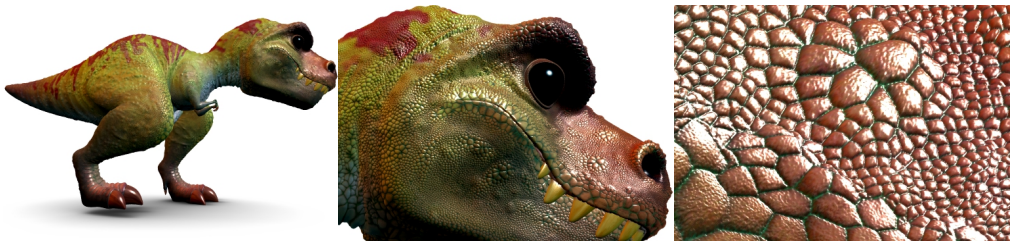
Antialiasing Physically Based Shading with LEADR Mapping



Part of:
Physically Based Shading in Theory and Practice
SIGGRAPH 2014 Course

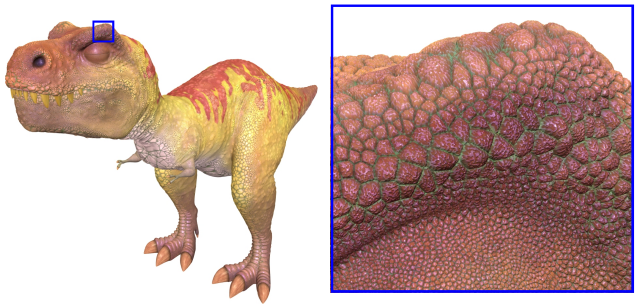
Jonathan Dupuy[1,2]

[1]Université Lyon 1, LIRIS, CNRS
[2]Université de Montréal, Dept. I.R.O., LIGUM

In this talk I would like to emphasize the importance of antialiasing in physically based rendering pipelines. The ideas developed here are based on the LEADR mapping paper I co-wrote with Eric and other colleagues.

Meet Tiny the T. Rex

- Combination of **displaced subdivision surfaces** and **physically based shading**
- Achieves **very high-resolution models**, with **low storage costs**



Tiny the T. Rex
(© Disney)

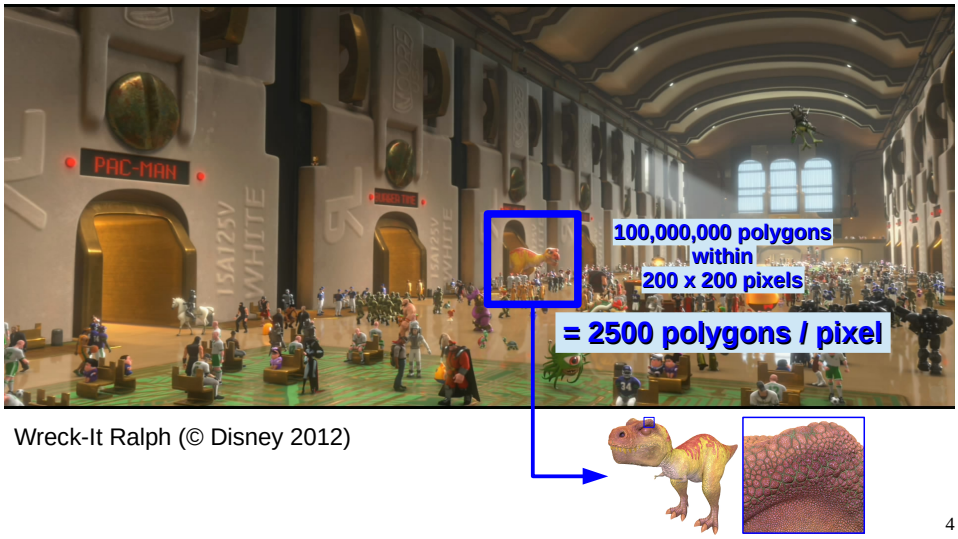
≈ 100,000,000 polygons
for 4 gigabytes

≈ 42 bytes per polygon

2

To start things off, I would like to introduce you to a 3D model we used in the paper. His name is Tiny the T. rex; he is composed of displaced subdivision surfaces and relies heavily on physically based shading. This representation is very common in offline rendering today, because it is very good at representing detail in a compact way. For instance, Tiny weighs a little more than 4 Gigabytes on disk, and has a resolution of roughly 100,000,000 polygons.

A Challenge in Rendering



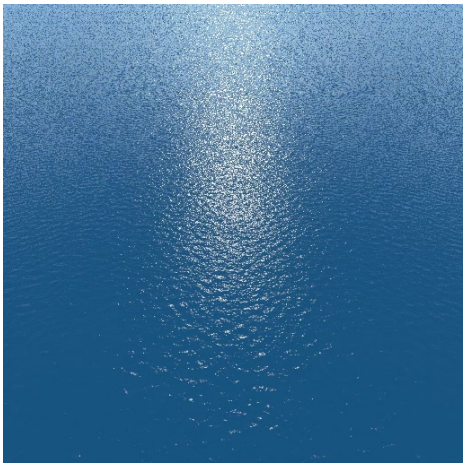
Wreck-It Ralph (© Disney 2012)

4

Here is a still frame from Disney's movie “Wreck-It Ralph”. If you look closely enough, you may notice Tiny performing in the background.
To render Tiny in this image, over a million polygons have to be processed in roughly 200x200 pixels, resulting in roughly 2500 polygons and texels per pixel. This requires generating several thousands of samples per pixel just to integrate visibility effects (and then comes shading). In practice, production renders typically use far less samples for this problem. So do video games, which rarely exceed 16 samples per pixel (4 or less is common). Hence, virtual scenes are always undersampled.

