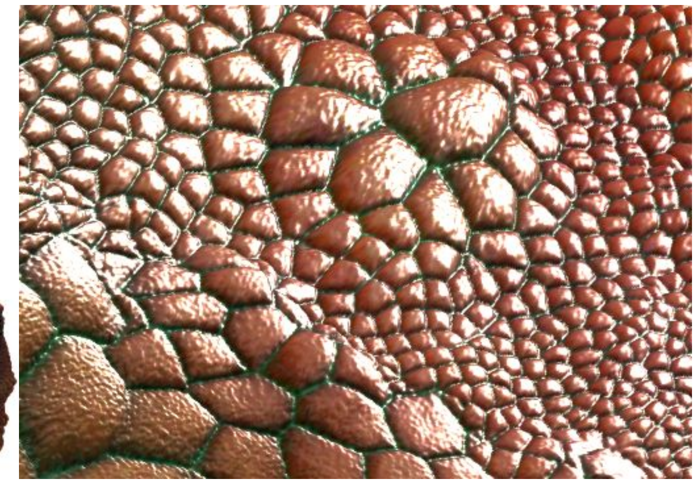
Antialiasing Physically Based Shading with LEADR Mapping

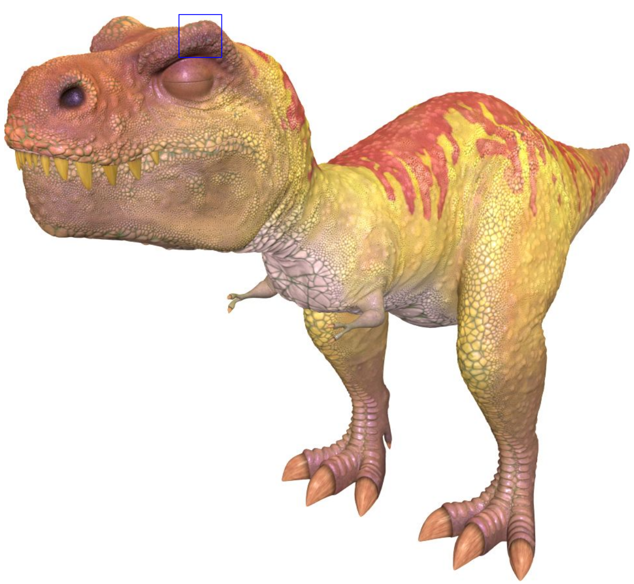


From:

- *Physically Based Shading in Theory and Practice*
  SIGGRAPH 2014 Course Jonathan Dupuy
- *Linear Efficient Antialiased Displacement and Reflectance Mapping*
  SIGGRAPH Asia 2013 Jonathan Dupuy Eric Heitz
  Jean-Claude Iehl Pierre Poulin Fabrice Neyret Victor Ostromoukhov
- *Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs*
  JCJT 2014 Eric Heitz
- *Filtering Color Mapped Textures and Surfaces*
  I3D 2010 - TVCG2014

Meet Tiny the T. Rex

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

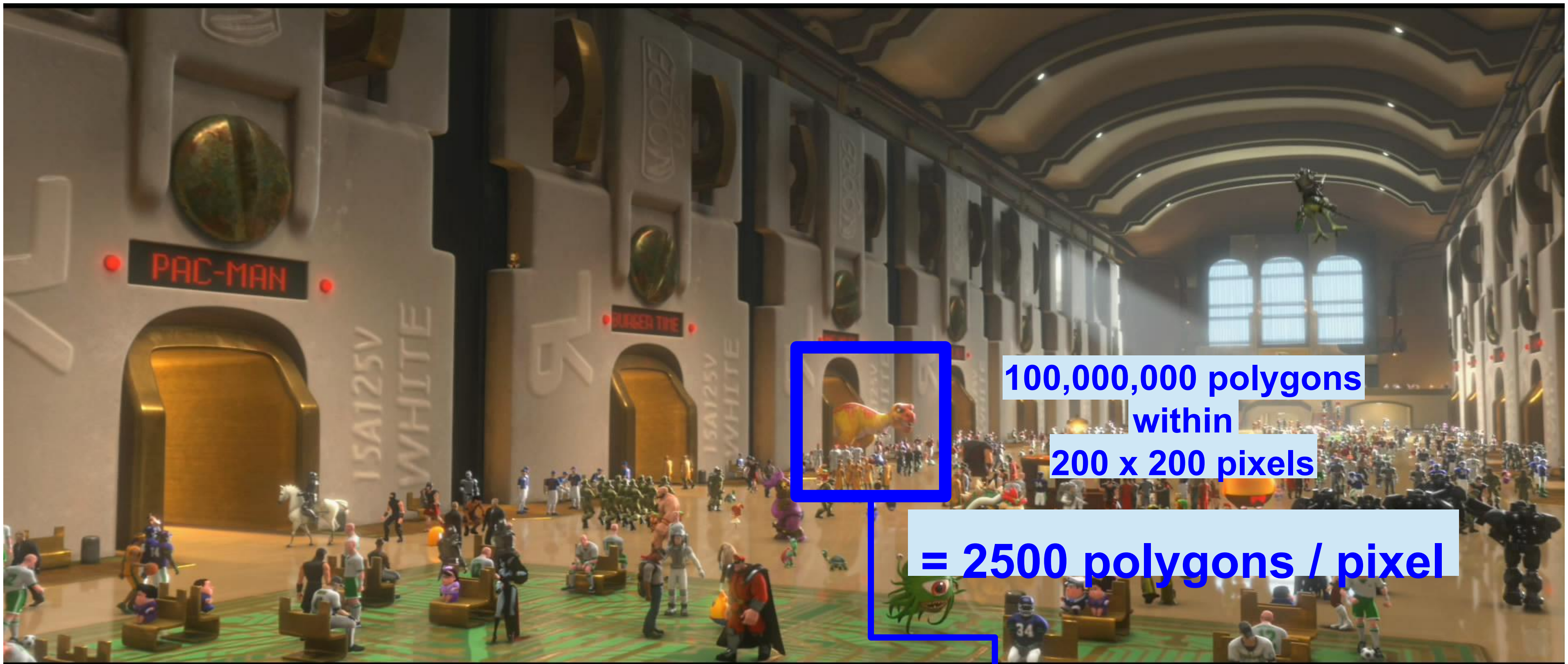


Tiny the T. Rex
(© Disney)

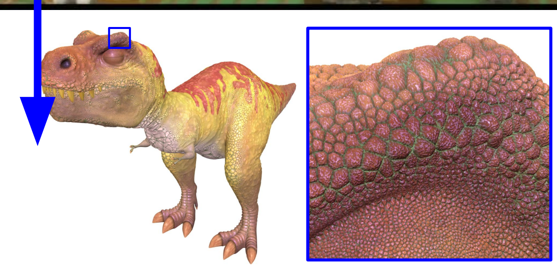
≈ 100,000,000 polygons
for 4 gigabytes

≈ 42 bytes per polygon

A Challenge in Rendering



Wreck-It Ralph (© Disney 2012)



