Material Appearance Modelling: In-Surface and Subsurface Light Scattering

Jeppe Revall Frisvad

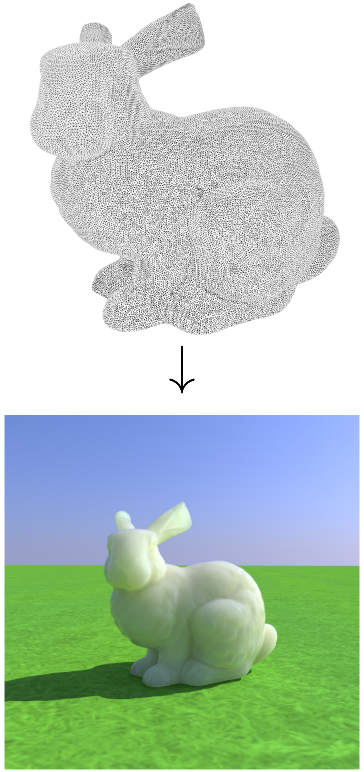
Technical University of Denmark

February 2020

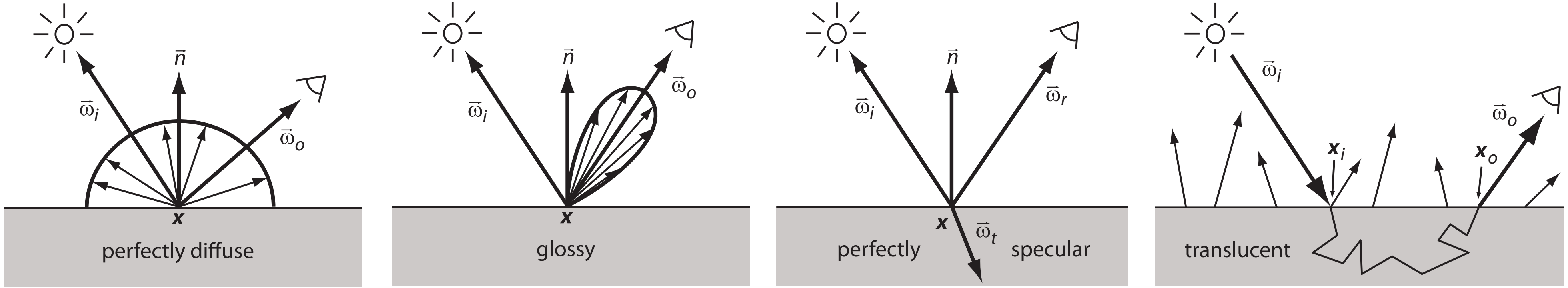
ApPEARS Materials Science Workshop

Material appearance

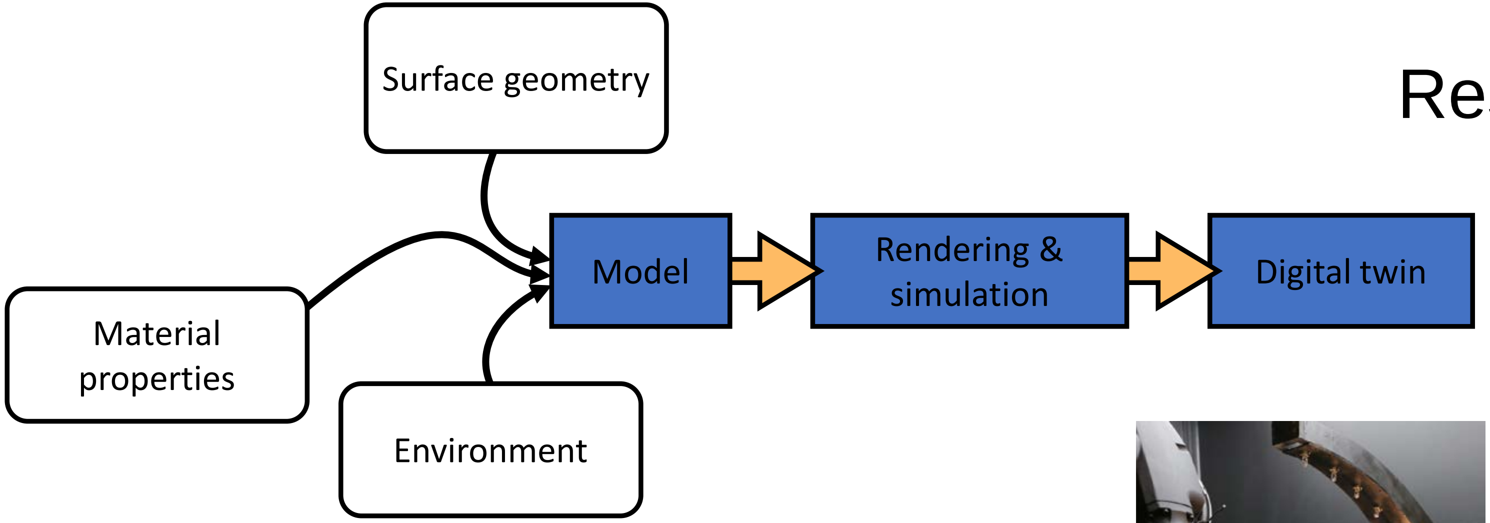
- Light is what you sense.
- Matter is what you see.
- Geometry is an abstraction over the shapes that you see.
- Appearance is a combination of the three.



Reflectance: surface and subsurface scattering of light



Research examples

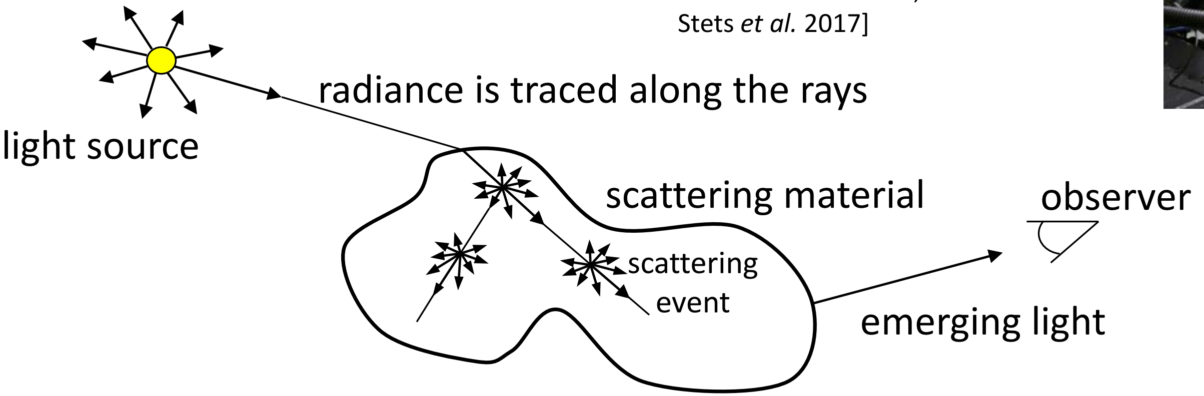


Appearance modelling (scattering theory)

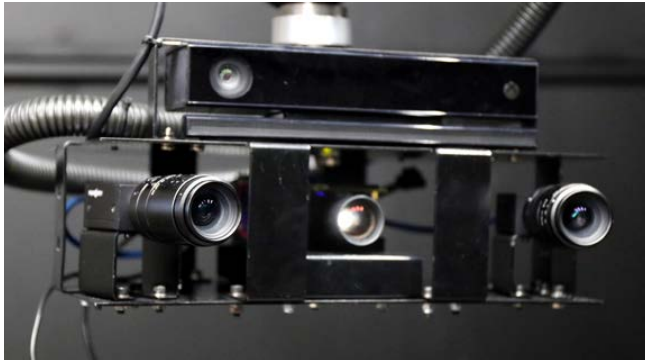
[Frisvad *et al.* 2007; Frisvad 2011; Luongo *et al.* 2017; Frisvad 2018]

Photorealistic rendering (Monte Carlo ray tracing)