Undersampling Misdeeds

Undersampling results in **noise** and/or **aliasing**



undersampled rendering

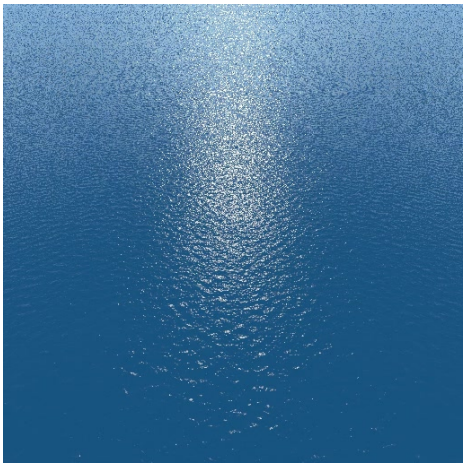
ground truth

5

Undersampling results in artifacts in the image. Note that at this point, the samples that constitute the pixels are still physically based, per se, so we have physically based noise/aliasing. These artifacts are unacceptable though, so images usually get post-processed. This is problematic for two reasons. First, it negates most of the benefits of using physically based shading models. Second, given how displays are evolving (better resolutions, faster refresh rates, etc.), it is almost certain that this process will eventually become a serious bottleneck for both movies and video games.
For the video sequence, see ocean.mp4.

Undersampling Misdeeds

Undersampling results in **noise** and/or **aliasing**

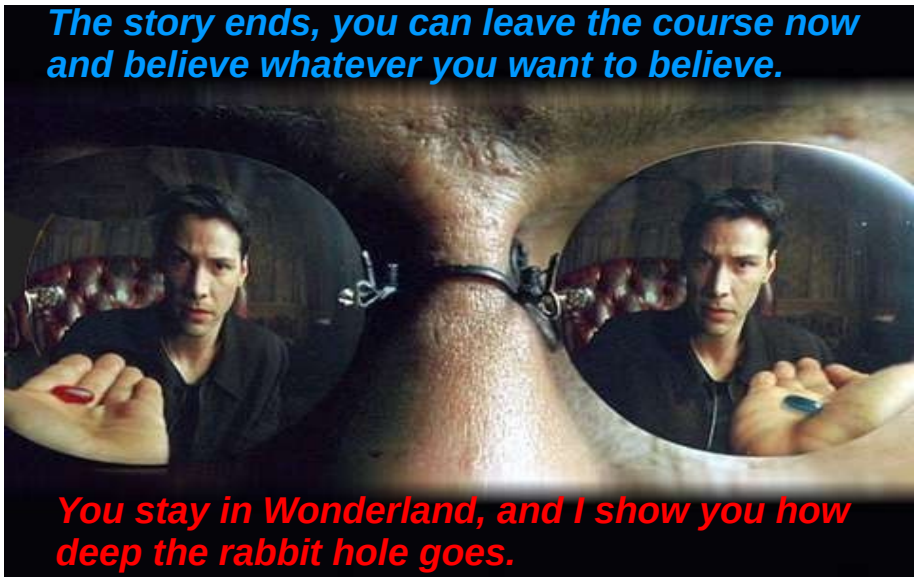


undersampled rendering

ground truth

Undersampling results in artifacts in the image. Note that at this point, the samples that constitute the pixels are still physically based, per se, so we have physically based noise/aliasing. These artifacts are unacceptable though, so images usually get post-processed. This is problematic for two reasons. First, it negates most of the benefits of using physically based shading models. Second, given how displays are evolving (better resolutions, faster refresh rates, etc.), it is almost certain that this process will eventually become a serious bottleneck for both movies and video games.
For the video sequence, see ocean.mp4.

What to do... What to do...



7

But let's face it: modern CG images look great! So what do we do now? Well, we can take the blue pill, and forget that these problems were ever mentioned. Or, we can take the red pill, and start looking for solutions to improve current rendering techniques.

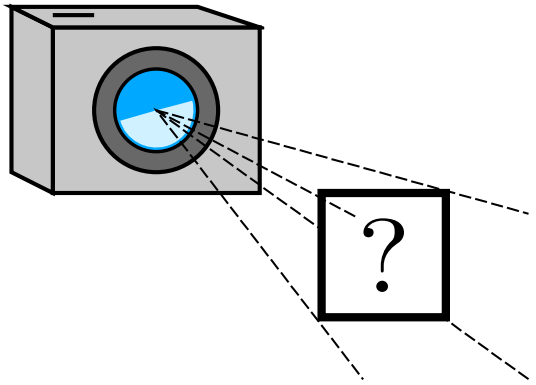
At the End of the Rabbit Hole

We **rederive the main results** from the **LEADR mapping** paper:

- We can filter normal and displacement texture maps with **microfacet theory**
- We can precompute these texture filtering operations efficiently using **MIP mapping**