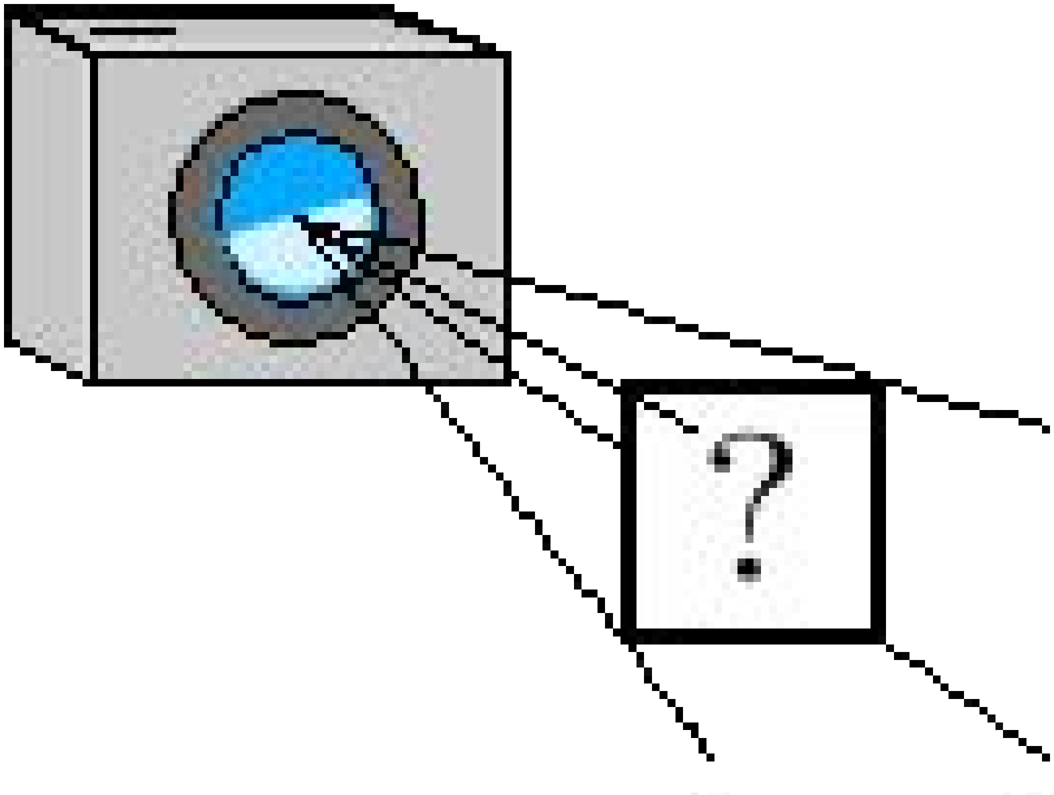
Undersampling Misdeeds

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

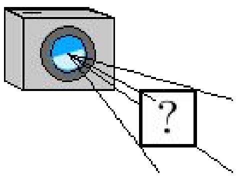
undersampled rendering

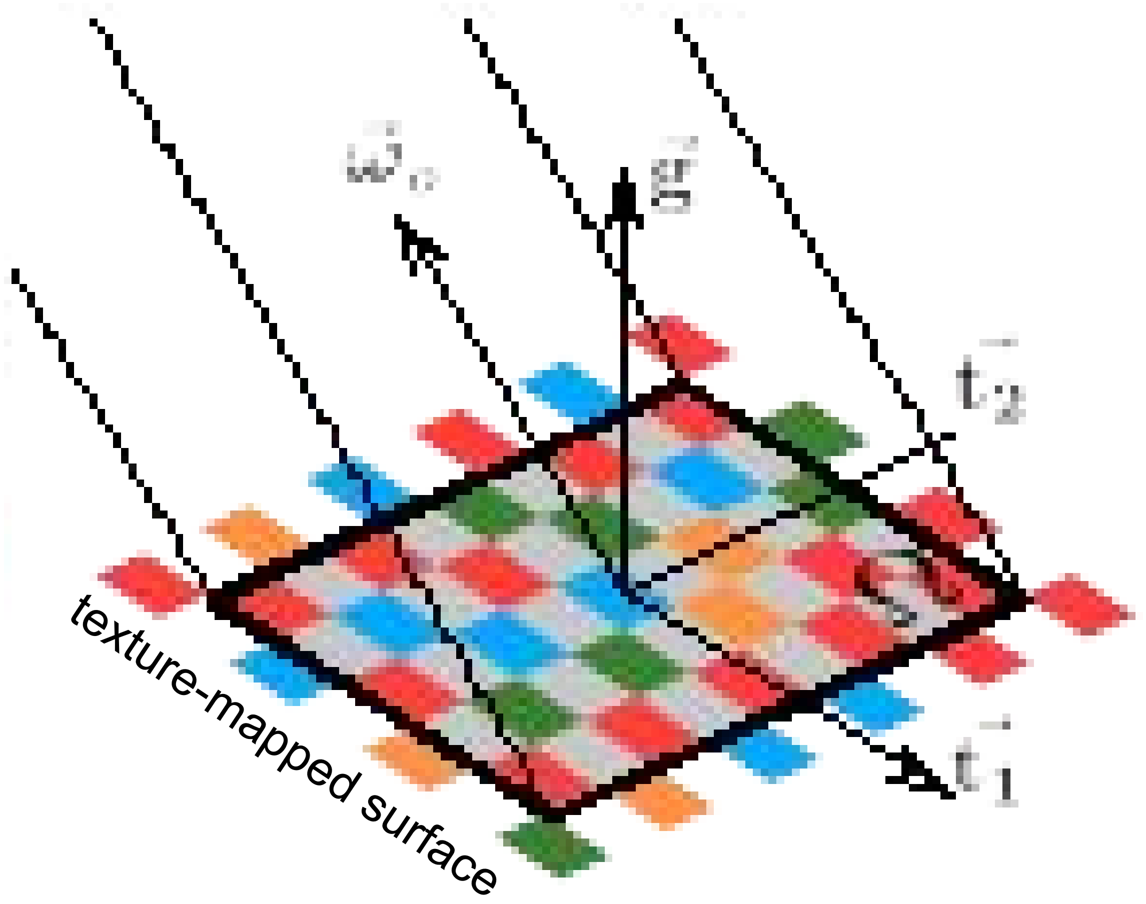
ground truth

Pixel Color

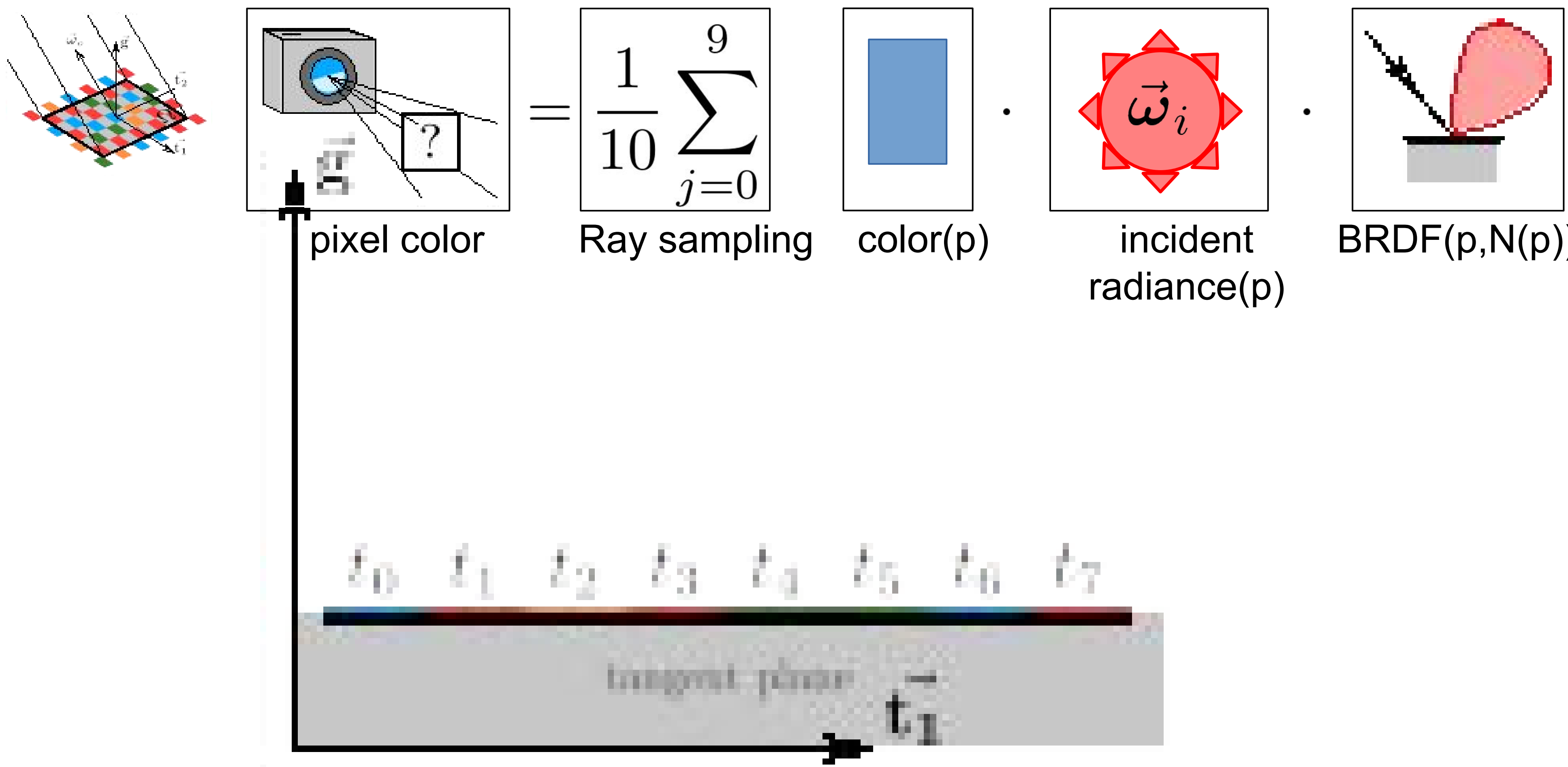


Pixel Color





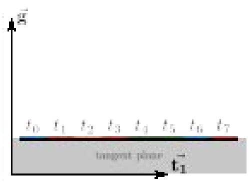
Pixel Color

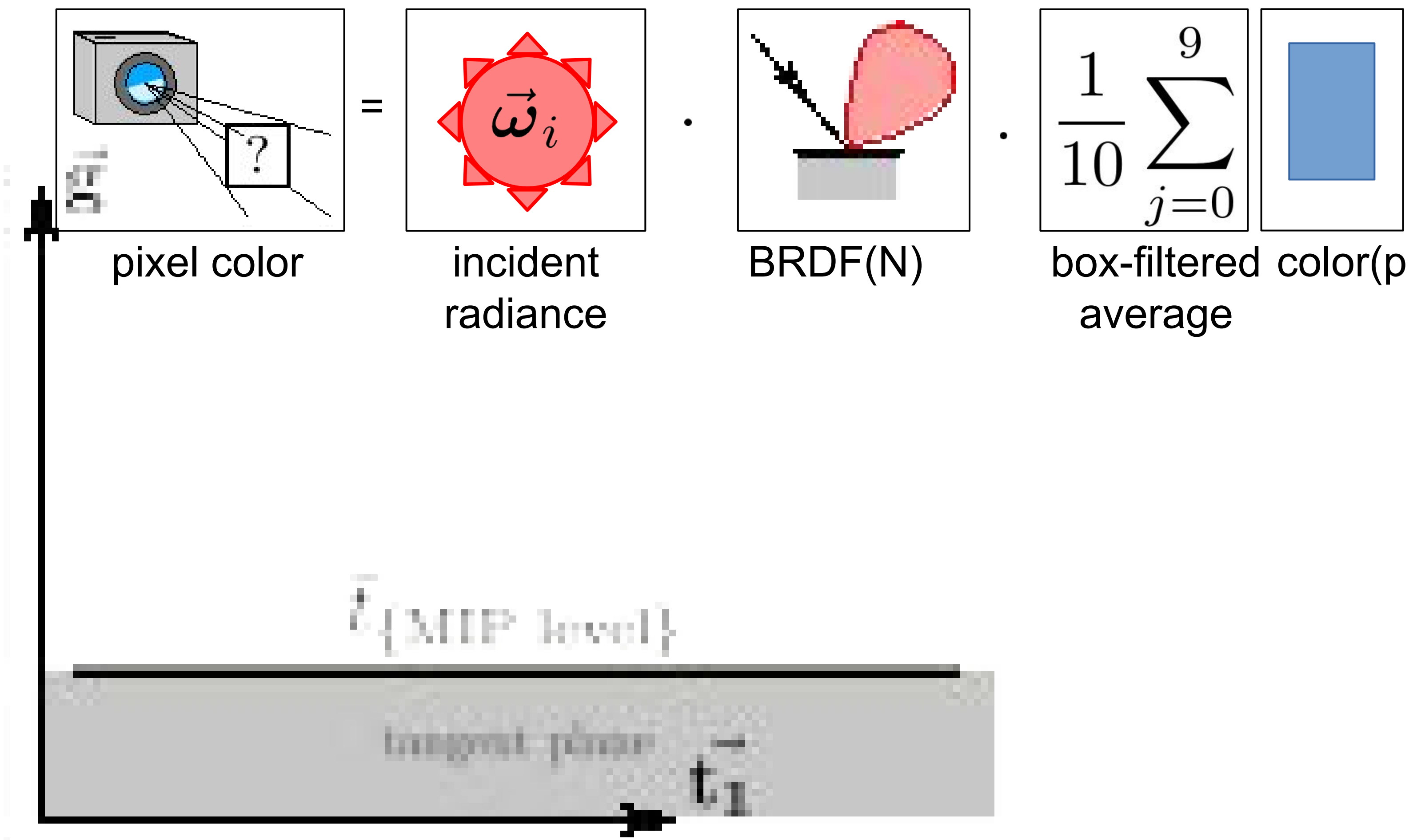


texture-mapped surface profile

(render time increases with texel count)

Pixel Color

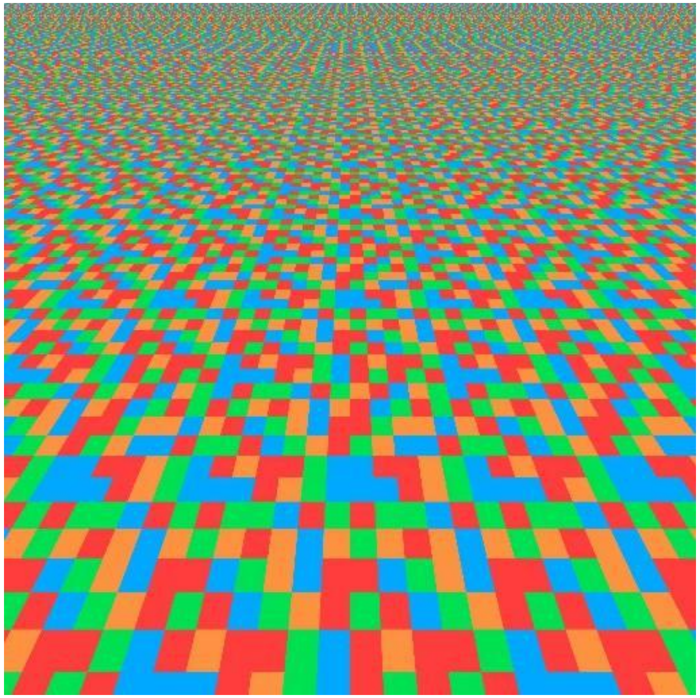




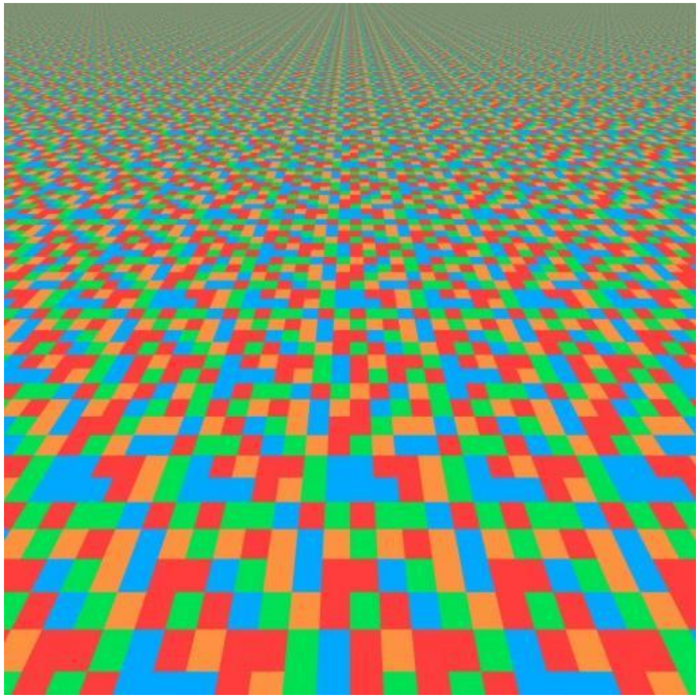
MIP-mapped, texture-mapped surface profile

(render time is constant)