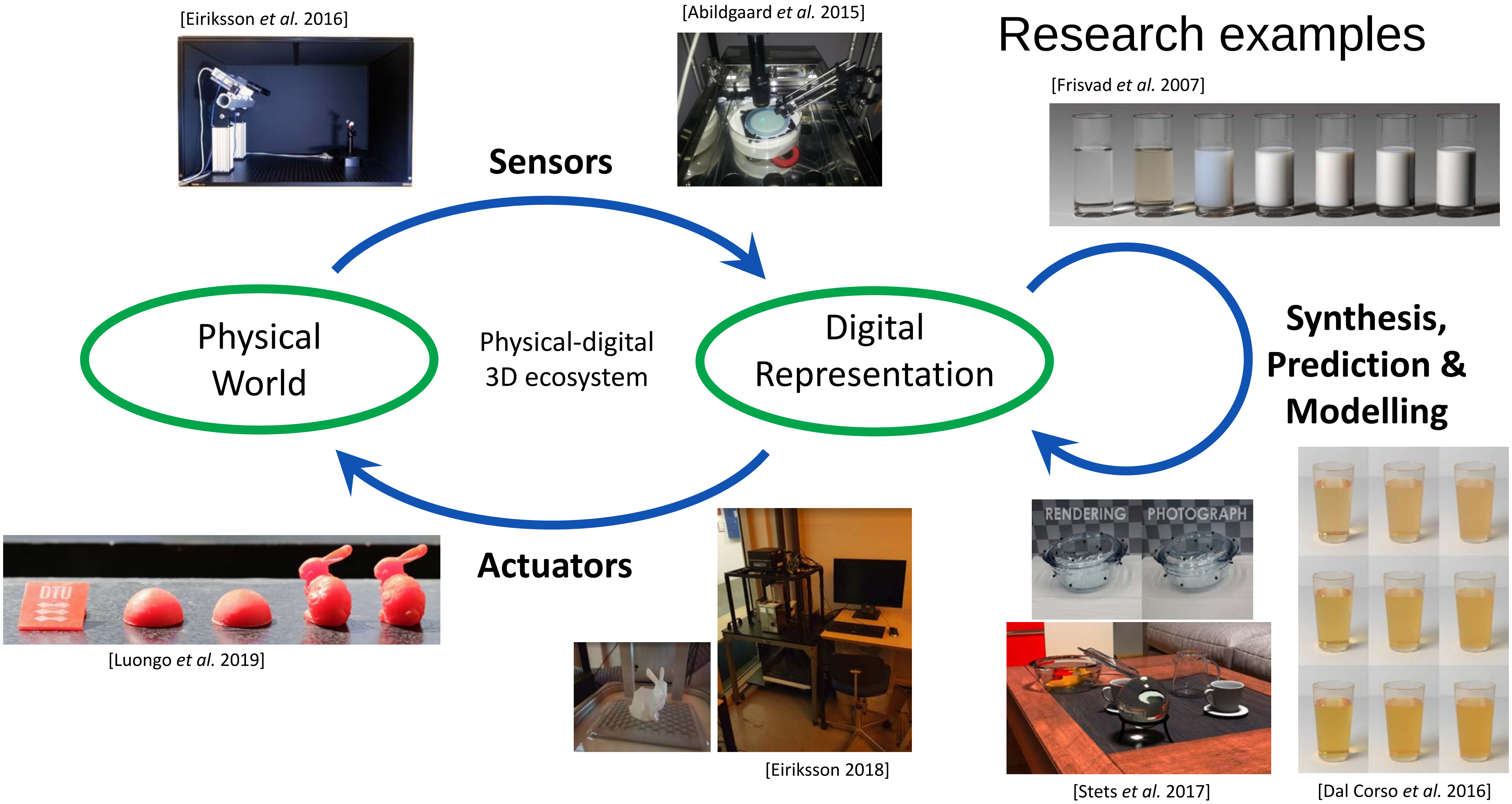
[Frisvad *et al.* 2012; Frisvad *et al.* 2014; Andersen *et al.* 2016; Dal Corso *et al.* 2016; Stets *et al.* 2017]



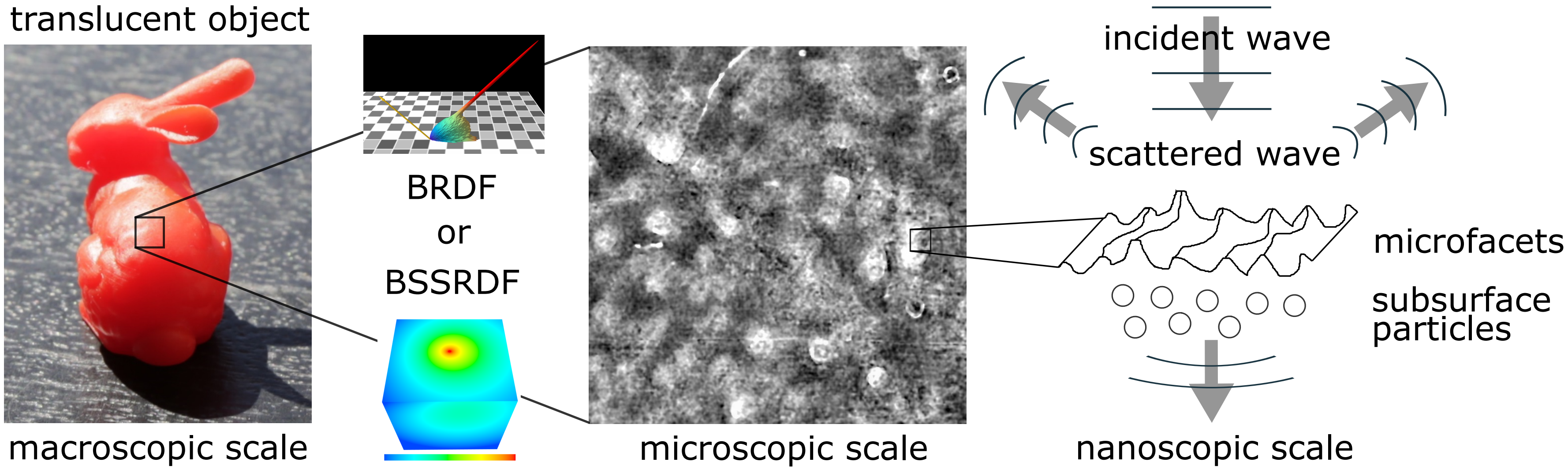
Acquiring model input



(robotic arm, illumination control, structured light scanner)


Research examples
[Eiriksson et al. 2016]
[Abildgaard et al. 2015]
[Frisvad et al. 2007]
Sensors
Physical World
Physical-digital 3D ecosystem
Digital Representation
Synthesis, Prediction & Modelling
Actuators
DTU
RENDERING
PHOTOGRAPH
[Luongo et al. 2019]
[Eiriksson 2018]
[Stets et al. 2017]
[Dal Corso et al. 2016]

Optical properties at multiple scales

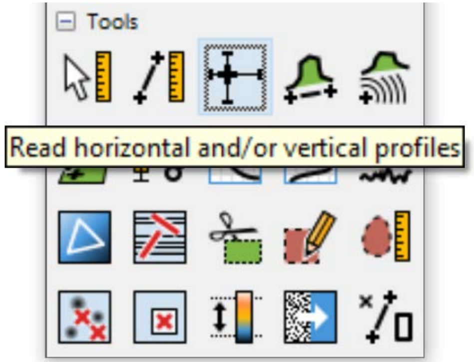


- Appearance matching: manual adjustment of optical properties is nearly impossible.
- We can render objects with a plausible appearance but have a hard time matching the appearance of a manufactured item to that of its digital twin.
- Research challenge: **digital representations of real translucent objects**.
- Important aspects: **validation** (photo-render alignment), **acquisition** (inverse rendering), **application** (quality control).

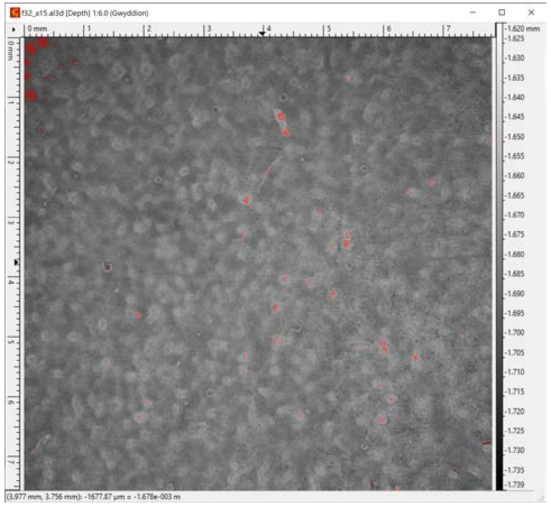
Measuring surface microgeometry

- Alicona Infinite Focus
  - Non-contact
  - Optical
  - Depth by focus-variation
  - Vertical resolution depending on choice of magnification
    - From x2.5: 2300 nm
    - Down to x100: 10 nm
  - Min. measurable roughness:
    - Sa: 3.5 down to 0.015 (arithmetic average height)
  - Output: xxx.al3d file

Inspecting and correcting the data

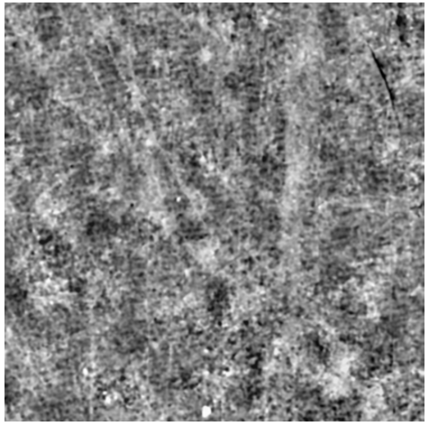


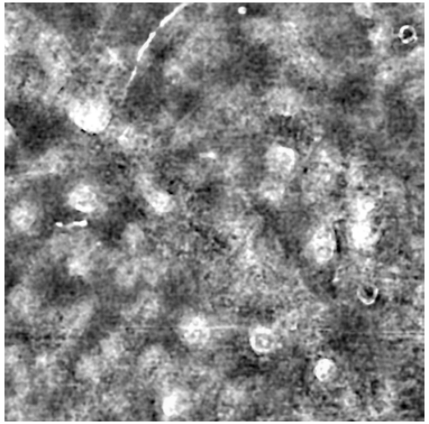
- Free tool for analysis of height fields obtained by microscopy
  Gwyddion: http://gwyddion.net/
- File→Open... (it opens .al3d files)
- Data inspection: Tools – Read horizontal and/or vertical profiles
  Mode: cross / horizontal / vertical (click the image to extract a profile)
- Data correction (outlier removal and adjustment of base):
  - Data Process→Correct Data→Mask of Outliers
    Data Process→Correct Data→Remove Data Under Mask
    Data Process→Mask→Remove Mask
  - Data Process→Level→Flatten Base
- Process: Remove outliers, flatten base, remove outliers
- File→Save as... (store processed data as a .gwy file)

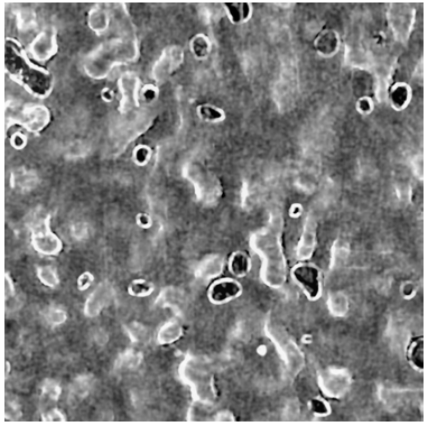


Profilometry and data export

- Cutout: Use the Crop data tool with Create new image checked
- Roughness measurement along a line:
  Use the Calculate roughness parameters tool
- Surface roughness: Use the Statistical quantities tool
- Export using File→Save as...
  Choose an image file type (.png, for example)
  Remove decorations (Value scale, Lateral scale, frame, etc.)