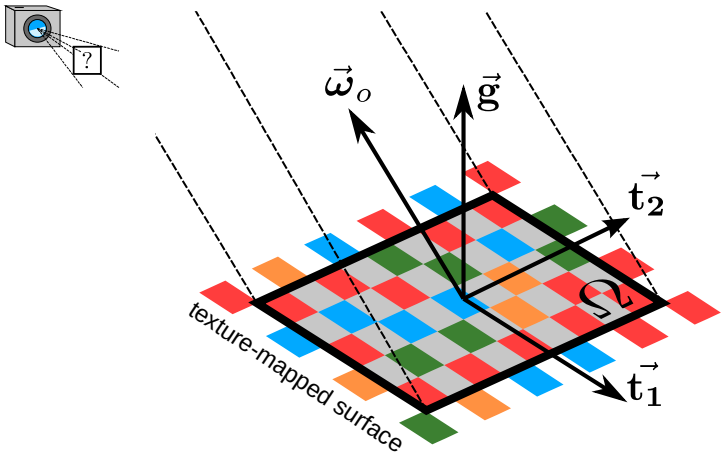
In this section of the course, we are going to have a look at how recent research has managed to deal with normal- and displacement-mapped surfaces. Dealing with these problems is a first step towards finding better CG representations that are free from the aforementioned limitations.

Pixel Color



We first start by looking at what happens at each pixel: how exactly is its color computed? For a physically based renderer, a pixel measures the amount of radiance that travels towards the camera, weighted by an anti-aliasing filter, such as a box filter.

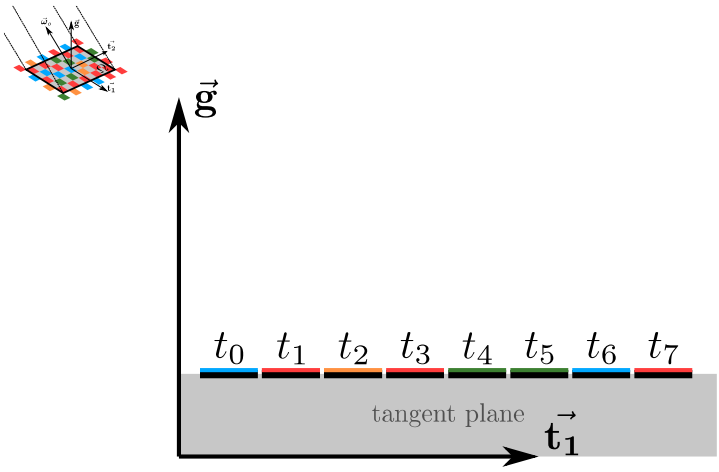
Pixel Color



10

The filter gives a footprint to the pixel in the 3D scene. Here, we assume that the footprint is entirely covered by a texture-mapped surface. A texture is composed of multiple texels, which are illustrated here as colored squares.

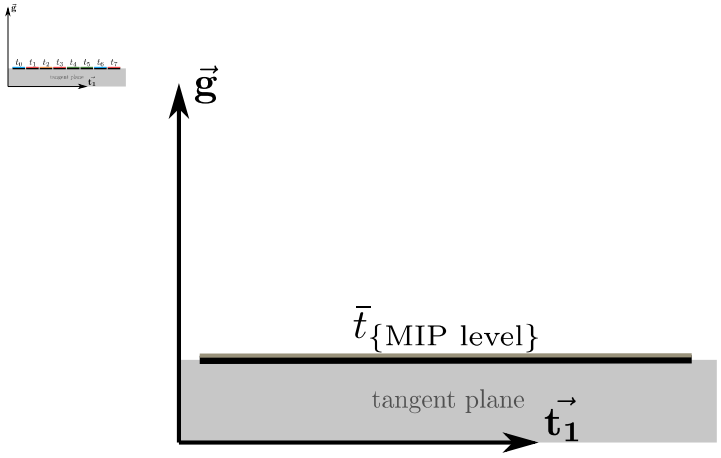
Pixel Color



texture-mapped surface profile
(render time increases with texel count)

Here is an illustration of the problem in 2D. If each texel gives the amount of radiance that leaves the surface towards the camera, then the computational complexity of the pixel scales linearly with the number of texels it covers.

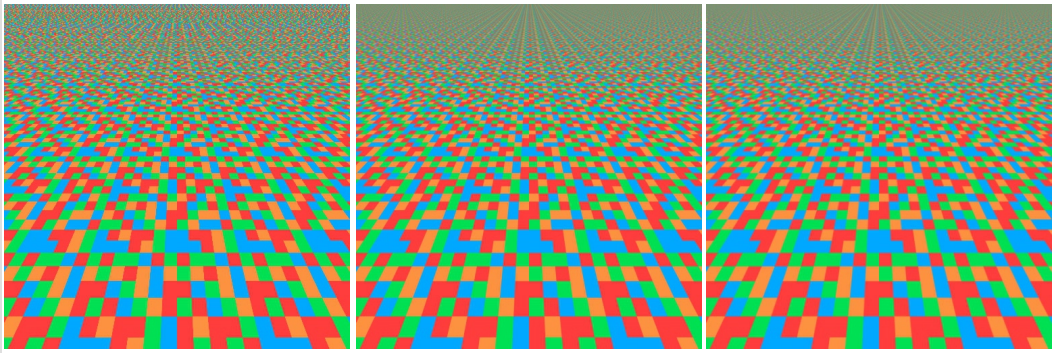
Pixel Color



MIP-mapped, texture-mapped surface profile
(render time is constant)

MIP maps offer a way to make the per-pixel computational complexity constant. This works by prefiltering the texture and storing the results inside a MIP hierarchy. If the texels represent an RGB triplet, then the precomputations are straightforward.