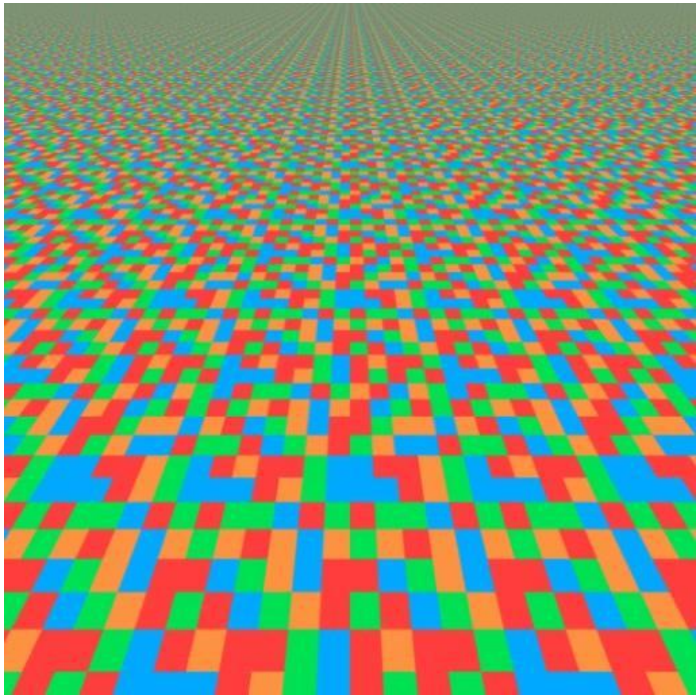
Pixel Color



undersampled rendering



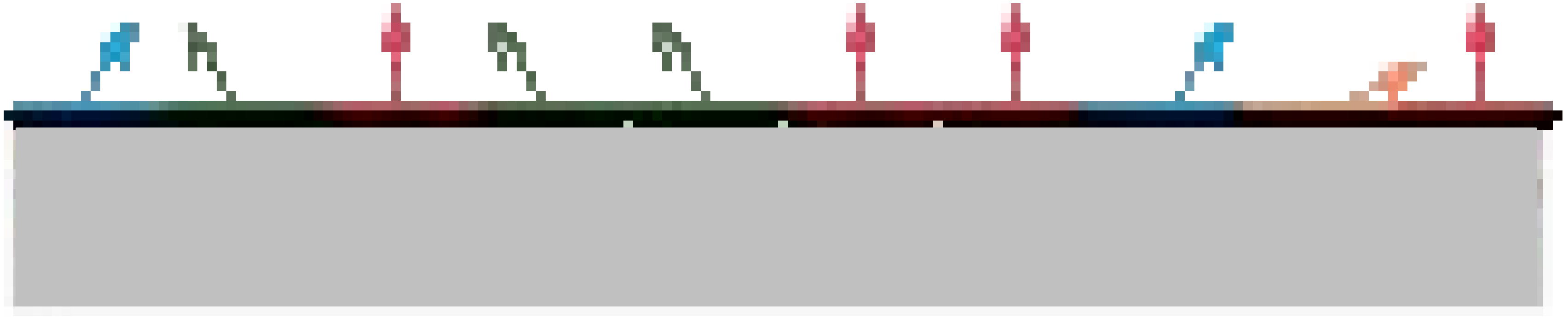
ground truth
(offline)



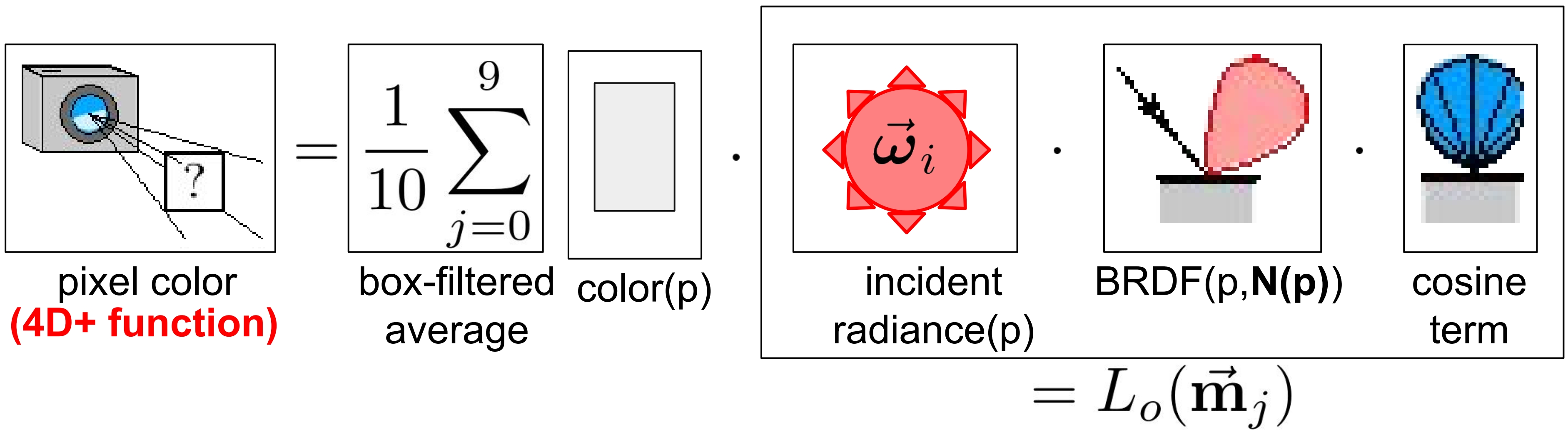
MIP mapped
(real time)

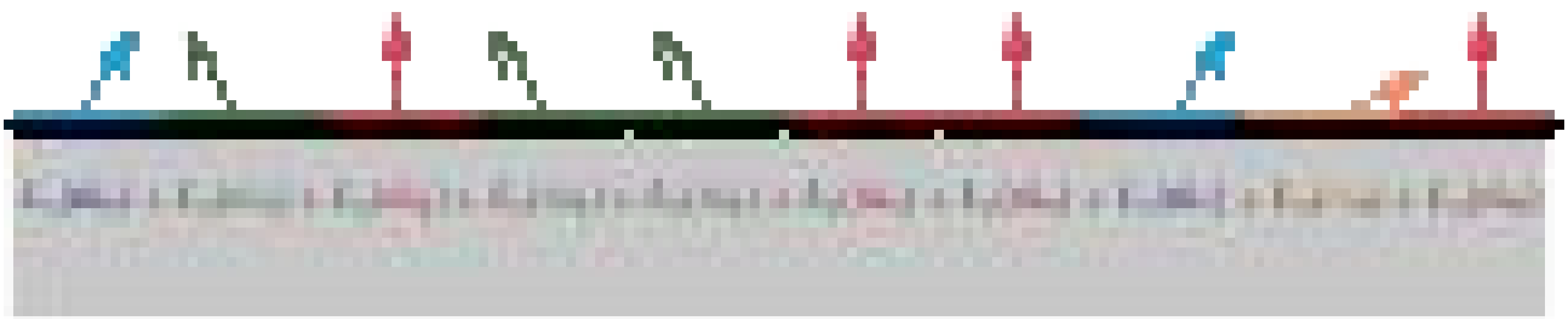
Problem 1:

MIP Mapping Normals



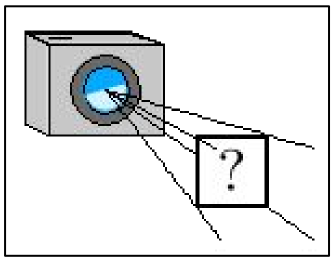
MIP Mapping Normals





normal-mapped surface profile

Normals Histogram



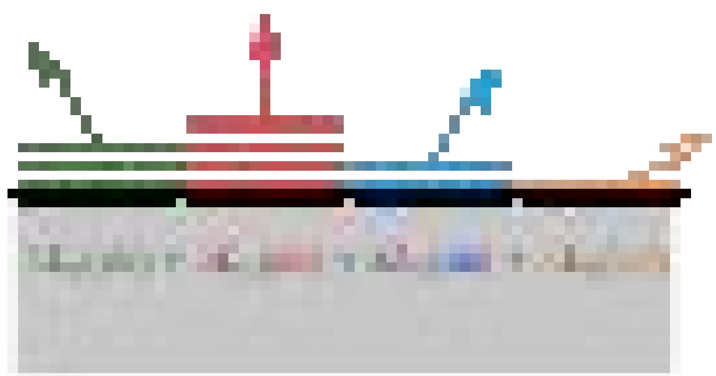
pixel color

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

statistical space

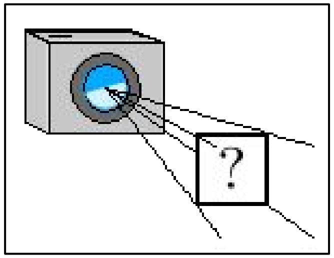
scattering events

occurrence frequencies



normals histogram-mapped surface profile

Fresnel Mirrors



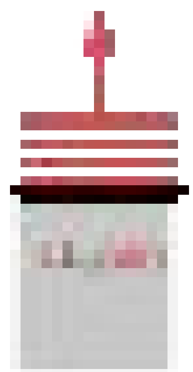
pixel color

$$= \vec{\omega}_i \cdot \frac{F(\vec{\omega}_h)}{4(\vec{\omega}_h \cdot \vec{\omega}_i)} \cdot D(\vec{\omega}_h)$$

$$= L_o(\vec{\omega}_h)$$

(2D function)

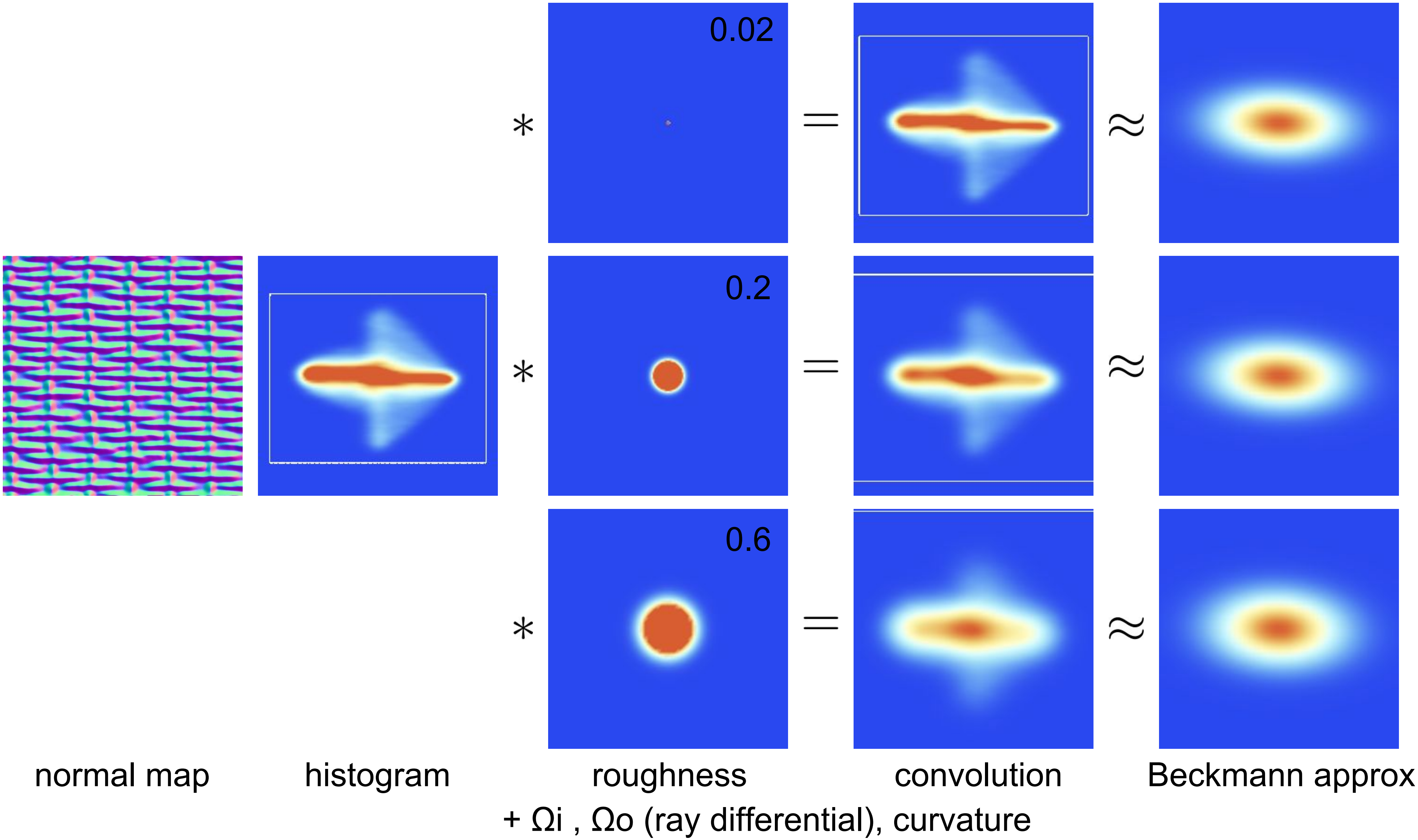
microfacet model!