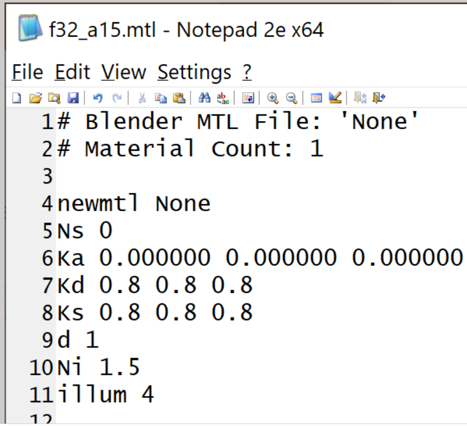




Note the **physical width** of the image and the **physical depth** that the grayscale values correspond to. These are available from the Statistical Quantities tool.

Height map to mesh by displacement mapping

- We can use Blender for this task https://www.blender.org/
- Following a tutorial https://johnflower.org/tutorial/make-mountains-blender-heightmaps
  - Open Blender and delete the default cube (press del)
  - Add→Mesh→Grid (increase X Subdivisions and Y Subdivisions in Add Grid)
  - Import the height map as a texture
  - Apply texture to grid as a displacement map
  - Set Strength to **physical depth** divided by **physical width**
  - Set shading to smooth (right click and select Shade Smooth)
  - File→Export→Wavefront (.obj) [you can uncheck Include UVs]
  - This produces xxx.obj and xxx.mtl
  - Edit xxx.mtl (set illum 4 and Ni 1.5)
  - Ni is the assumed index of refraction

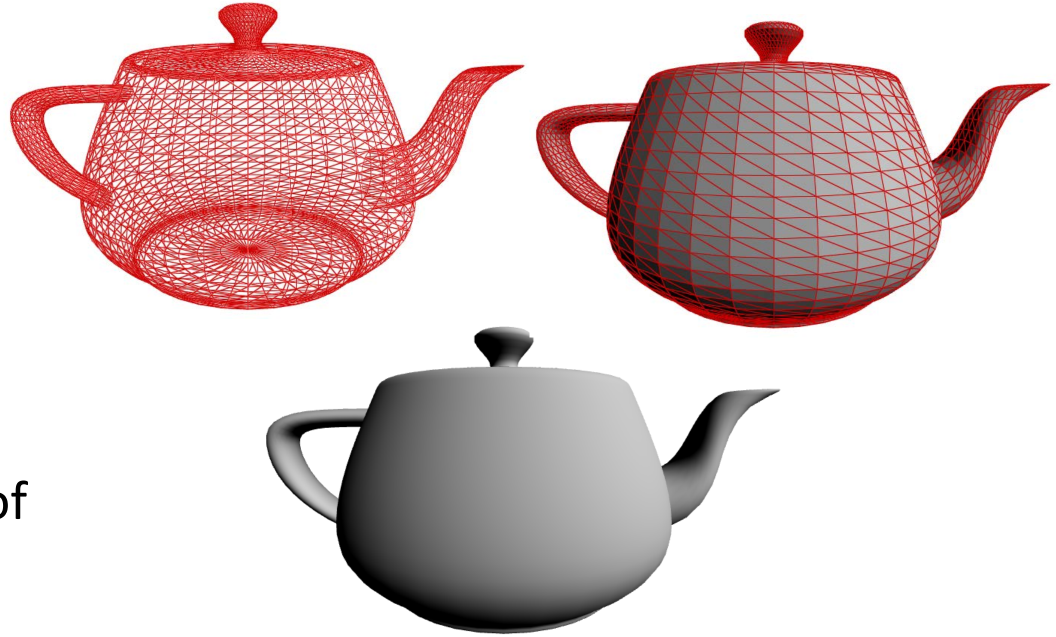


f32_a15.mtl - Notepad 2e x64

File Edit View Settings ?

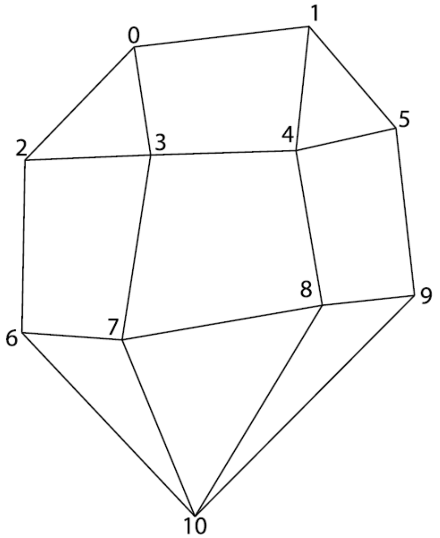
```
# Blender MTL File: 'None'
# Material Count: 1

newmtl None
Ns 0
Ka 0.000000 0.000000 0.000000
Kd 0.8 0.8 0.8
Ks 0.8 0.8 0.8
d 1
Ni 1.5
illum 4
```

What is a mesh?



- Surface geometry is often modeled by a collection of triangles, where some of them share edges (a triangle mesh).
- Triangles provide a discrete representation of an arbitrary surface. See teapot example.

- The **indexed face set** is a popular data representation of polygon meshes.
- Any polygon mesh can be converted to a triangle mesh.
- A .obj file contains indexed face sets.

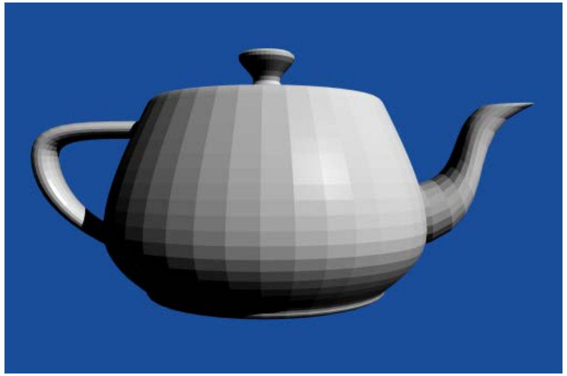


VERTICES
0: (-0.2, 1.5, 0)
1: (1.3, 1.7, 0)
2: (-1.1, 0.4, 0)
3: (0.0, 0.45, 1)
4: (1.1, 0.5, 1.2)
5: (2.1, 0.75, 0.2)
6: (-1.2, -1,0.01)
7: (-0.3, -1.2,2)
8: (1.3,-0.9, 3)
9: (2.0 -0.8,1.2)
10: (0.4, -2.1, -1.1)

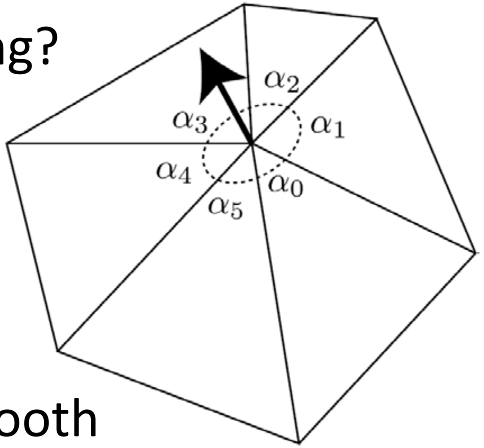
FACES
0: 0,2,3
1: 0,3,4,1
2: 1,4,5
3: 2,6,7,3
4: 3,7,8,4
5: 4,8,9,5
6: 6,10,7
7: 7,10,8
8: 8,10,9

Figure from Bærentzen et al. Guide to Computational Geometry Processing, Springer, 2012.

What is then a smooth mesh?

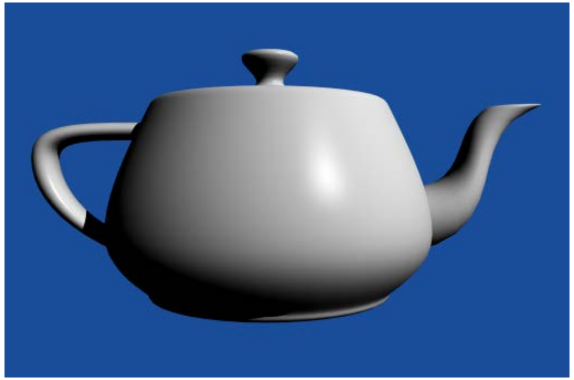


- Triangles are flat. Their **geometric normal**s lead to flat shading.
- How do we make the object smooth? Interpolated per-vertex lighting?
- What is the normal in a vertex?
  The angle-weighted pseudo-normal is a good choice.



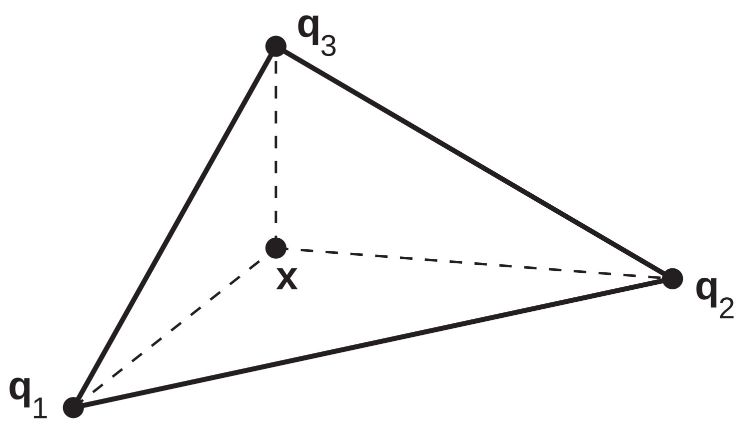
$$\vec{n} = \frac{\sum \alpha_i \vec{n}_i}{\|\sum \alpha_i \vec{n}_i\|}$$

- Another indexed face set is created for the vertex normals.
- Interpolation of the vertex normals across each triangle leads to smooth shading.
- The interpolated normal is called the **shading normal**.