Pixel Color

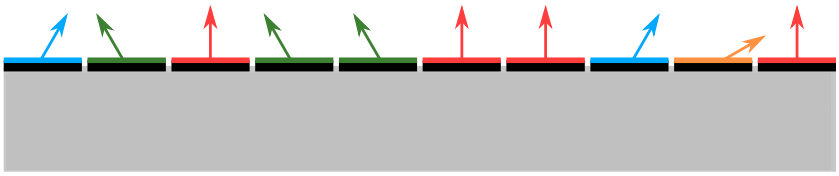


undersampled rendering | ground truth **(offline)** | MIP mapped **(real time)**

Here is an illustration of the benefit of using MIP maps. The ground truth and the pre-filtered renders are very close, only one is real time and has constant complexity per pixel, while the other is not. We would like to use a similar strategy for normal and displacement texture maps.
For the video sequence, see albedo.mp4.

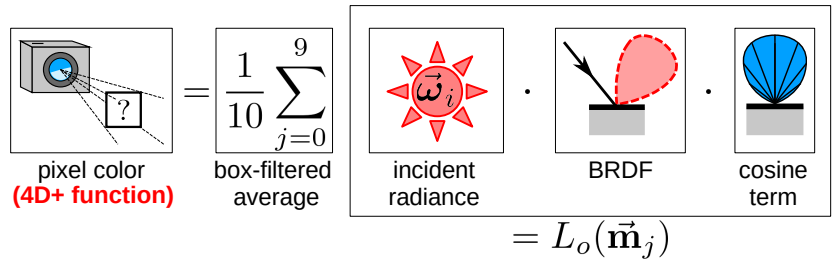
Problem 1:

MIP Mapping Normals

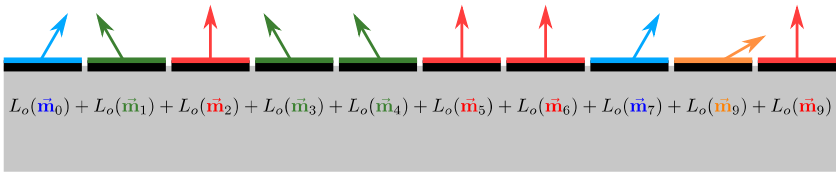


We'll start by looking at how to deal with normal maps. In this case, each texel stores a unit vector, which is used as input to a physically based shading model.

MIP Mapping Normals



$$= \frac{1}{10}\sum_{j=0}^{9} \vec{\omega}_i \cdot \cdot = L_o(\vec{\mathbf{m}}_j)$$

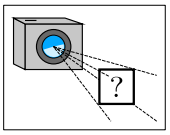


normal-mapped surface profile

15

Following physically based shading principles, we express the pixel color as a filtered, per-texel local illumination function (L_o). The illumination function is a product of an incident radiance function, a BRDF, and a cosine term. We use a box filter here, so the pixel color is simply the average of the illumination function. The resulting function is 4D: it depends on the orientation of the camera, and of the light source radiating the surface. Furthermore, it can be of arbitrarily high frequency. At this point, prefiltering is too expensive to be practical.

Normals Histogram



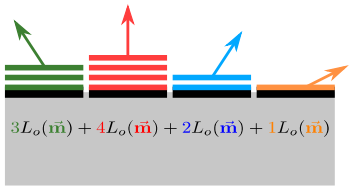
pixel color

$$= \sum_{\vec{\mathbf{m}} \in \mathcal{M}}$$

statistical space

$$\begin{bmatrix} L_o(\vec{\mathbf{m}}) \\ L_o(\vec{\mathbf{m}}) \\ L_o(\vec{\mathbf{m}}) \\ L_o(\vec{\mathbf{m}}) \end{bmatrix} \cdot \begin{bmatrix} D(\vec{\mathbf{m}}) = 3/10 \\ D(\vec{\mathbf{m}}) = 4/10 \\ D(\vec{\mathbf{m}}) = 2/10 \\ D(\vec{\mathbf{m}}) = 1/10 \end{bmatrix}$$

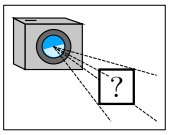
scattering events occurrence frequencies



normals histogram-mapped surface profile

The local illumination function has a nice property: it produces the same result for two similar normals. Hence, we can reduce the shading complexity by using a histogram. Intuitively, the histogram restructures the surface so that only different normals remain. The outgoing radiance at each normal is then scaled in proportion to its frequency of occurrence on the original surface.