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

normals histogram-mapped Fresnel mirror profile

Histogram Plots (slopes)



Physically based BRDFs

Microfacet NDF = microfacet slope PDF

–

gaussian slope slope variance

$$\text{Beckmann} \equiv \frac{\exp(-\tan^2(\alpha)/\sigma^2)}{\pi\sigma^2\cos^4(\alpha)}, \quad \alpha = \arccos(N \cdot H)$$

Jacobian

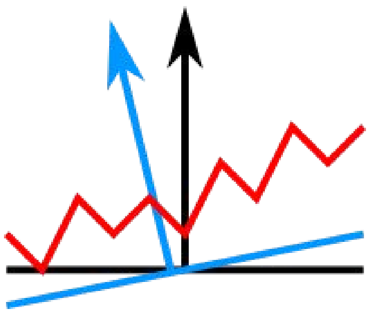
Beckmann parameters = slope distribution

$$\frac{\exp\left(-\frac{1}{2}\left(\tilde{n} - \mathbb{E}[\tilde{n}]\right)^t \Sigma^{-1} \left(\tilde{n} - \mathbb{E}[\tilde{n}]\right)\right)}{2\pi\sqrt{|\Sigma|}}$$

- +anisotropic

$$\Sigma = \begin{bmatrix} \sigma_x^2 & c_{xy} \\ c_{xy} & \sigma_y^2 \end{bmatrix}$$

- Missing: average normal – D() can be non-centered

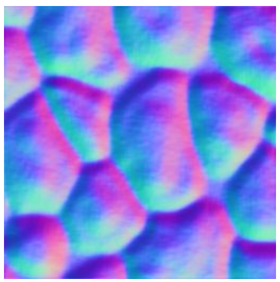


LEAN Mapping

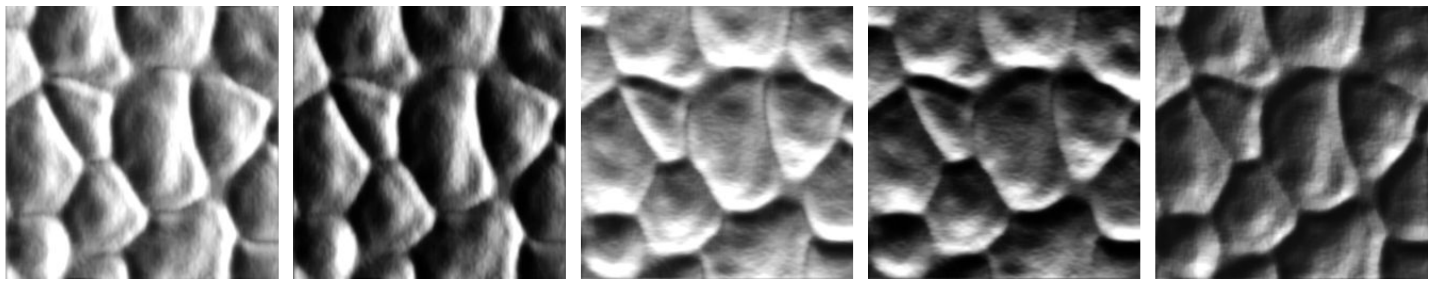
Store statistics (mean, var) at each scale

E(x) linear→ MIPmap

var = $E(x^2) - E(x)^2$ → MIPmap



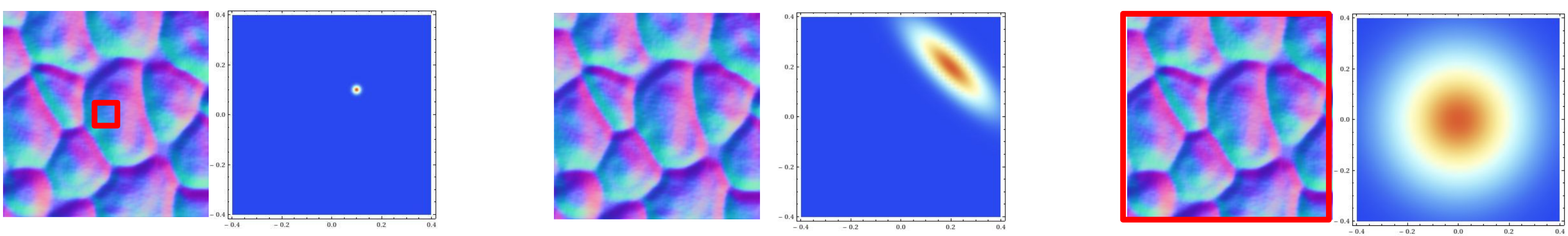
(x_n, y_n, z_n)



$\zeta_1 = -x_n/z_n$ ζ_1^2 $\zeta_2 = -y_n/z_n$ ζ_2^2 $\zeta_1\zeta_2$

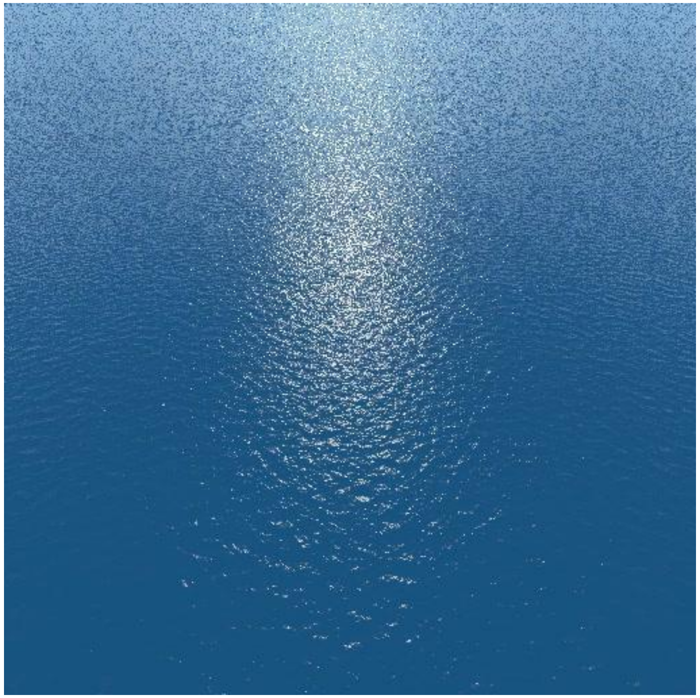
= LEAN map

(5 MIP mapped floats)



compact histogram per texel MIP

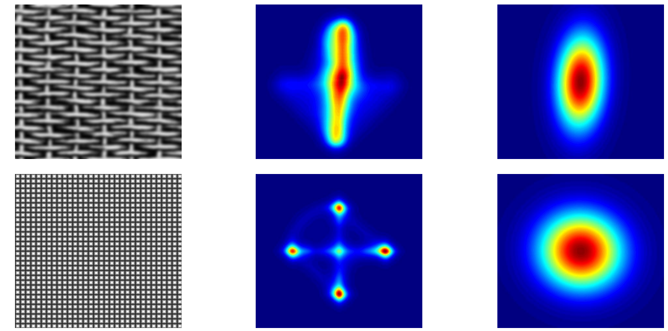
Result


undersampled rendering


ground truth
(slow)


LEAN mapping
+
microfacet BRDF
(real time)

Hypothesis / Approximations



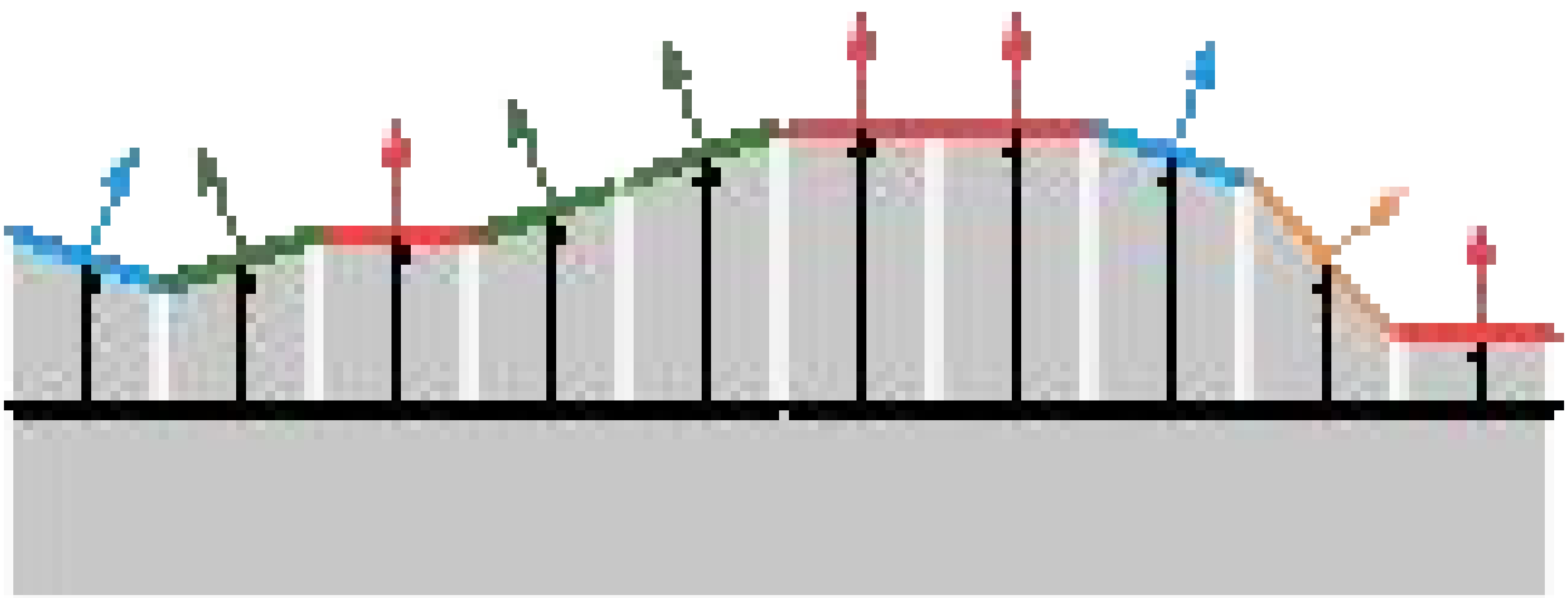
displacement slope PDF Gaussian approximation

- Slopes gaussianity →
- Slope space → (S)GGX
- Separability → *"Filtering Color Mapped Textures and Surfaces"* [I3D'13]
- Flat macro-surface → * curvature trick
- MIPmap approxs → (independent issue)
- interpolation imprecision for $\sigma=0$ & $Ex\neq0$