Linear interpolation across triangles

- A point $\boldsymbol{x}$ in a triangle is given by a weighted average of the triangle vertices $(\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3)$:



$$\boldsymbol{x} = \alpha \boldsymbol{q}_1 + \beta \boldsymbol{q}_2 + \gamma \boldsymbol{q}_3\,, \qquad \alpha + \beta + \gamma = 1$$

- The weights (α, β, γ) are the barycentric coordinates.
- The point is in the triangle if $\alpha, \beta, \gamma \in [0;\ 1]$. That is,

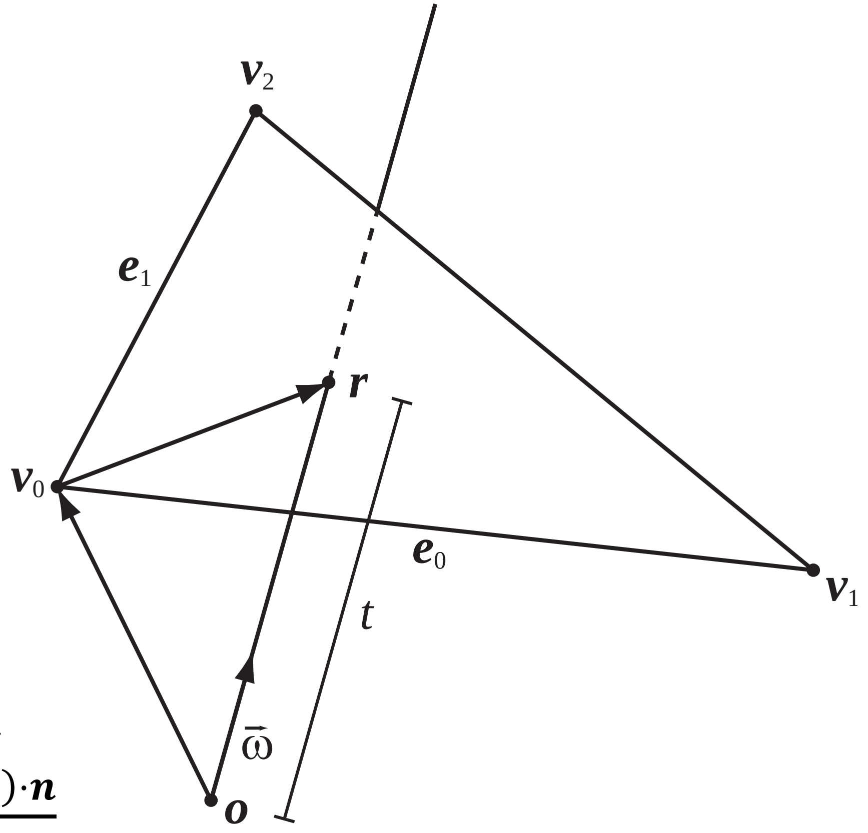
$$\alpha \geq 0 \quad \text{and} \quad \beta \geq 0 \quad \text{and} \quad \alpha + \beta \leq 1$$

- Replace the triangle vertices $(\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3)$ by vertex normals and normalize to get the interpolated normal.

Ray-triangle intersection

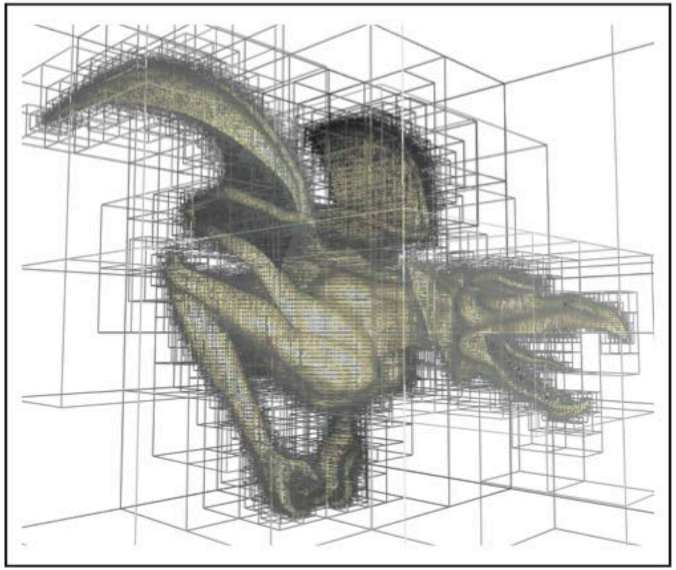
- Ray: $\boldsymbol{r}(t) = \boldsymbol{o} + t\vec{\omega}, \quad t \in [t_{\min}, t_{\max}]$
- Triangle: $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2$
- Edges and geometric normal:
  $\boldsymbol{e}_0 = \boldsymbol{v}_1 - \boldsymbol{v}_0, \boldsymbol{e}_1 = \boldsymbol{v}_0 - \boldsymbol{v}_2, \boldsymbol{n} = \boldsymbol{e}_0 \times \boldsymbol{e}_1$
- Barycentric coordinates:
  $\boldsymbol{r}(\alpha, \beta, \gamma) = \alpha\boldsymbol{v}_0 + \beta\boldsymbol{v}_1 + \gamma\boldsymbol{v}_2 = \boldsymbol{v}_0 + \beta\boldsymbol{e}_0 - \gamma\boldsymbol{e}_1$
- The ray intersects the triangle's plane at $t' = \frac{(\boldsymbol{v}_0 - \boldsymbol{o})\cdot\boldsymbol{n}}{\vec{\omega}\cdot\boldsymbol{n}}$
- Find $\boldsymbol{r}(t') - \boldsymbol{v}_0$ and decompose it into portions along the edges $\boldsymbol{e}_0$ and $\boldsymbol{e}_1$ to get β and γ. Then check

$$\alpha \geq 0 \ \text{ and } \ \beta \geq 0 \ \text{ and } \ \alpha + \beta \leq 1$$



Spatial subdivision

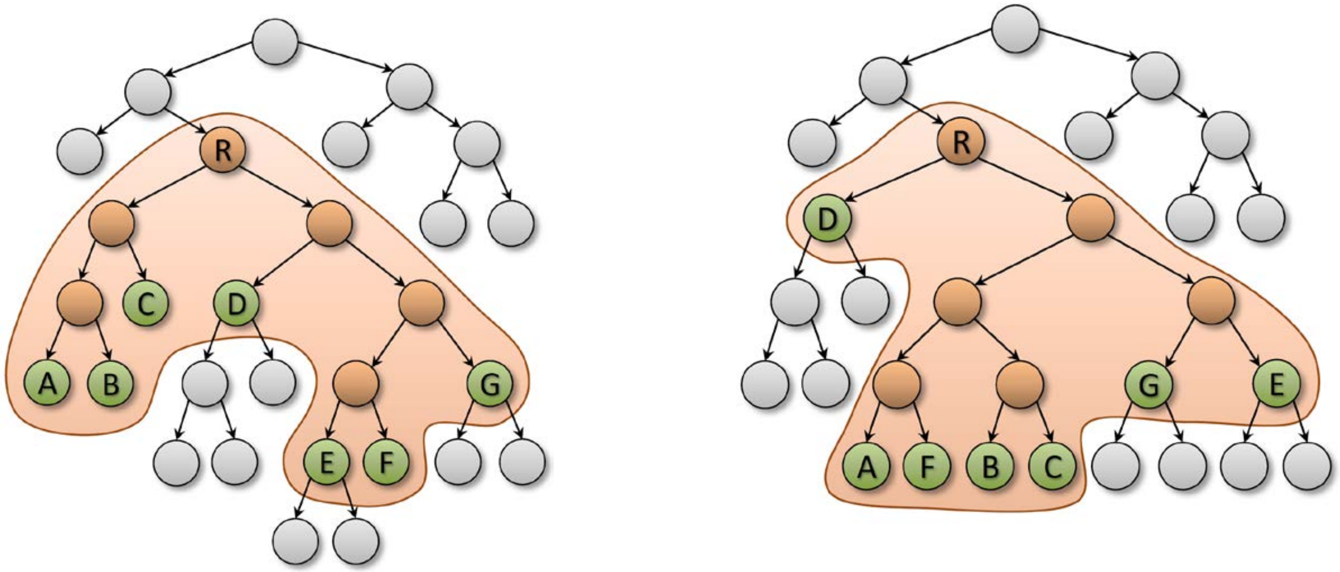
- To model arbitrary geometry with triangles, we need many triangles.
- A million triangles and a million pixels are common numbers.
- Testing all triangles for all pixels requires 10^{12} ray-triangle intersection tests.
- If we do a million tests per millisecond, it still takes more than 15 minutes.
- This is prohibitive. We need to find the relevant triangles.
- Spatial data structures offer logarithmic complexity instead of linear.
- A million tests become twenty operations $\log_2 10^6 \approx 20$
- 15 minutes become 20 milliseconds.



Gargoyle embedded in oct tree [Hughes et al. 2014]

Treelet restructuring bounding volume hierarchy

treelet



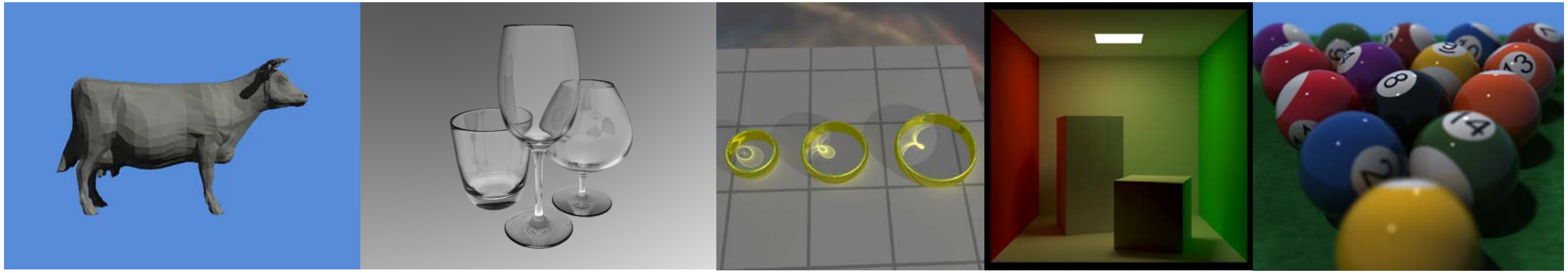
reorganized treelet

- Practical GPU-based bounding volume hierarchy (BVH) builder.
  1. Build a low-quality BVH (parallel linear BVH).
  2. Optimize node topology by parallel treelet restructuring (keeping leaves and their subtrees intact).
  3. Post-process for fast traversal.