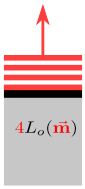
Fresnel Mirrors



pixel color

$$= \underbrace{\vec{\omega}_i \cdot \frac{F(\vec{\omega}_h)}{4(\vec{\omega}_h \cdot \vec{\omega}_i)}}_{= L_o(\vec{\omega}_h)} \cdot \underbrace{D(\vec{\omega}_h)}_{\text{(2D function)}}$$

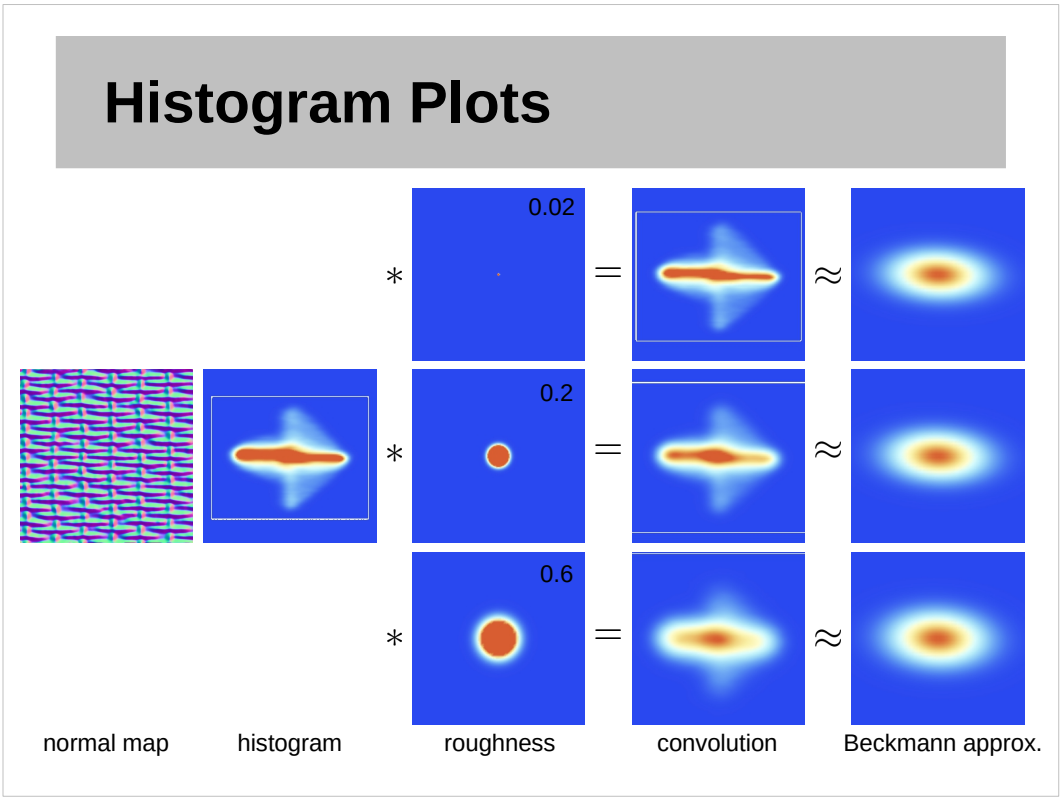
microfacet model!



$\left(\vec{\mathrm{m}} = \frac{\vec{\omega}_o + \vec{\omega}_i}{|\vec{\omega}_o + \vec{\omega}_i|} = \vec{\omega}_h\right)$

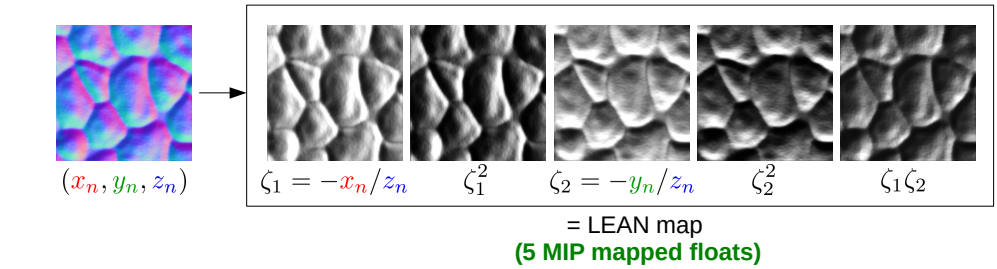
normals histogram-mapped Fresnel mirror profile

If we use a Fresnel mirror BRDF, the equation simplifies to a microfacet model. We can deal with any glossy BRDF by convolving their histograms. While the histogram is a 2D function, It is still problematic to precompute...

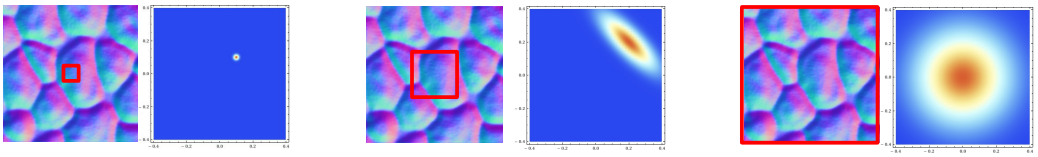


To make this clearer, here is the histogram of a normal map, plotted in slope space. Normals maps can be of arbitrarily high frequency, so they may require memory-intensive discretizations, which we wish to avoid. However, we can still make progress, since in practice normal distribution functions are convolved with some base roughness at shading time. The third column shows an amount of base roughness which is used to convolve the histogram; notice how much smoother it becomes as roughness increases. The fourth column shows an approximation of the histogram as a parametric Gaussian density. While some details are not present, we are able to capture the main direction of anisotropy.