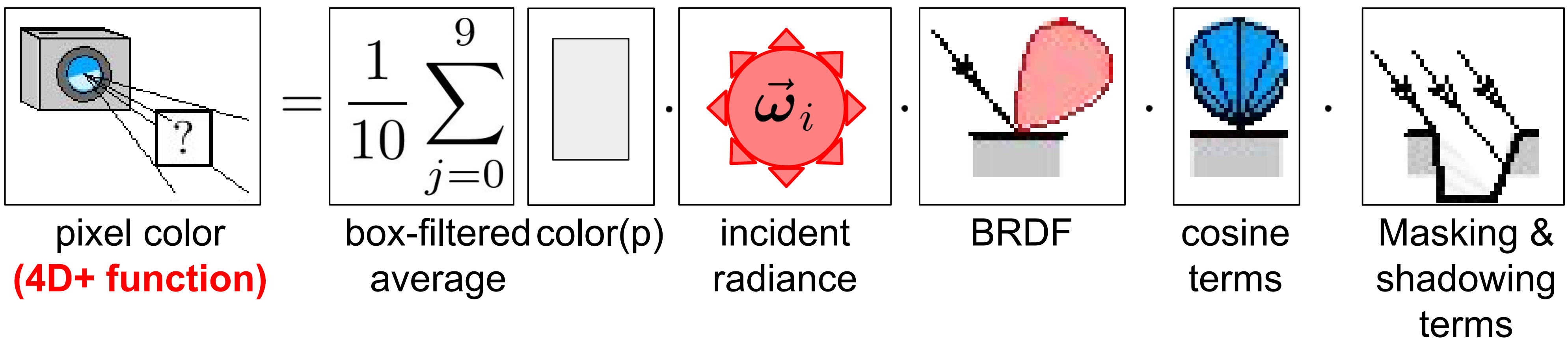
- Normal map

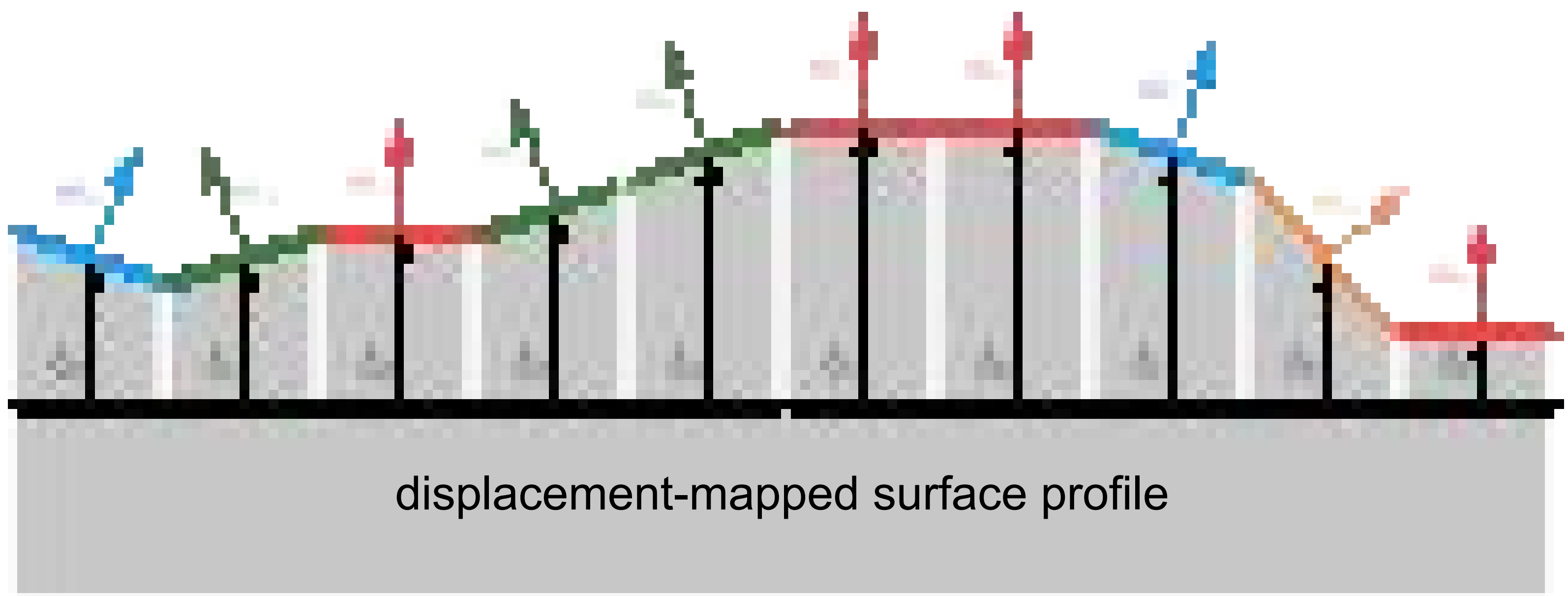
Problem 2:

MIP Mapping Displacements

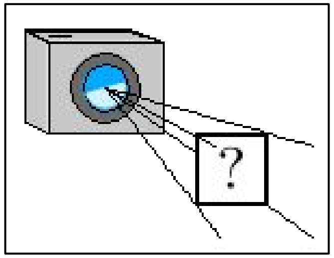


MIP Mapping Displacements





Visible Normals Histogram



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

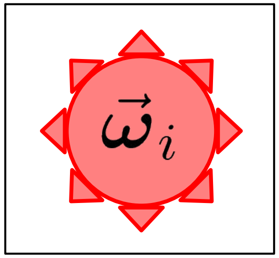
pixel color **(4D+ function)**

statistical space

scattering events

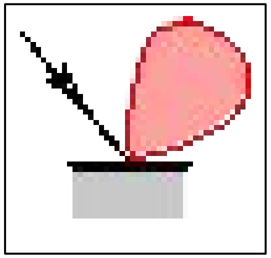
occurrence frequencies

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



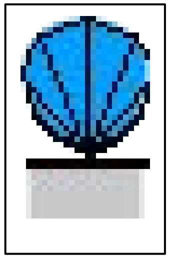
incident radiance

$\cdot$



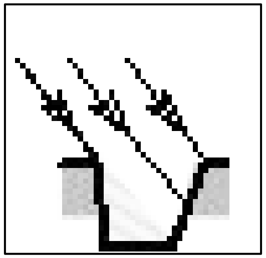
BRDF

$\cdot$



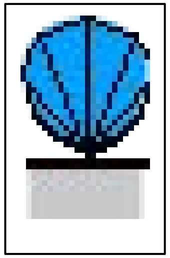
cosine terms

$\cdot$



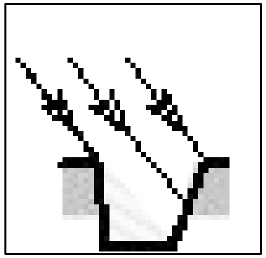
shadowing

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



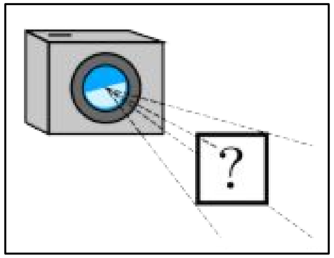
cosine terms

$\cdot$



masking

Fresnel Mirrors



pixel color
(4D+ function)

$$= \vec{\omega}_i \cdot \frac{F(\vec{\omega}_h)G(\vec{\omega}_i)}{4} \cdot D_{\text{vis}}(\vec{\omega}_h, \vec{\omega}_o)$$

$$= L_o(\vec{\omega}_h)$$

(4D function)

microfacet model!

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

Diffuse: no close form, (light) numerical integration

Statistical Model

- $\mathbb{E}[R(p,\omega_o,\omega_i)]$ ~ $\mathbb{E}[V(p,\omega_o)V(p,\omega_i)]\,\mathbb{E}[R_n(\omega_n,\omega_o,\omega_i)]$
- Masking-shadowing term: [Smith / Ross / Bourlier00]
- $$\mathbb{E}[V(p,\omega_o)V(p,\omega_i)] = \frac{1}{1+\Lambda(\omega_o)+\Lambda(\omega_i)}$$

$$\Lambda(\omega) = \frac{\exp\left(-\nu^2\right)}{2\nu\sqrt{\pi}} - \frac{\operatorname{erfc}(\nu)}{2} \approx \begin{cases} \frac{1.0-1.259\nu+0.396\nu^2}{3.535\nu+2.181\nu^2} & \text{if } \nu < 1.6 \\ 0 & \text{otherwise.} \end{cases}$$

- $\nu = \frac{\cot\theta-\mu(\phi)}{\sigma(\phi)\sqrt{2}}$
- $\mu(\phi) = \cos\phi\,\mathbb{E}[x_{\tilde{n}}] + \sin\phi\,\mathbb{E}[y_{\tilde{n}}]$
- $\sigma^2(\phi) = \cos^2\phi\,\sigma_x^2 + \sin^2\phi\,\sigma_y^2 + 2\cos\phi\sin\phi\,c_{xy}$
-
- Missing: non-centered

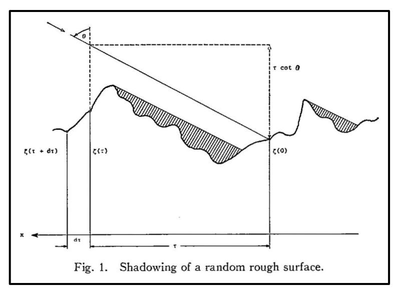


Fig. 1. Shadowing of a random rough surface.

[Bec1965]

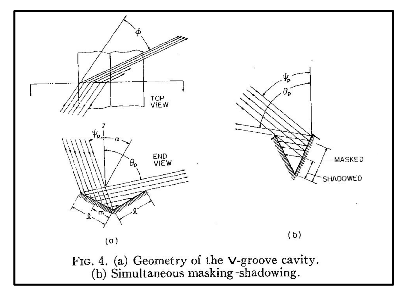


FIG. 4. (a) Geometry of the V-groove cavity. (b) Simultaneous masking–shadowing.

[TS1967]

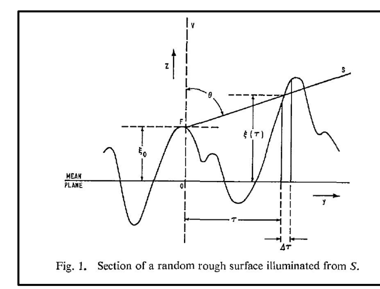


Fig. 1. Section of a random rough surface illuminated from S.

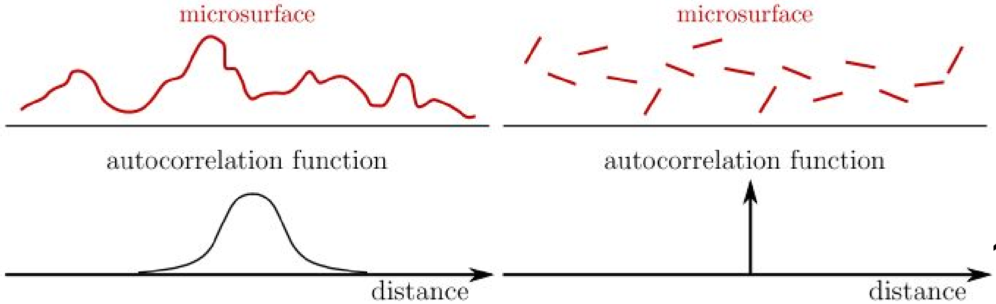
[Smi1967]

Hypothesis / Approximations

- Smith is the only physically coherent

  white furnace validation , cf *"Understanding Masking-Shadowing..."* [JCJT'14]

- Hypothesis: visibility & slopes not correlated

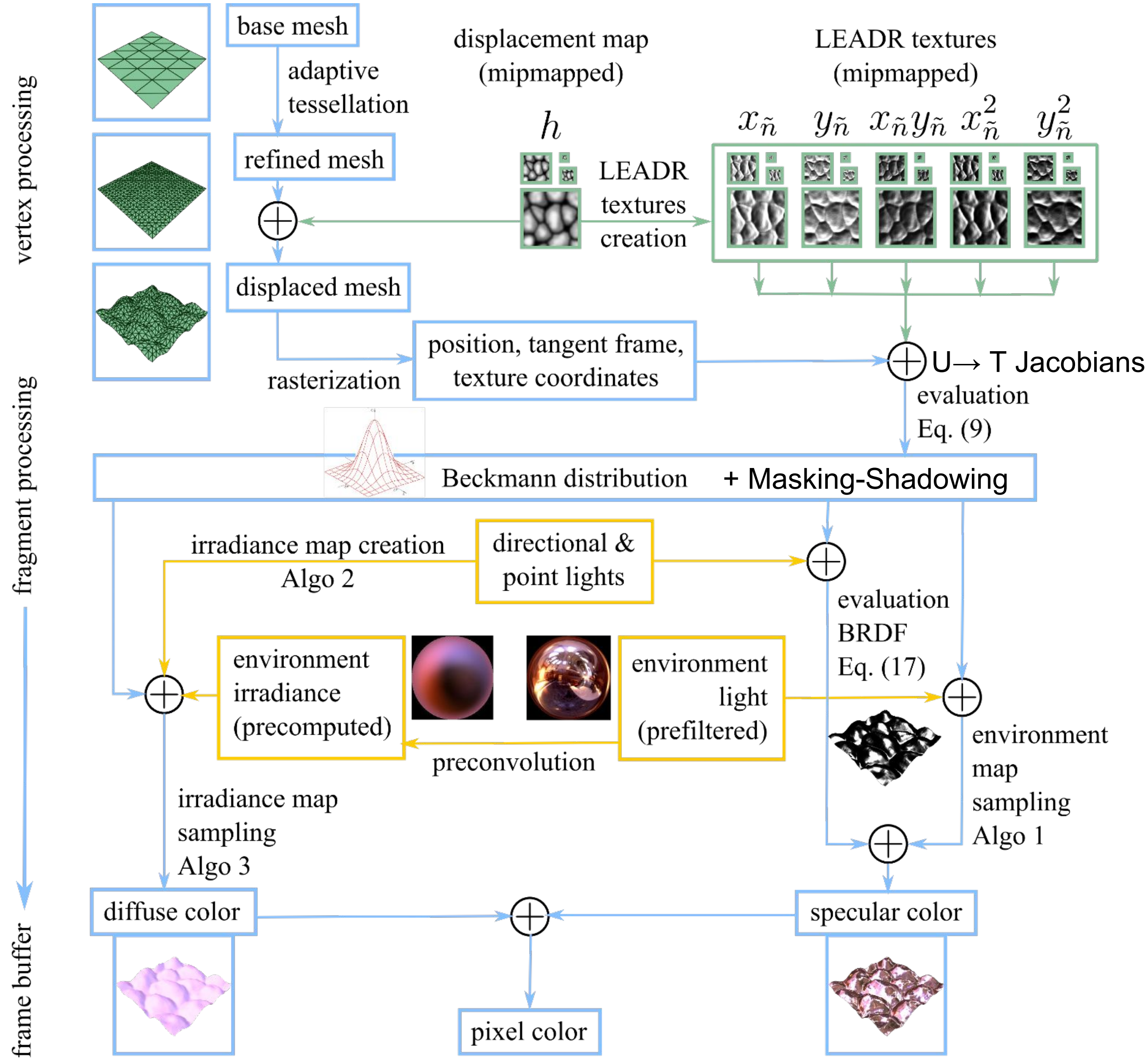


→eval [Bourl 1 : θ>74°)