Reference:
Karras, T., and Aila, T. Fast parallel construction of high-quality bounding volume hierarchies. In *Proceedings of HPG 2013*, pp. 89-99. ACM, July 2013.

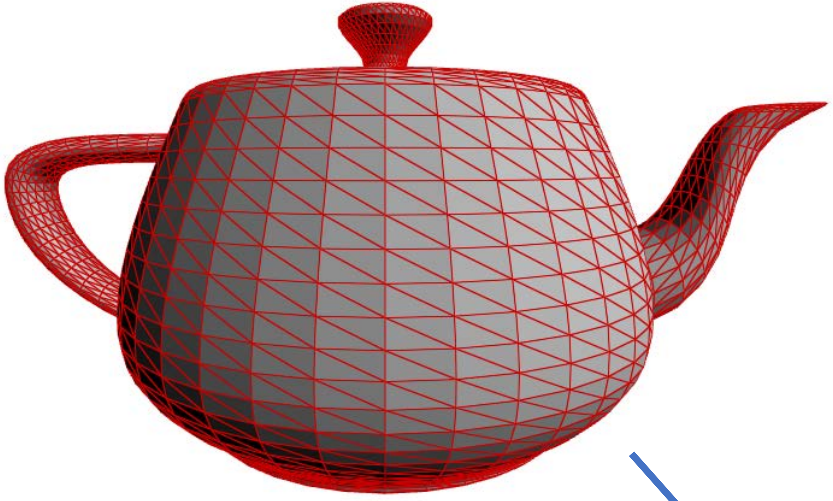
NVIDIA OptiX https://developer.nvidia.com/optix

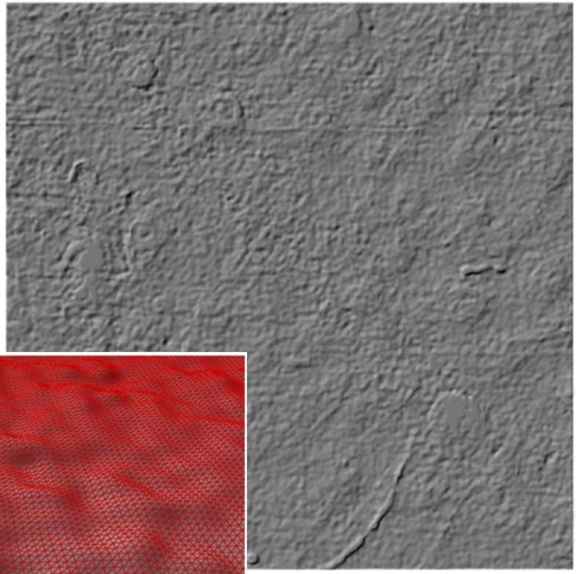
- Interactive ray tracing demos: cow (sample 6), glass, PPM, PT, Cook.

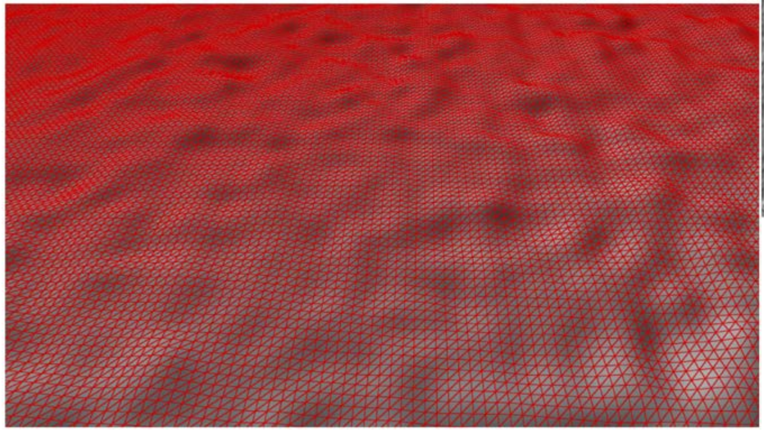


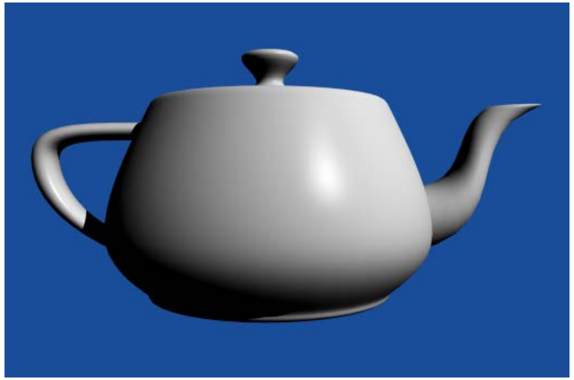
Reference:
Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., and Stich, M. OptiX: a general purpose ray tracing engine. *ACM Transactions on Graphics (SIGGRAPH 2010)* **29**(4):66, July 2010.

Two meshes – how to combine?







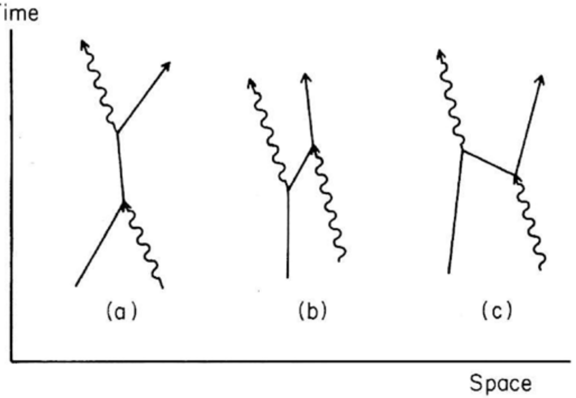


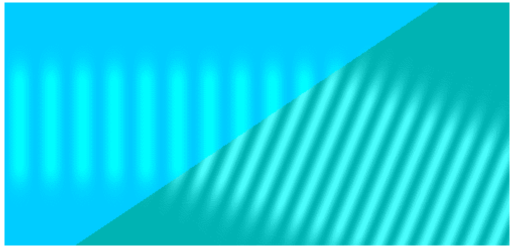
Multiscale modeling of surface geometry

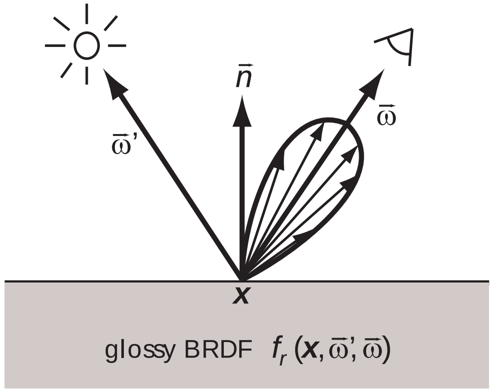
- Smallest scale: everything is quantum particles. Matter is electrons going from place to place (ignore nuclei).
  Light-matter interaction: A photon interacts with an electron.

- Microscopic scale: surfaces but no roughness. All details are defined.
  Light-matter interaction: interaction of electromagnetic waves with surfaces.

- Macroscopic scale: objects with a macrosurface and material specification (roughness/absorption) mapped onto them.
  Light-matter interaction: bidirectional (scattering) distribution functions for rays of light.







Simulation to go from micro to macro

- Take the plane wave solution for Maxwell's equations.
- The (complex) index of refraction n is a quantity summarizing the microscopic material properties (permittivity, permeability, conductivity).
- Consider an electromagnetic plane wave incident on a surface between two half-space media (of refractive indices n_i and n_t).
- By requiring continuity across the interface, we can derive:
  - The **law of reflection** (direction of reflected light)
  - The **law of refraction** (direction of transmitted light)
  - **Fresnel's equations** for reflection (amount of reflection vs. transmitted light)
- Neglecting wavelength (assume $\lambda \to 0$), we can trace rays along the direction of energy propagation in the waves (along Poynting's vector).
- Given surface microgeometry, we can use such **ray tracing** to compute **bidirectional scattering distribution functions** (BSDFs).