LEAN Mapping



= LEAN map
(5 MIP mapped floats)



compact histogram per texel MIP

19

Gaussian densities are especially interesting because they have five parameters that can be calculated from a LEAN texture map. A LEAN map consists of five terms that can be linearly prefiltered (MIP mapped). These terms can be derived from a normal map very straightforwardly.
This is extremely cheap, and works at all scales. For more details on how this works, please take a look at the LEAN or LEADR mapping papers, or my course notes. If we plug this parametric density into the microfacet BRDF that we just derived, we arrive at a constant per-pixel rendering cost.

Result

undersampled rendering

ground truth
(slow)

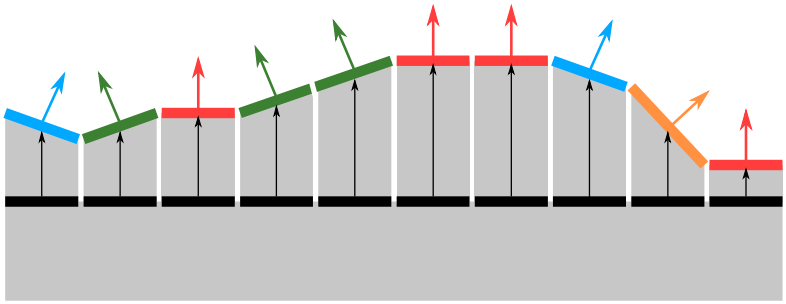
LEAN mapping
+
microfacet BRDF
(real time)

20

Here is the result for an ocean scene. The Gaussian density tends to smooth out the image compared to the ground truth, but the image quality is way better than that of an undersampled solution. So by combining a microfacet model and a LEAN map, we can achieve artifact-free images that still retain all physically based shading principles.
For the video sequence, see ocean.mp4.

Problem 2:

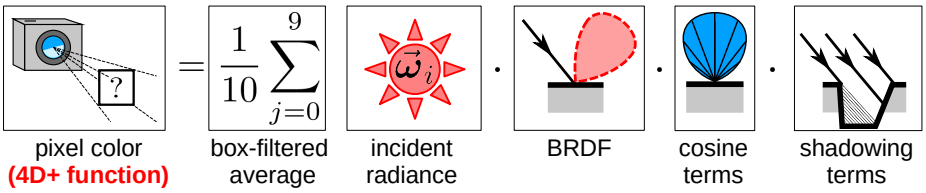
MIP Mapping Displacements



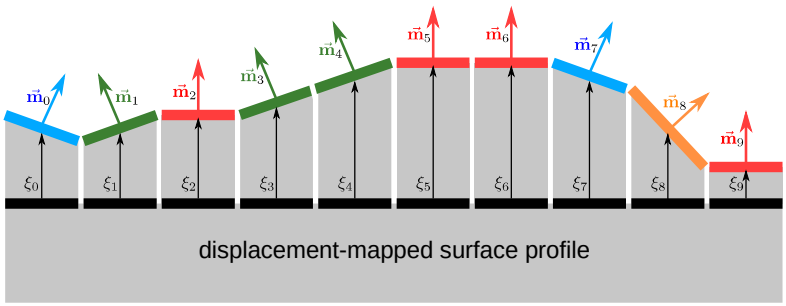
21

We would like to apply the same idea to displacements as well. Displacement maps are more complex than normal maps because they affect surfaces both in terms of shape and appearance.

MIP Mapping Displacements

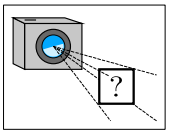


pixel color **(4D+ function)** | box-filtered average | incident radiance | BRDF | cosine terms | shadowing terms



As a consequence, the scattering model becomes more complex. Note the introduction of the shadowing terms, that must account for the amount of occlusion that occurs on the surface. Just like the normal-mapped case, this function cannot be precomputed at this point.

Visible Normals Histogram



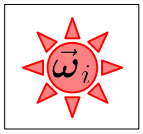
pixel color
(4D+ function)

$$= \sum_{\vec{\mathbf{m}} \in \mathcal{M}} \begin{bmatrix} L_o(\vec{\mathbf{m}}) \\ L_o(\vec{\mathbf{m}}) \\ L_o(\vec{\mathbf{m}}) \\ L_o(\vec{\mathbf{m}}) \end{bmatrix} \cdot \begin{bmatrix} D_{\text{vis}}(\vec{\mathbf{m}}, \vec{\boldsymbol{\omega}}_o) \\ D_{\text{vis}}(\vec{\mathbf{m}}, \vec{\boldsymbol{\omega}}_o) \\ D_{\text{vis}}(\vec{\mathbf{m}}, \vec{\boldsymbol{\omega}}_o) \\ D_{\text{vis}}(\vec{\mathbf{m}}, \vec{\boldsymbol{\omega}}_o) \end{bmatrix}$$

statistical space

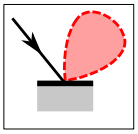
scattering events | occurrence frequencies

$$L_o(\vec{\mathbf{m}}) =$$



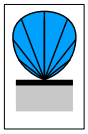
incident radiance

$\cdot$



BRDF

$\cdot$

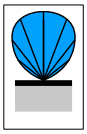


cosine terms

$\cdot$

shadowing

$$D_{\text{vis}}(\vec{\mathbf{m}}, \vec{\boldsymbol{\omega}}_o) = D(\vec{\mathbf{m}}) \cdot$$



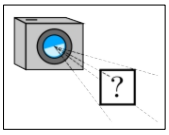
cosine terms

$\cdot$

masking

We can still make progress with the histogram approach, but this time it is higher dimensional because we only wish to retrieve the normals that are visible from the camera.

