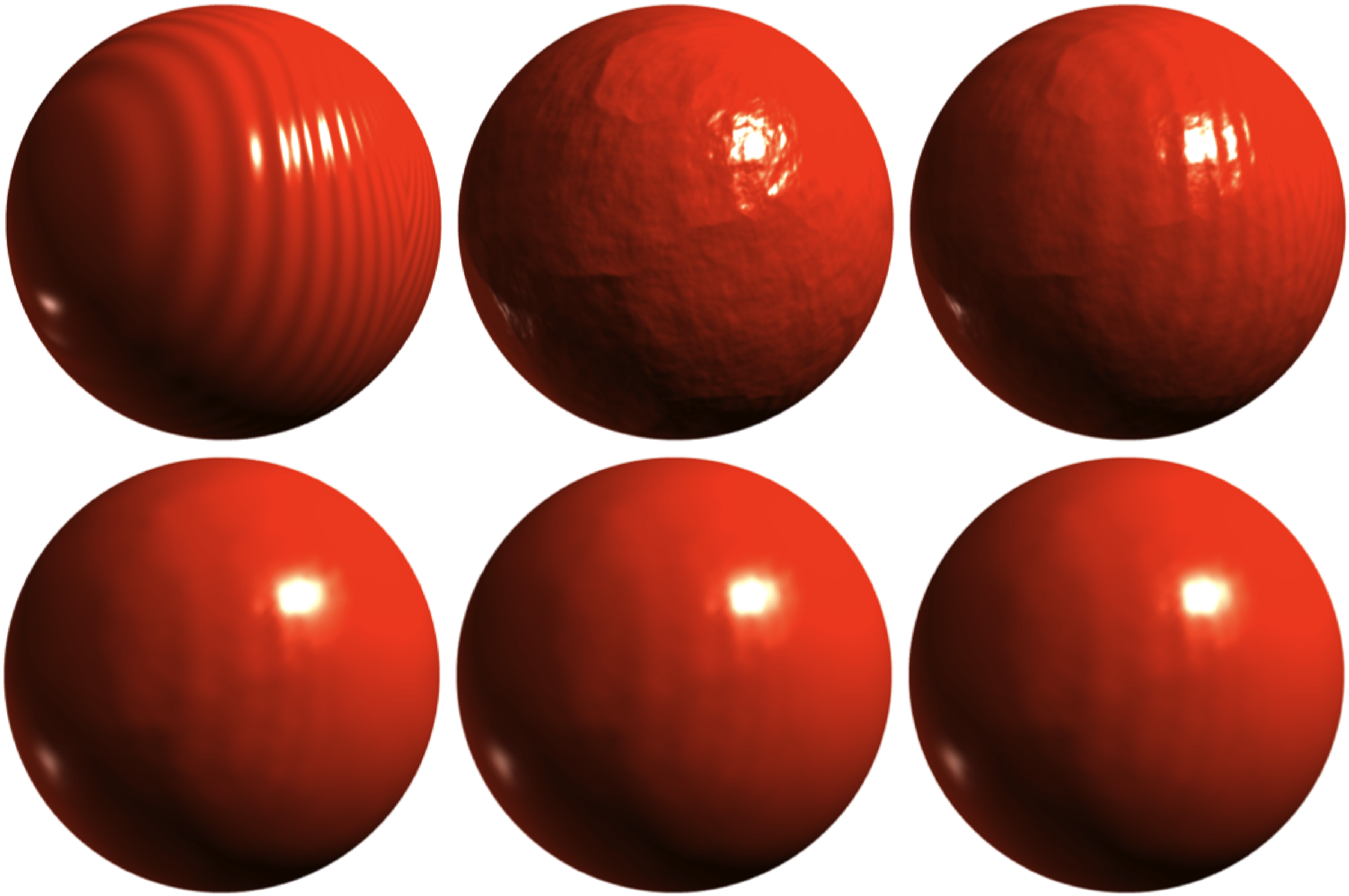
Mixture textures cont.

- Advantages
  - Can change relation of bump maps
  - No artifacts
- Disadvantages
  - If layers move relatively to each other the textures need to be recomputed
  - Requires additional memory
    - One additional texture per pair of bump maps