Ray tracing specular surfaces

- Fresnel's equations for reflection (R is reflected, $T = 1 - R$ is transmitted)

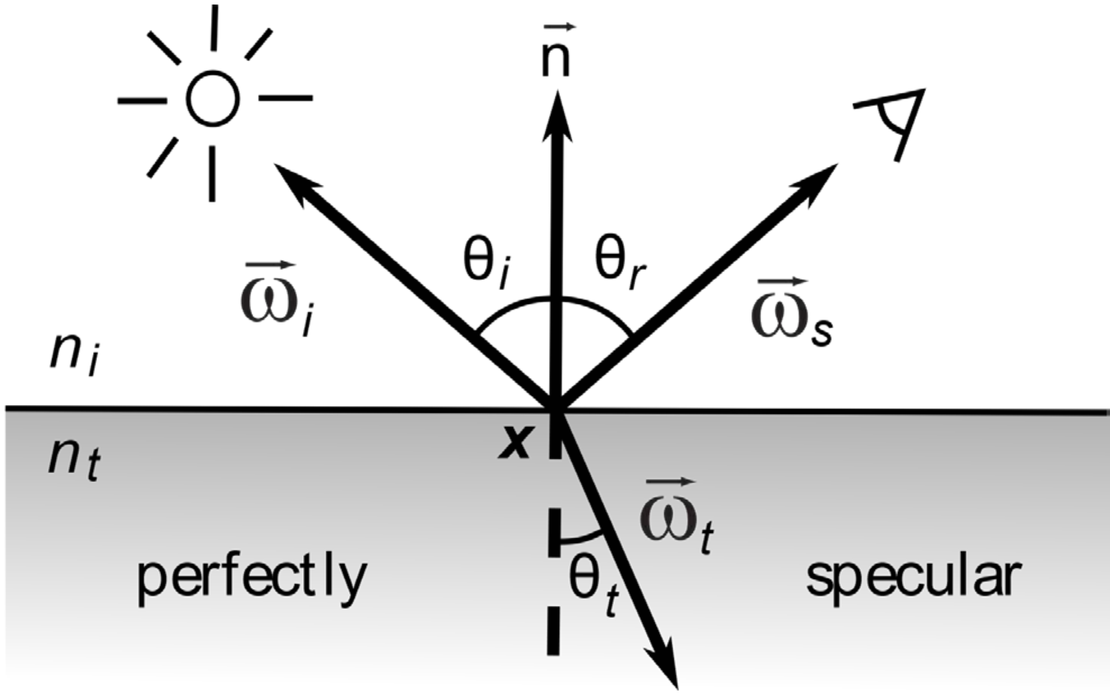
$$\tilde{r}_\perp = \frac{n_i \cos\theta_i - n_t \cos\theta_t}{n_i \cos\theta_i + n_t \cos\theta_t}, \qquad \tilde{r}_\parallel = \frac{n_t \cos\theta_i - n_i \cos\theta_t}{n_t \cos\theta_i + n_i \cos\theta_t}, \qquad R = \frac{1}{2}\left(|\tilde{r}_\perp|^2 + |\tilde{r}_\parallel|^2\right)$$

- The law of refraction

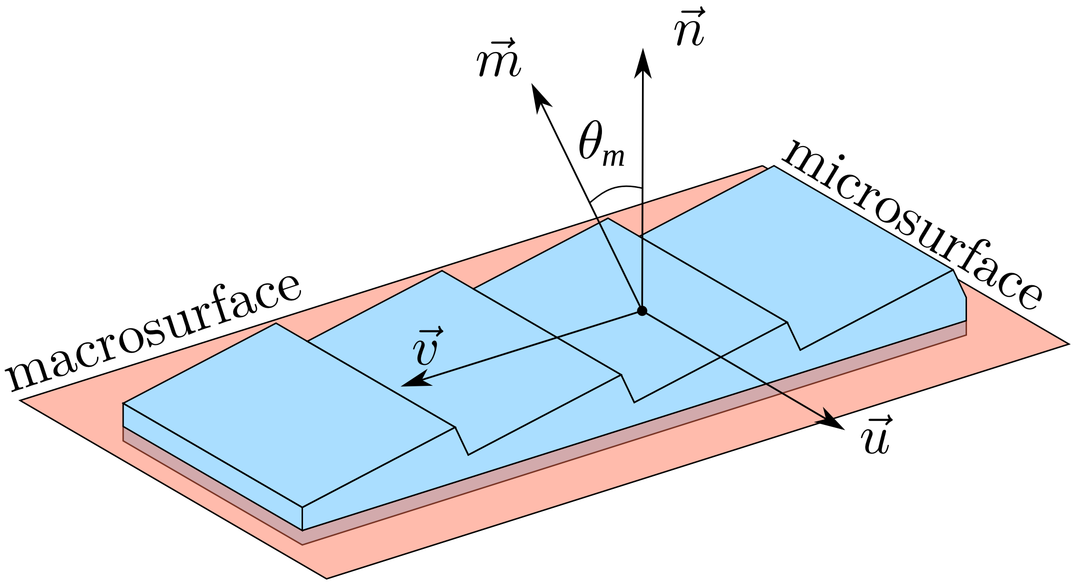
$$n_t \sin\theta_t = n_i \sin\theta_i \;\Rightarrow \cos\theta_t = \sqrt{1 - \left(\frac{n_i}{n_t}\right)^2 \sin^2\theta_i}$$

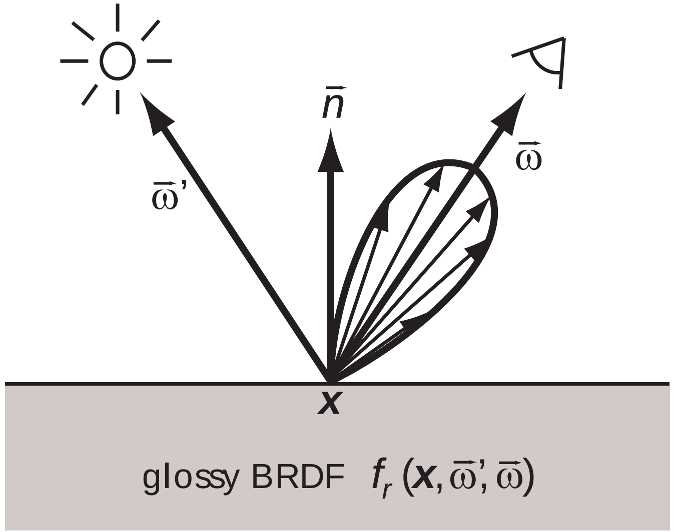
- Directions of reflected and refracted light

$$\vec{\omega}_s = 2(\vec{\omega}_i \cdot \vec{n})\vec{n} - \vec{\omega}_i$$
$$\vec{\omega}_t = \frac{n_i}{n_t}\left((\vec{\omega}_i \cdot \vec{n})\vec{n} - \vec{\omega}_i\right) - \vec{n}\cos\theta_t$$



Macro and micro surface





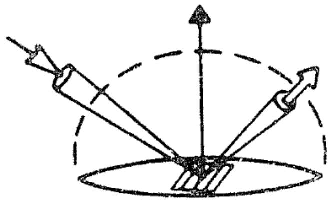
- The BSDF defines the ratio of light incident in a surface point $\boldsymbol{x}$ from a direction $\vec{\omega}'$ that scatters into another direction $\vec{\omega}$.
- The BRDF f_r is the reflectance part of the BSDF.

The microsurface defines the geometry of the differential area $\mathrm{d}A$ at the position $\boldsymbol{x}$ on the macrosurface.

Bidirectional Reflectance Distribution Function

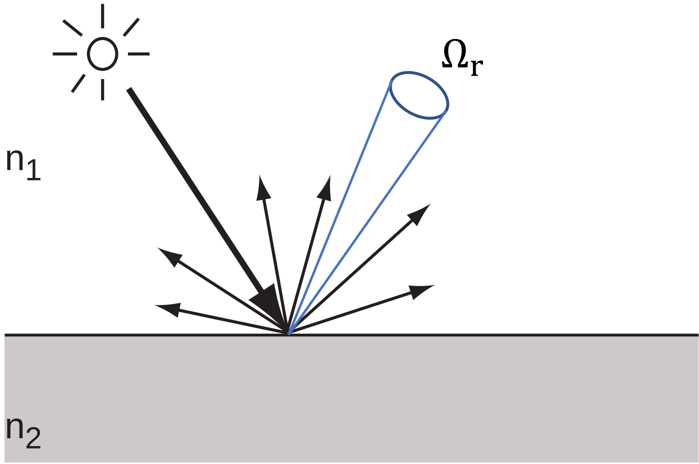
- The definition of the BRDF: $f_r(\boldsymbol{x}, \vec{\omega}_i, \vec{\omega}_r) = \frac{\mathrm{d}L_r(\boldsymbol{x}, \vec{\omega}_r)}{\mathrm{d}E(\boldsymbol{x}, \vec{\omega}_i)}$

  The ratio of an element of reflected radiance $\mathrm{d}L_r$ to an element of irradiance $\mathrm{d}E$.



- An element of irradiance due to incident radiance within a differential element of solid angle $\mathrm{d}\omega_i$:

$$\mathrm{d}E(\boldsymbol{x}, \vec{\omega}_i) = L_i(\boldsymbol{x}, \vec{\omega}_i) \cos\theta_i \, \mathrm{d}\omega_i \,, \qquad L_r = \int_{2\pi} f_r \, L_i \cos\theta_i \, \mathrm{d}\omega_i$$



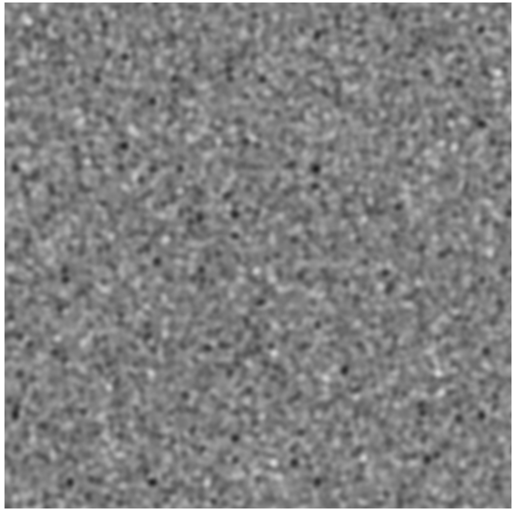
- Radiance is radiant flux per projected area per solid angle:

$$L = \frac{\mathrm{d}^2\Phi}{\cos\theta \, \mathrm{d}A \, d\omega}, \qquad \Phi_r = \int_A \int_{\Omega_\mathrm{r}} L_r \cos\theta_r \, \mathrm{d}\omega_\mathrm{r} \, \mathrm{d}A$$

- Radiant flux Φ is a measurable quantity. So, we can set up a measurement equation for reflected radiant flux Φ_r, where A is the microsurface.
- We can then **evaluate a BRDF by solving the measurement equation**.