Fresnel Mirrors



pixel color
(4D+ function)

$$= \underbrace{\underbrace{\vec{\omega}_i \cdot \frac{F(\vec{\omega}_h)G(\vec{\omega}_i)}{4}}_{= L_o(\vec{\omega}_h)} \cdot D_{\text{vis}}(\vec{\omega}_h, \vec{\omega}_o)}_{\text{microfacet model!}}$$

(4D function)

$$L_o(\vec{\mathbf{m}}) = \vec{\omega}_i \cdot \cdot \cdot \cdot$$

incident radiance | BRDF | cosine terms | shadowing

$$D_{\text{vis}}(\vec{\mathbf{m}}, \vec{\omega}_o) = D(\vec{\mathbf{m}}) \cdot \cdot$$

cosine terms | masking

Interestingly, by using a Fresnel mirror BRDF, the model produces the equation of a microfacet BRDF.

Shadowing Models

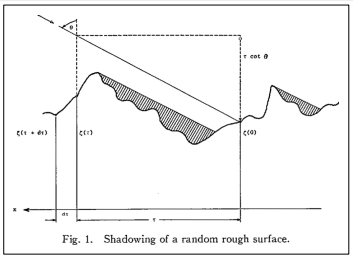


Fig. 1. Shadowing of a random rough surface.

[Bec1965]

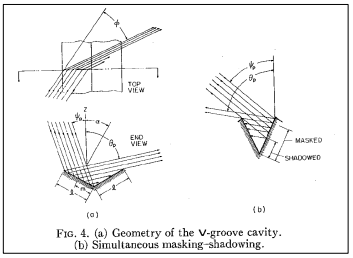


FIG. 4. (a) Geometry of the V-groove cavity. (b) Simultaneous masking–shadowing.

[TS1967]

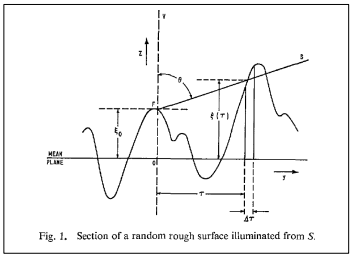
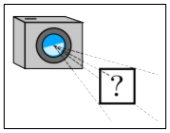


Fig. 1. Section of a random rough surface illuminated from S.

[Smi1967]

Hence, we can take advantage of the shadowing theory that was developed in the 1960s and progressively incorporated into microfacet theory.

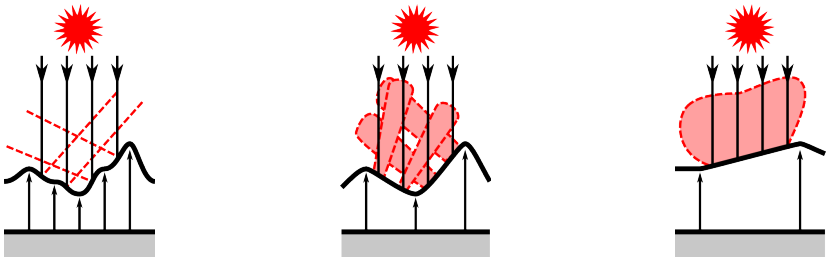
Fresnel Mirrors



pixel color

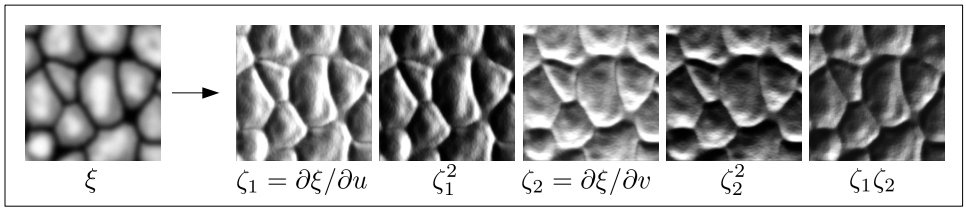
$$= \vec{\omega}_i \cdot \frac{F(\vec{\omega}_h)G(\vec{\omega}_i)}{4} \cdot \frac{(\vec{\mathbf{n}} \cdot \vec{\mathbf{g}})G(\vec{\omega}_o)D(\vec{\omega}_h)}{(\vec{\omega}_i \cdot \vec{\mathbf{g}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})(\vec{\omega}_h \cdot \vec{\mathbf{g}})}$$

LEADR mapping microfacet model

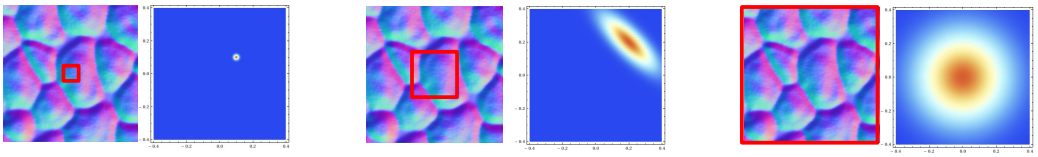


This is how LEADR mapping solves the problem of modeling visibility at all scales. In the paper, we use the Smith shadowing function. Note that we had to extend microfacet theory to work with non-central normal densities, in order to apply it on displacement maps. This is because displacement maps are used to deviate the mean microsurface normal from the tangent-frame up direction.