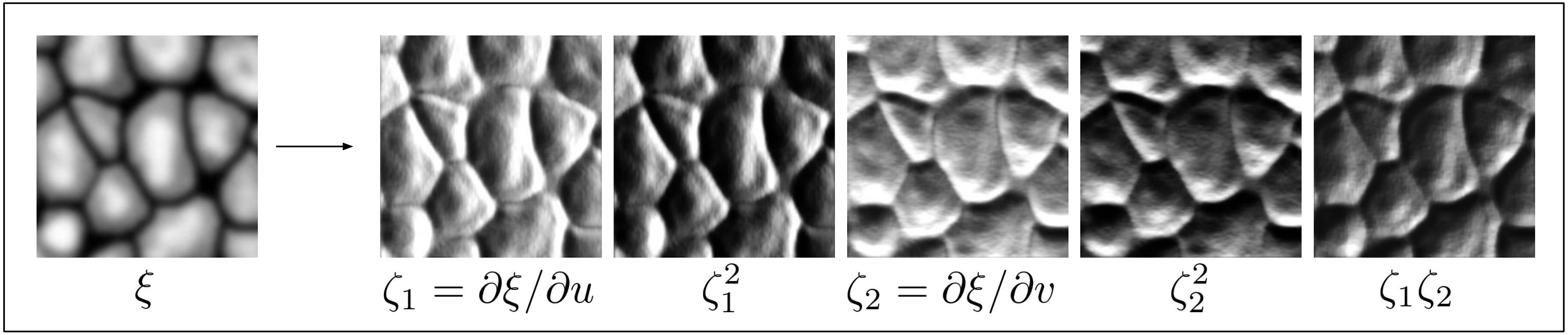
- Separability, Flat macro-surface, MIP-map

Lead-R Pipeline (GPU)

–



LEADR Mapping : U → X



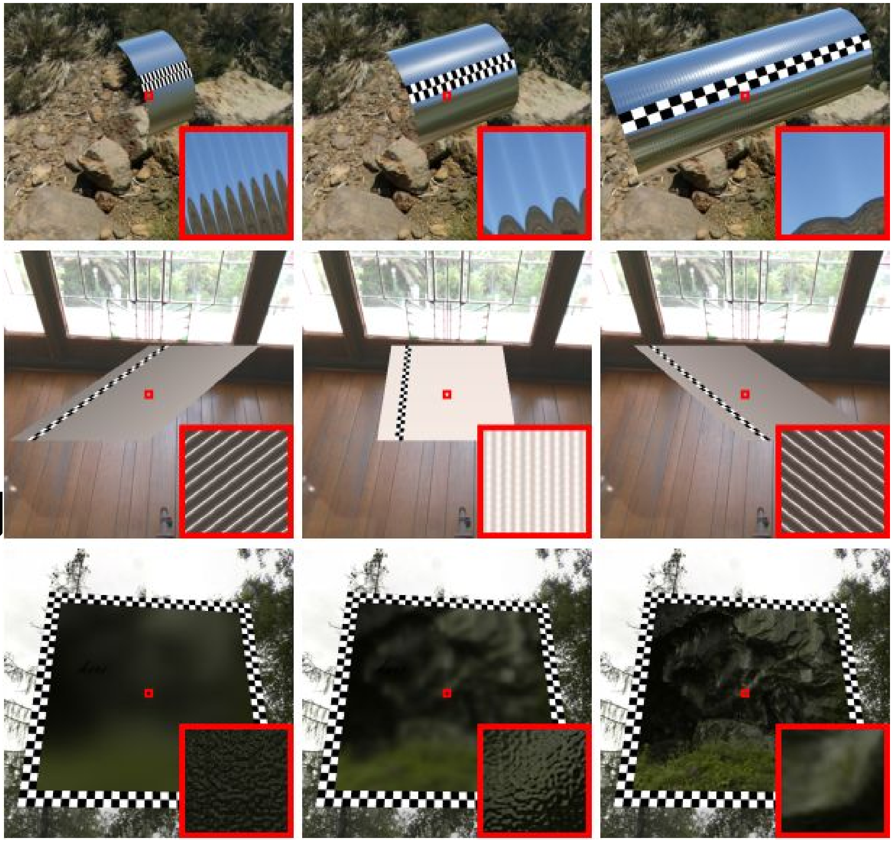
- pixel footprint (usual MIP-map)
- $E(X) = J.E(U)$
- $\Sigma_x = J^t . \Sigma_u . J$

LEADR Map operations

- **Adding* displacements**:
  add LEAN maps

- **Scaling heights** by k:
  scale E(x) by k and $E(x^2)$ by k^2

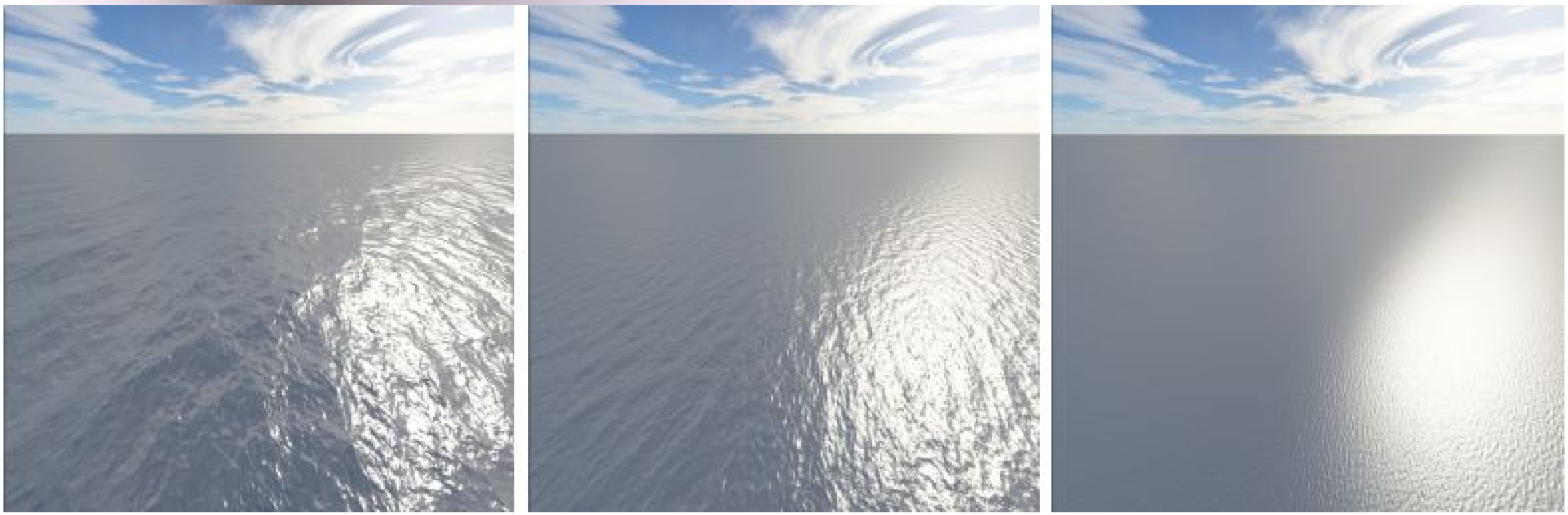
- **Stretching displacement**:
  cf Jacobian: $E_{X'} = J.E_X \; ; \; \Sigma_{x'} = J^t \, . \, \Sigma_X \, . \, J$



Can't: all non-linear operations

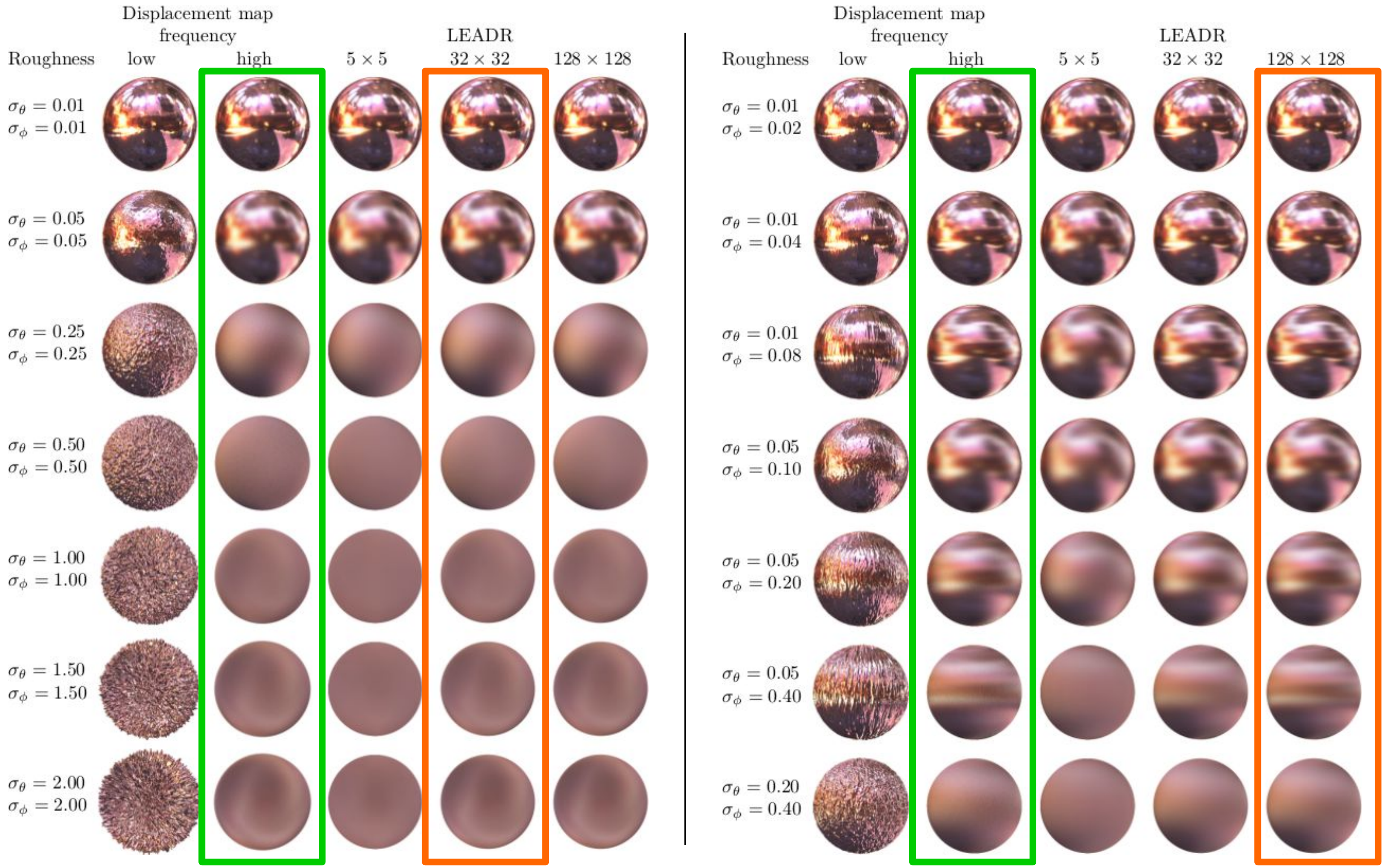
- clamp, sigma
- min, max
- displ correlated with attributes (color, mask, base roughness ...)