Noise

- Noise explorer:
  https://people.compute.dtu.dk/jerf/code/noise/



$$0.5\,(A\ \mathrm{noise}(B\,\boldsymbol{p}) + 1)$$
$$A = 1, B = 1$$

- Sparse convolution noise
  - Convolution of randomly placed ($\boldsymbol{x}_{i,j}$) impulses of random value ($\alpha_{i,j}$).
  - Use a filter kernel with compact support and insert a regular grid (cell vertices $\boldsymbol{q}_i$).

$$\mathrm{cubic}(\boldsymbol{v}) = \begin{cases} (1 - 4\,\boldsymbol{v}\cdot\boldsymbol{v})^3 & \text{for } v\cdot v < \dfrac{1}{4} \\ 0 & \text{otherwise} \end{cases}$$

  - Use a seeded RNG: $\mathrm{rnd}(t)$
  - Choose a number of impulses per cell (N).

$$\mathrm{noise}(\boldsymbol{p}) = \frac{4}{5\sqrt[3]{N}} \sum_{i=0}^{7} \sum_{j=1}^{N} \alpha_{i,j}\,\mathrm{cubic}(\boldsymbol{x}_{i,j} - \boldsymbol{p})$$

$$\boldsymbol{x}_{i,j} = \boldsymbol{q}_i + \boldsymbol{\xi}_{i,j}$$

$$\alpha_{i,j} = \mathrm{rnd}\left(t_{n_{i,j}}\right)(1 - 2(j \bmod 2))$$

$$\boldsymbol{\xi}_{i,j} = \left(\mathrm{rnd}\left(t_{n_{i,j}+1}\right), \mathrm{rnd}\left(t_{n_{i,j}+2}\right), \mathrm{rnd}\left(t_{n_{i,j}+3}\right)\right)$$

$$n_{i,j} = 4(N\boldsymbol{q}_i \cdot \boldsymbol{a} + j)$$

$$\boldsymbol{q}_i = \left\lfloor \boldsymbol{p} - \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) \right\rfloor + \left(i \bmod 2, \left\lfloor \frac{i}{2} \right\rfloor \bmod 2, \left\lfloor \frac{i}{4} \right\rfloor \bmod 2\right)$$

My choice: $N = 30$ and $\boldsymbol{a} = (1, 1000, 576)$

Noise-based modeling

- Noise with octaves
  - Number of octaves $\Omega \geq 1$
  - Lacunarity $\ell > 1$
  - Fractional increment (roughness) $H \in (0,1]$

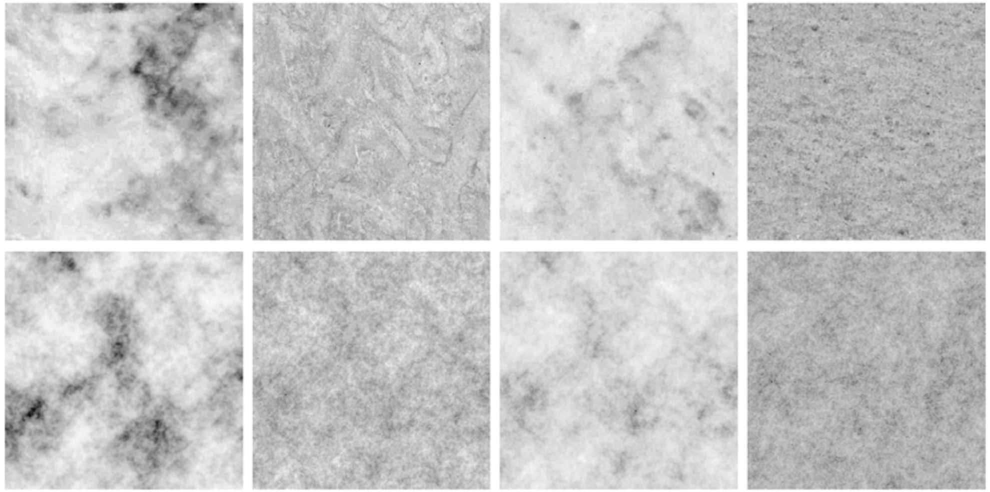
$$\mathrm{fBm}(\boldsymbol{p}) = \sum_{i=0}^{\Omega-1} \ell^{-Hi} \mathrm{noise}(\boldsymbol{p}\, \ell^{i})$$

  - $H = 1$ is a monofractal
    (same fractal dimension everywhere)

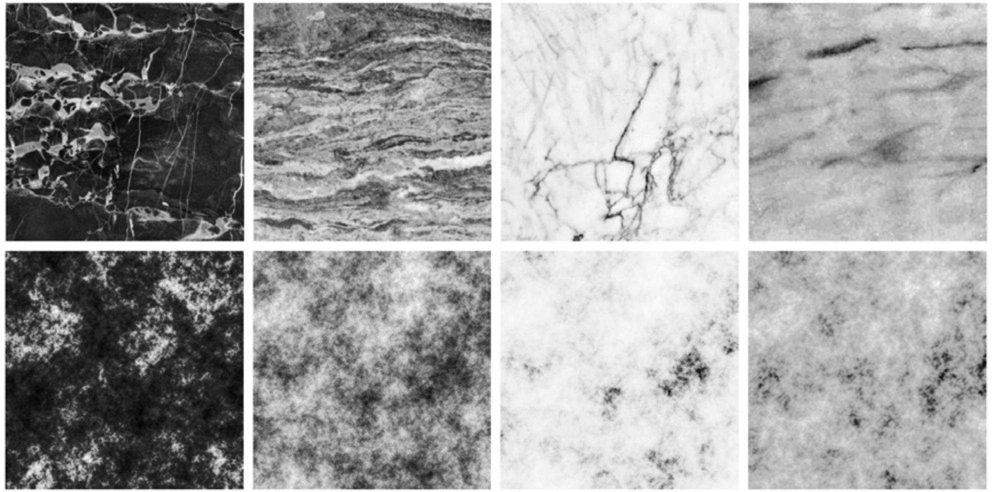
- Absolute value for sharp edges

$$\mathrm{turbulence}(\boldsymbol{p}) = \sum_{i=\Omega_{\mathrm{low}}}^{\Omega_{\mathrm{high}}} \frac{1}{2^{i}} \left|\mathrm{noise}(2^{i}\, \boldsymbol{p})\right|$$

Top rows: input from marble images



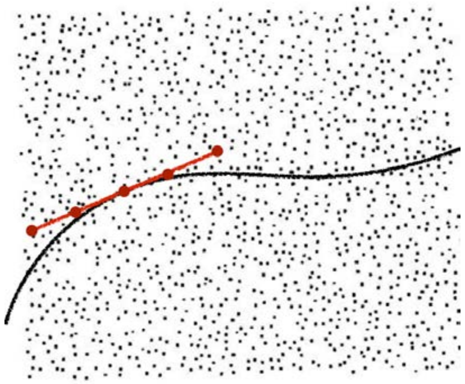
Kim Harder Fog. Noise-based texture synthesis by analysis of image examples. MSc thesis, Technical University of Denmark, 2017.



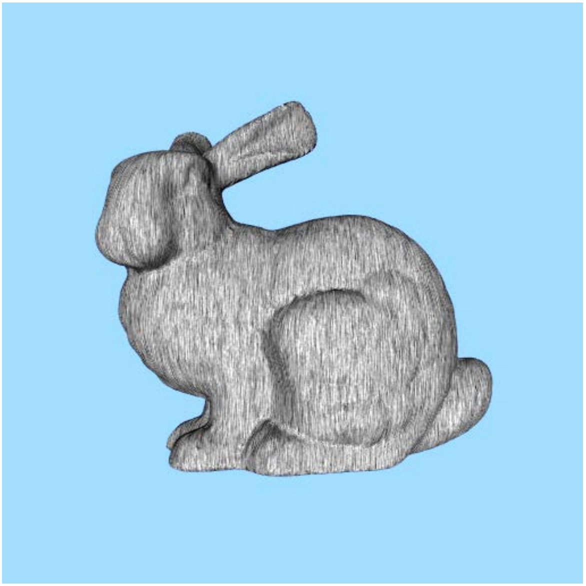
Bottom rows: fBm fit

Line integral convolution

- Given a 3D noise function like sparse convolution noise:
  - For each pixel take the integral along a line.