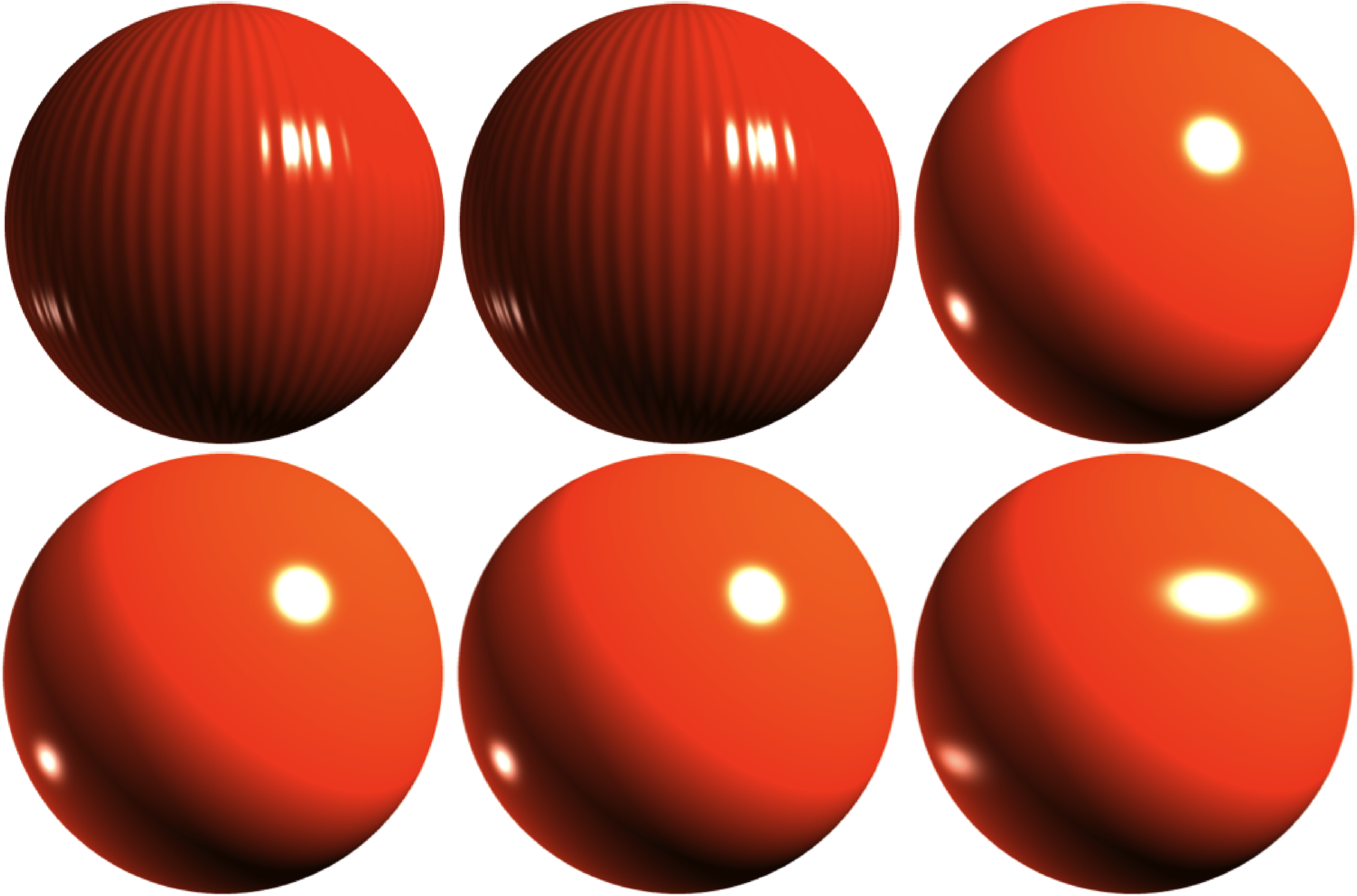
Approximation

- Bump combination can be approximated at run time
  - This is not covered in detail
- Advantages
  - Requires no additional memory
  - Requires no recomputation of textures
- Disadvantages
  - Artifacts when layers are coherent

Two layers, no coherency



Two layers with coherency



Performance

Blinn-Phong	Single LEAN map
1570 FPS 30 ALU + 1 tex	1540 FPS 42 ALU + 2 tex

Per-frame generation	Mixture textures	Mixture approx.
917 FPS gen: 8 ALU + 1 tex use: 42 ALU + 2 tex	1450 FPS 54 ALU + 5 tex	1458 FPS 54 ALU + 4 tex

Using ATI Radeon HD 5870

1600x1200 Full screen, all pixels covered

Limitations

- Can only handle one direction of anisotropy
- None of the options for combinations is ideal
  - The three methods together can handle most cases
- Needs 16 bits to store the shininess (s) to achieve sufficient precision
  - Ordinary bump mapping rarely above s = 64 because of aliasing
  - With lean maps 10000 – 20000 is used

Questions?

- TL;DR
  - Normal/bump maps don't combine linearly when MIP mapping
    - Results in spatial aliasing in specular highlights
  - LEAN maps combine linearly
    - Compatible with Blinn-Phong
    - Compatible with standart MIP mapping
    - Supplies with three methods for combining layers
    - Low overhead