Results



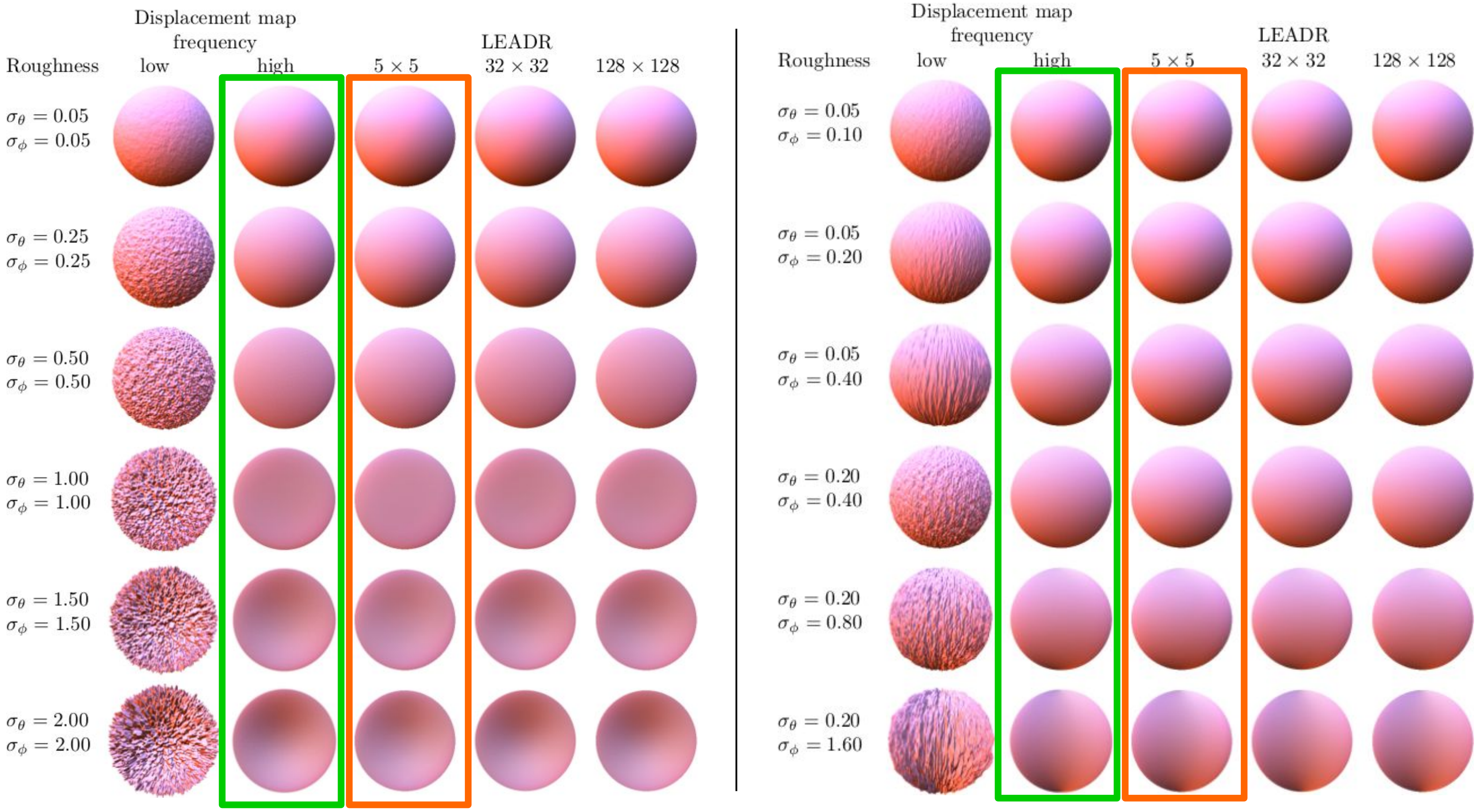
Validation

Fresnel mirror BRDFs



Validation

Diffuse BRDFs



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)

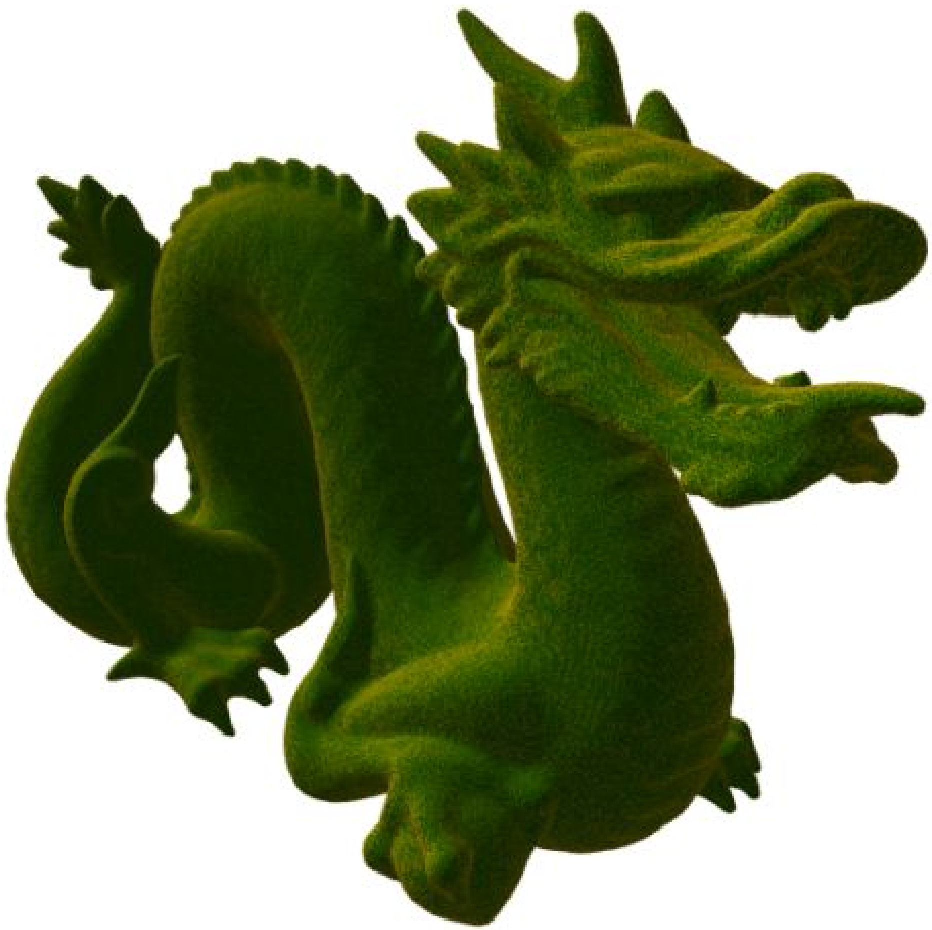
Future Work

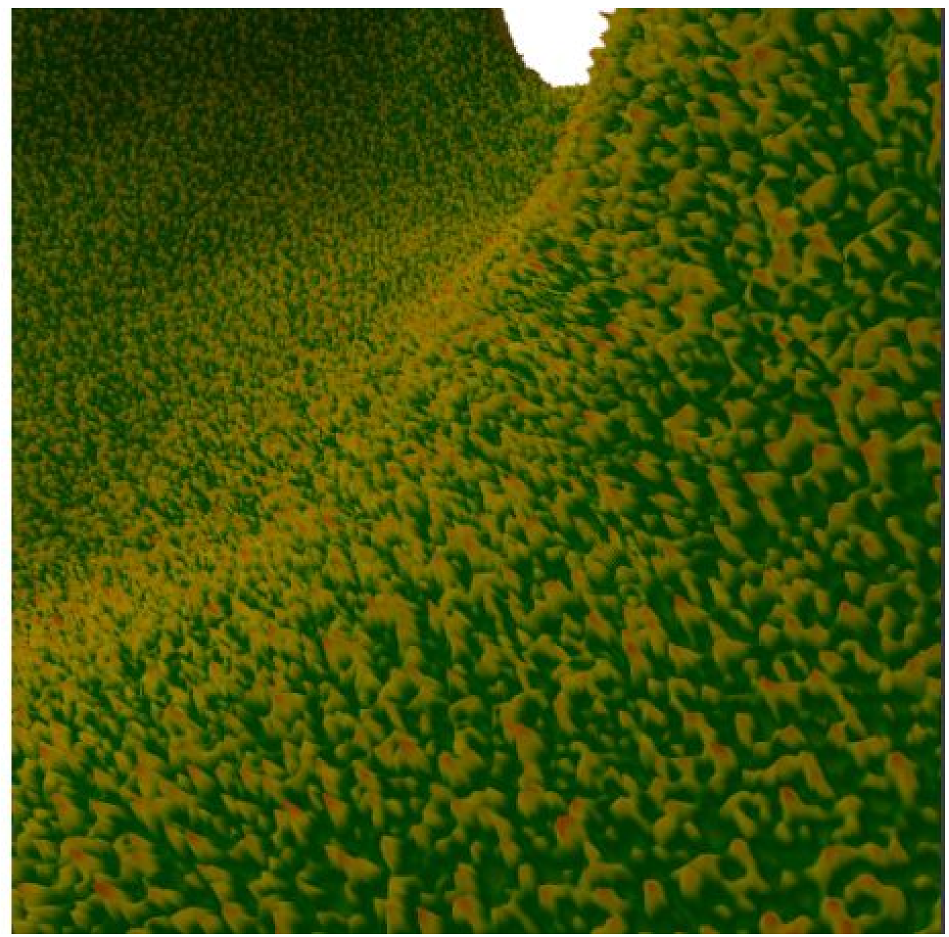
- Multi-lobe, Glints
- Curvature
- 3D displacement
- Angular space → GGX
- Volumes → SGGX
- Correlation with attribs (color)

⟶ *"Filtering Color Mapped Textures and Surfaces"* [I3D'13]

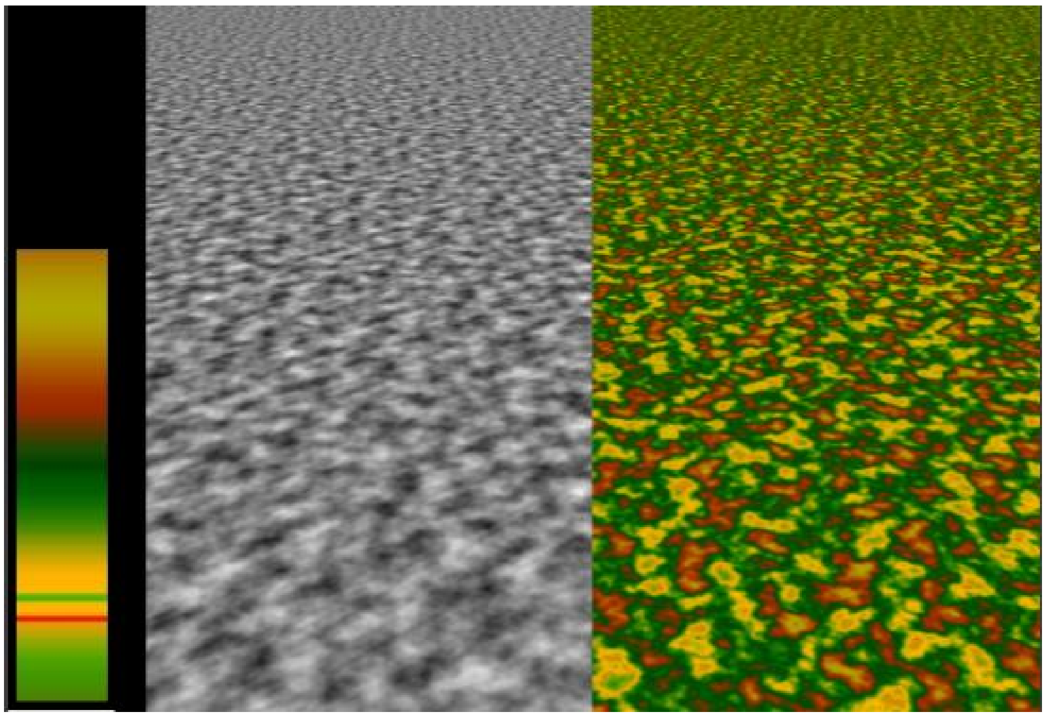
Bonus: filter non-linear & colors

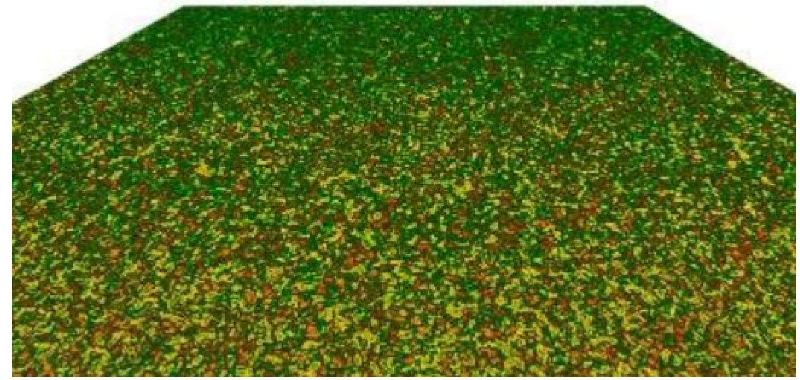
- Displacement – color correlation is important