LEADR Mapping



= LEADR map
(6 MIP mapped floats)



compact histogram per texel MIP

27

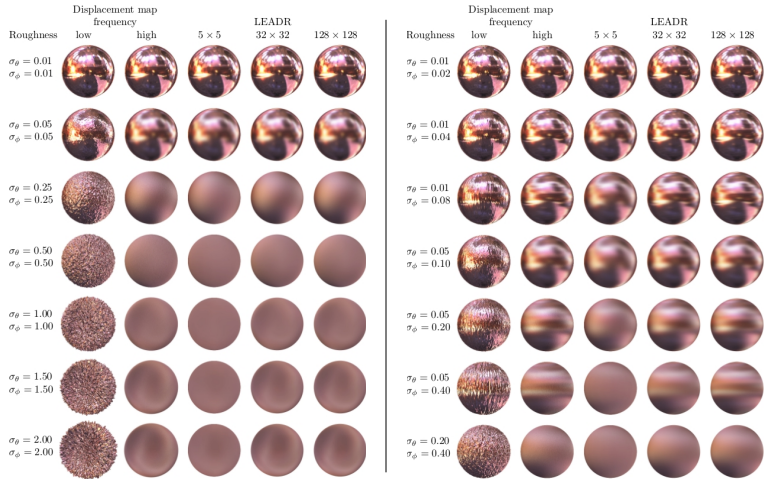
We also show that we can use the same representation as a LEAN map to compute the correct histogram at all scales.

Results

Here is the main video of the LEADR mapping paper. The video is available on YouTube: www.youtube.com/watch?v=ND1USWLNGb0, or on HAL: hal.inria.fr/hal-00858220/en.

Validation

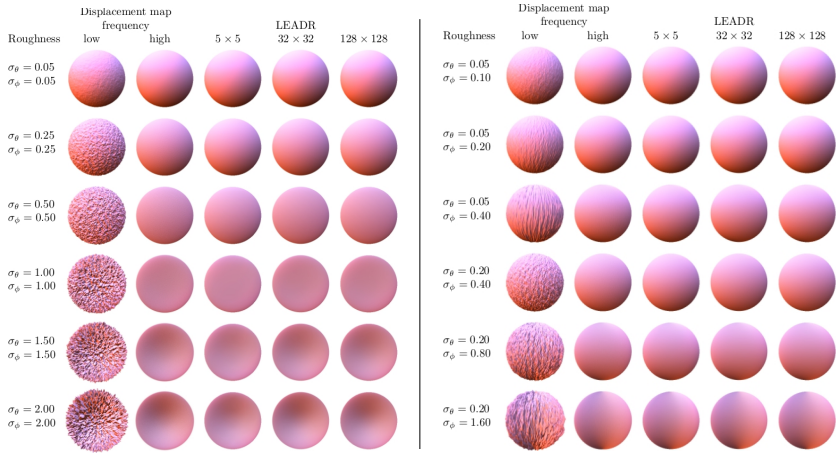
Fresnel mirror BRDFs



We derived and validated our model by generating Gaussian microscopic slopes on the surface of a sphere.

Validation

Diffuse BRDFs



And also introduced a model for rough diffuse surfaces.

Properties

- **Scalable** (constant per pixel rendering time)
- **Linear** (prefilter = `GenerateMipmap()`)
- **Lightweight** (5 floats per texel)
- **Physically based BRDFs** (energy conservation)
- **All-frequency BRDFs** (diffuse and specular)
- **Anisotropic BRDFs**
- **All-frequency lighting** ({point, directional, IBL} lighting)
- **Compatible with animation** (supports mesh deformation)

LEADR mapping possesses the fundamental properties that make it general enough to be applied in production.

Future Work

More prefiltered representations

Scalability beyond texture-mapped surfaces





This concludes the talk. With LEADR mapping, we switched rendering complexity from the number of details to the number of surfaces in the pixel. A big avenue for future work is to be able to deal with collections of such surfaces. We already have some theoretical models that we can use to start making progress as well, such as the “microflakes” model.

References

[Bec1965] Petr Beckmann. Shadowing of Random Rough Surfaces. IEEE Trans. on Antennas and Propagation, 13(3):384--388, May 1965.

[DHIPNO2013] Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, Fabrice Neyret, and Victor Ostromoukhov. Linear Efficient Antialiased Displacement and Reflectance Mapping. ACM Trans. Graph., 32(6):211:1--11, November 2013.

[OB2010] Marc Olano and Dan Baker. LEAN Mapping. In Proc. Symp. on Interactive 3D Graphics and Games, pages 181--188. ACM, 2010.

[Smi1967] B. Smith. Geometrical Shadowing of a Random Rough Surface. IEEE Trans. on Antennas and Propagation, 15(5):668--671, 1967.

[TS1967] K. E. Torrance and E. M. Sparrow. Theory for Off-Specular Reflection from Roughened Surfaces. J. Opt. Soc. Am., 57(9):1105--1112, Sep 1967.