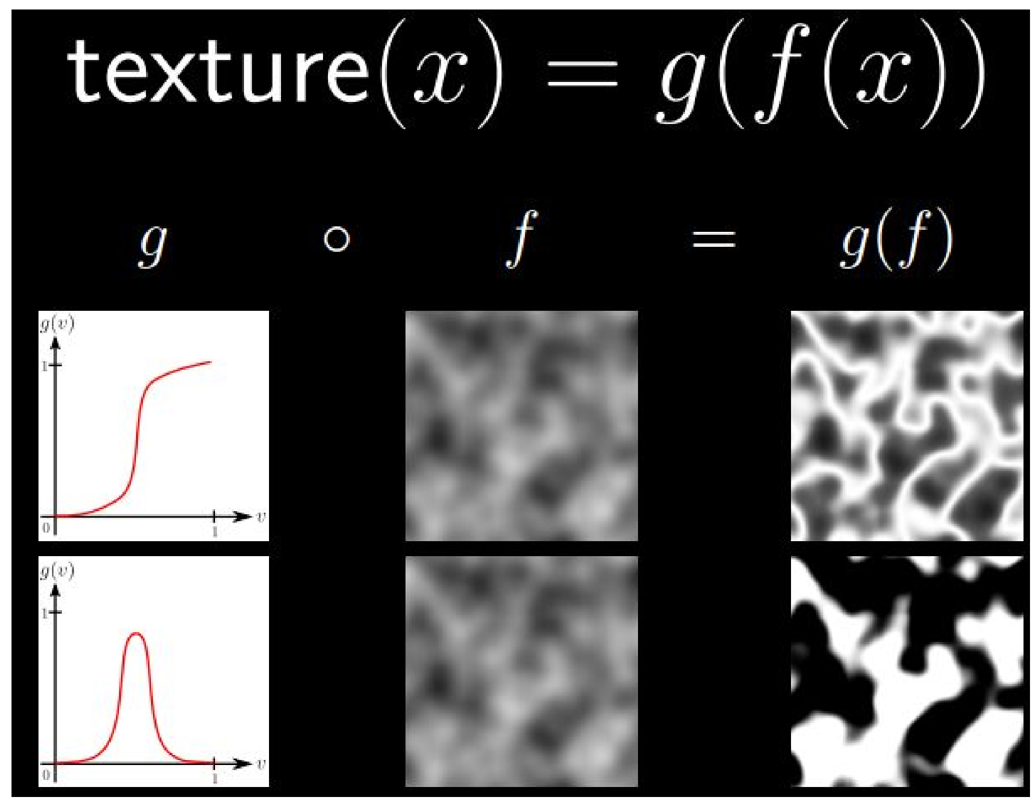




2D filter non-linear & colors

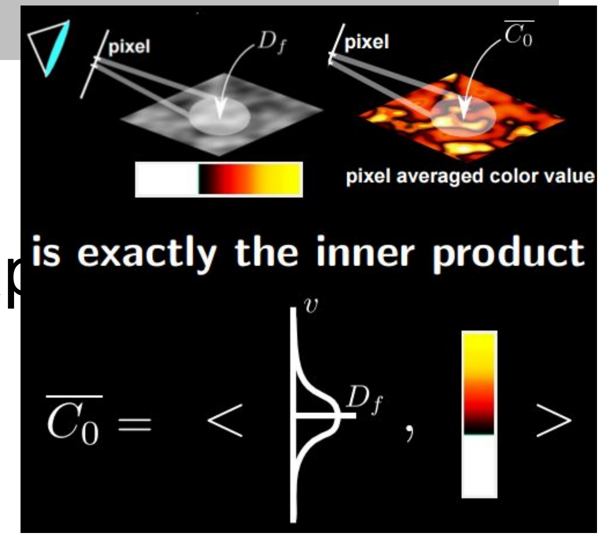


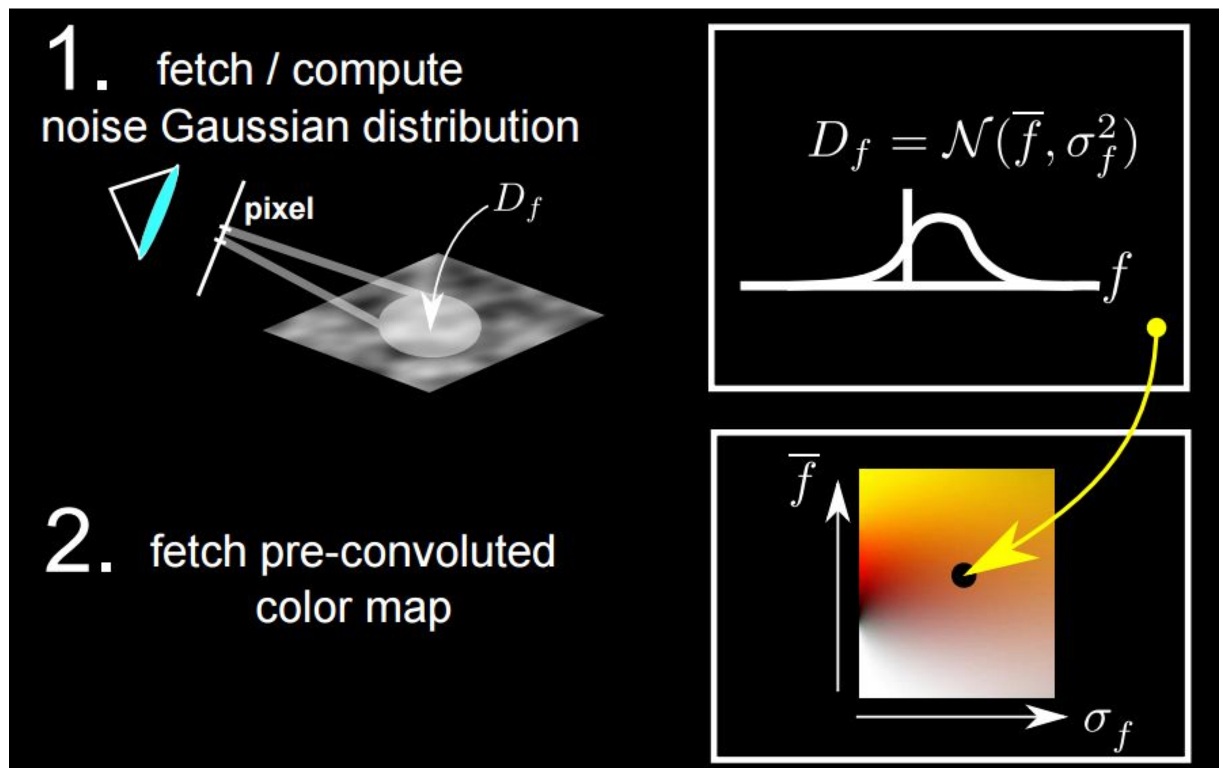




Histogram space + Gaussian (as usual :-))

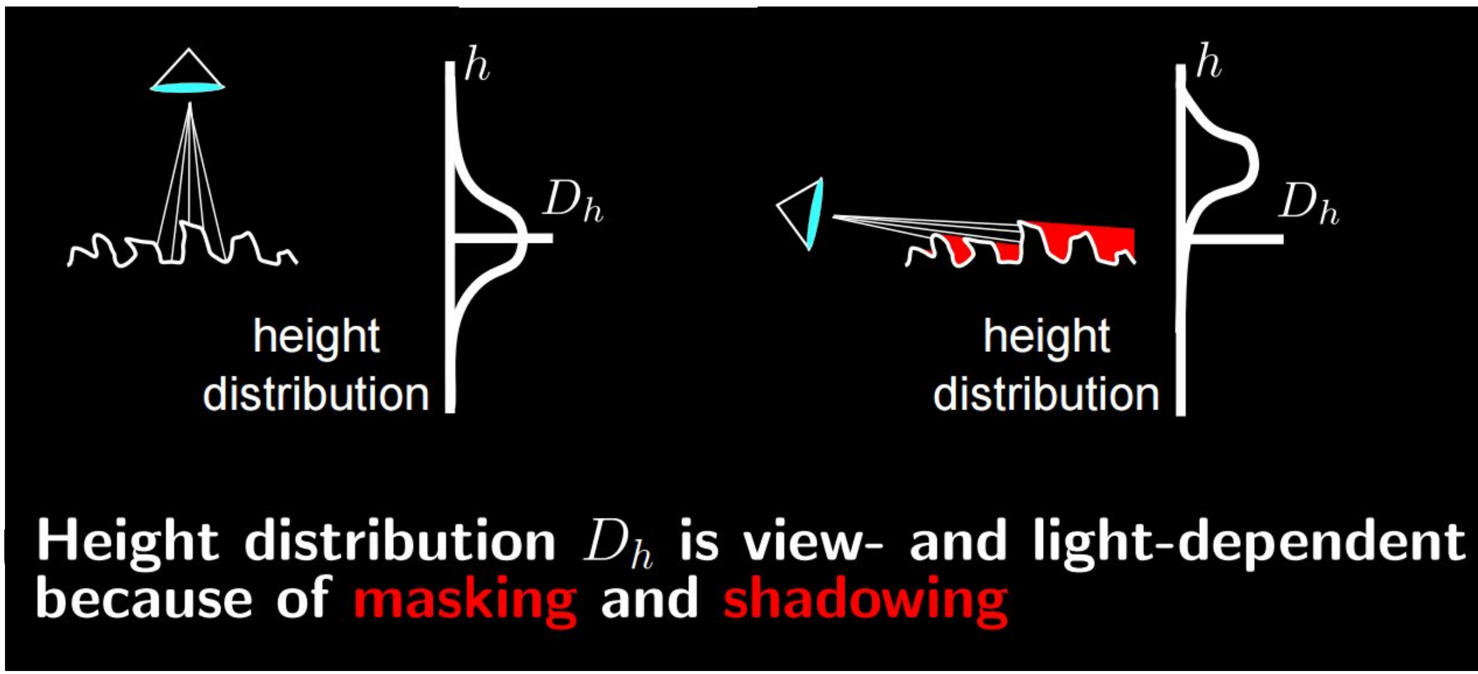
- $\Sigma_{pix} C(T(p)) = \Sigma_{cmap} H(c)C(c) = H.Cmap$
- H(c) ~ Gaussian
- (pre)convolution





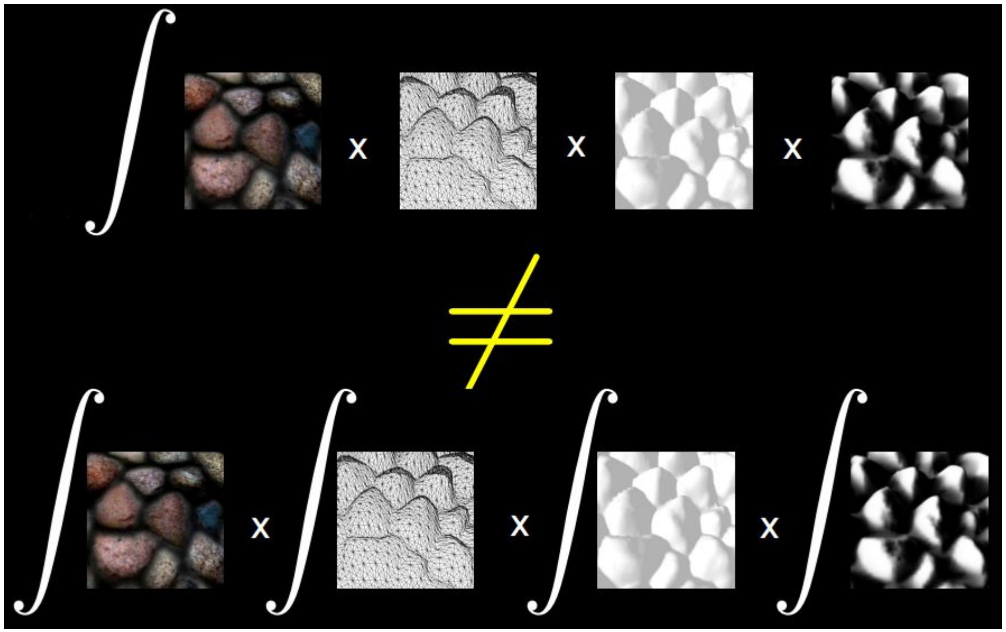
3D filter non-linear & colors

- Gaussian distrib: Lean(h)
- Problem:
-
-
-
-
-

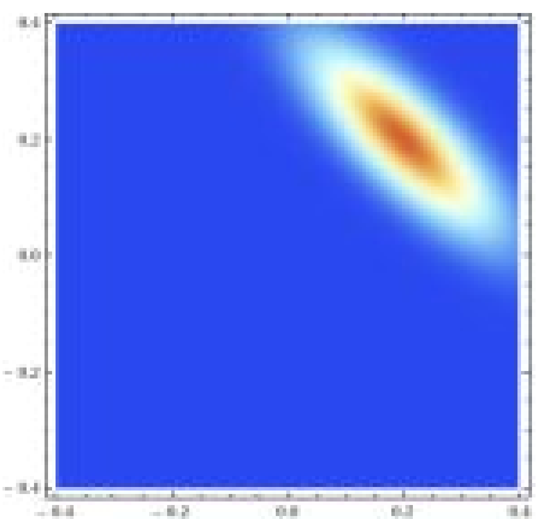


- Sol … al :-))

Take home messages



$$\int C_0(f) \neq C_0\left(\int f\right)$$



$$\int_{x\in S} f(g(x))dx = \int_{histo} f(h)P(h)dh$$

All steps are necessary :-)