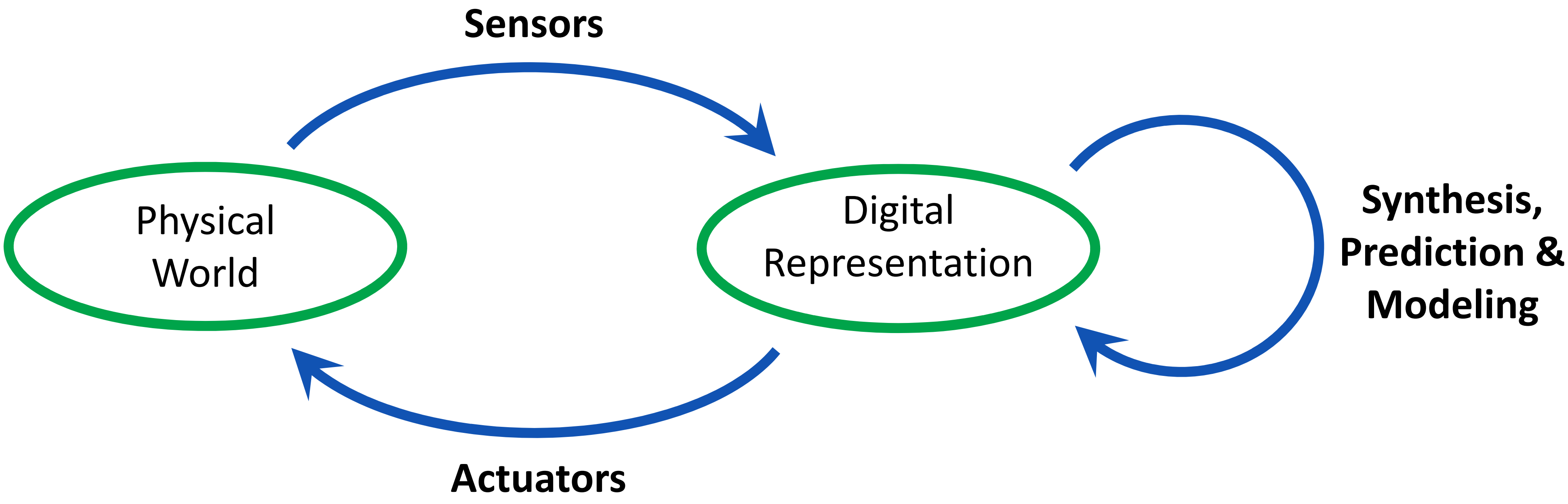


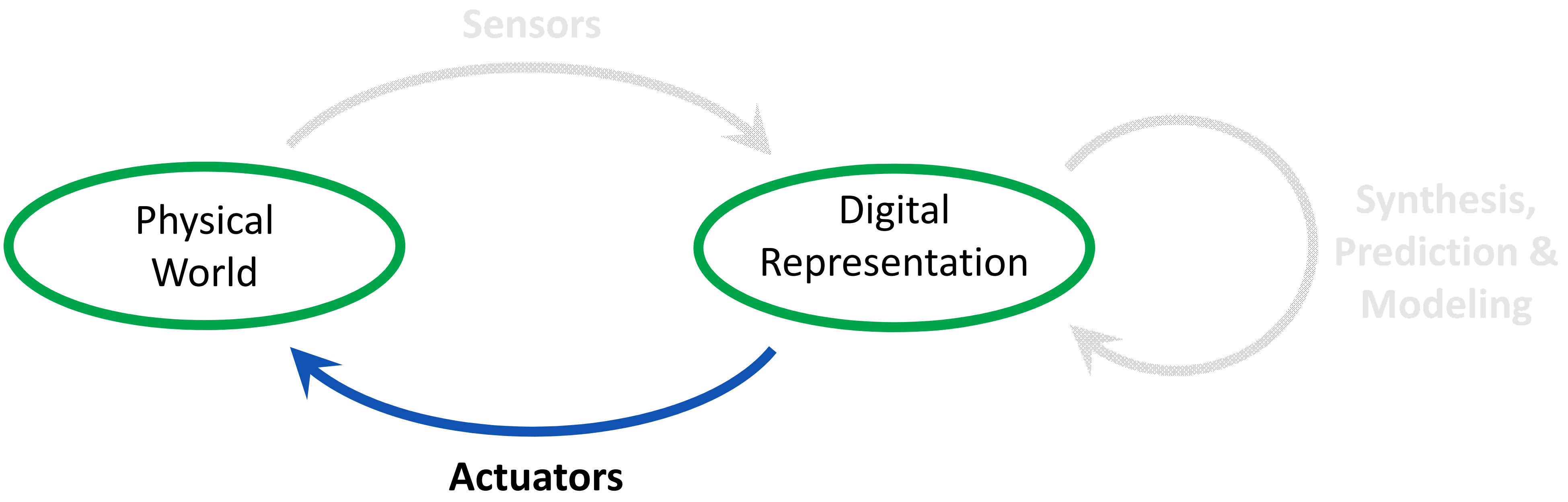
+



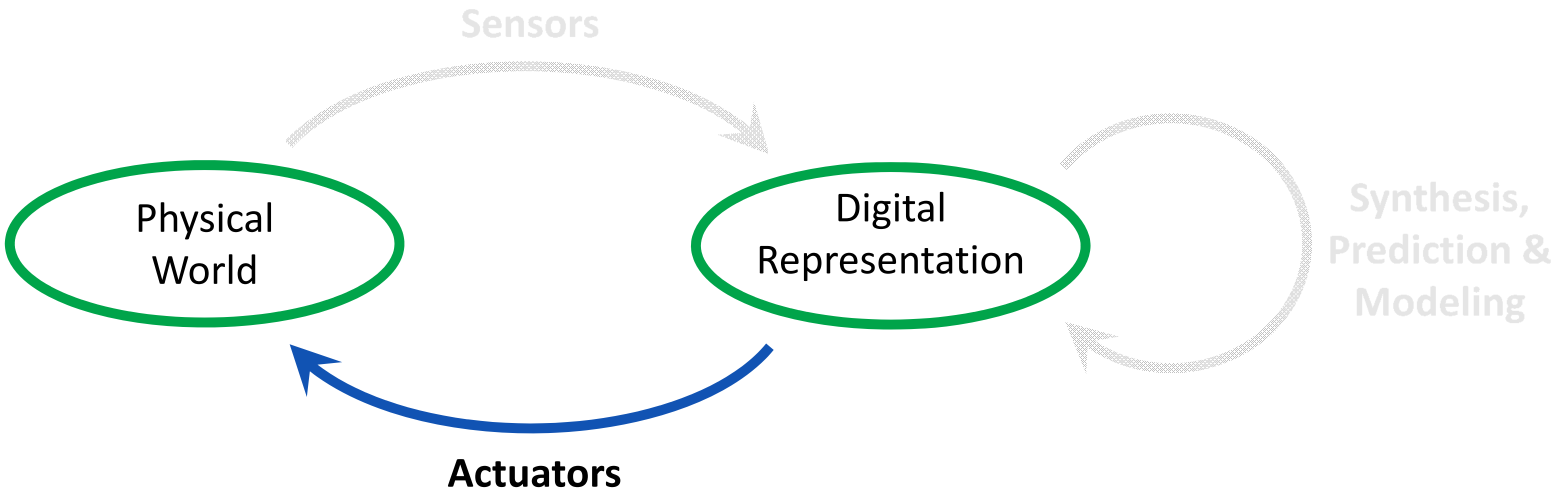
- Obtain a tangent space of your 3D surface to be printed.
  https://people.compute.dtu.dk/jerf/code/hairy/

References:
- Battke, H., Stalling, D., Hege H.-C. Fast line integral convolution for arbitrary surfaces in 3D. In *Visualization and Mathematics*, pp. 181-195. Springer, 1997.
- Frisvad, J. R. Building an orthonormal basis from a 3d unit vector without normalization. *Journal of Graphics Tools* **16**(3):151-159, August 2012.


Sensors
Physical World
Digital Representation
Synthesis, Prediction & Modeling
Actuators

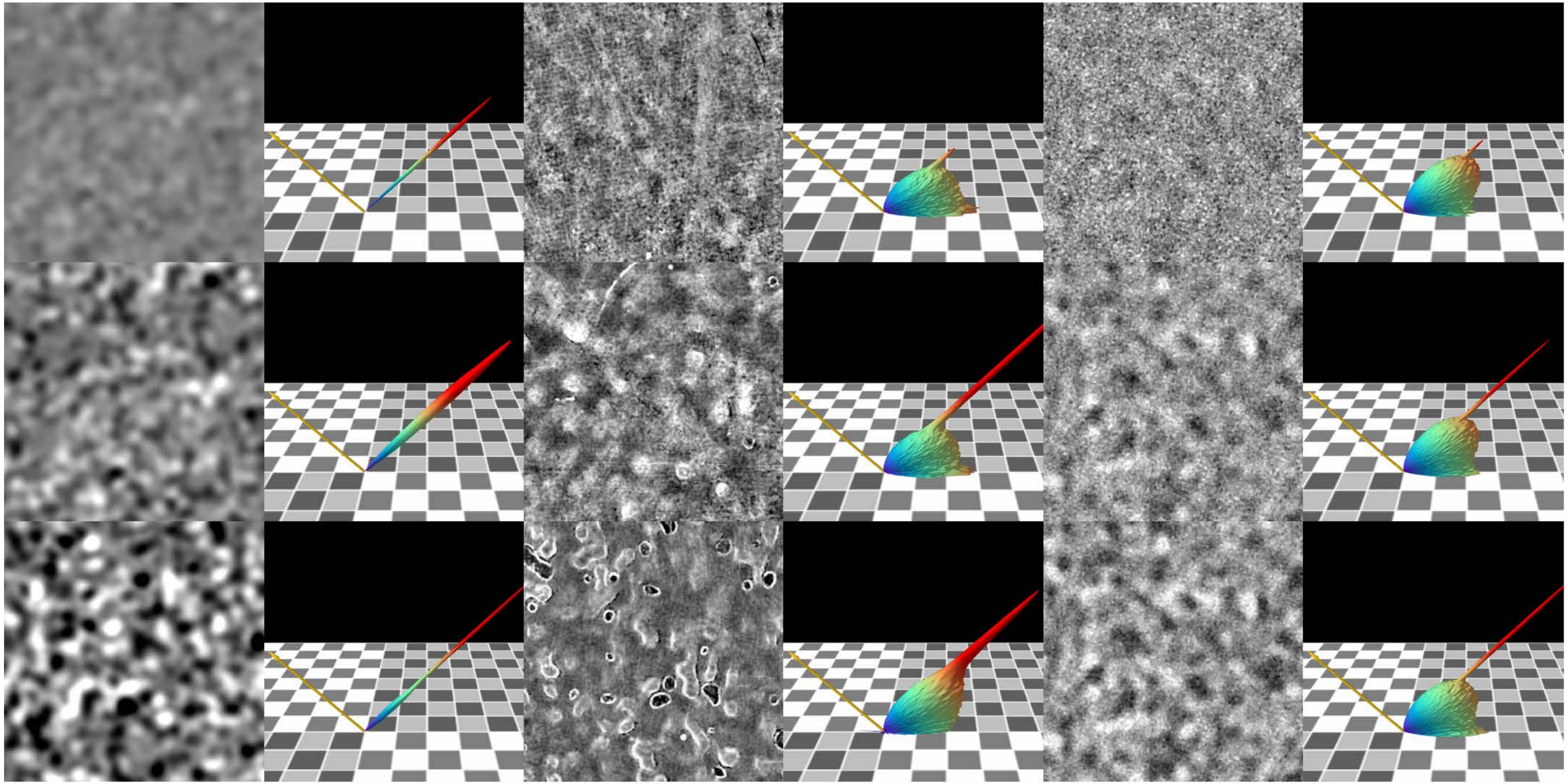

Sensors
Physical World
Digital Representation
Synthesis, Prediction & Modeling
Actuators

Appearance printing



appearance control through 3D printing

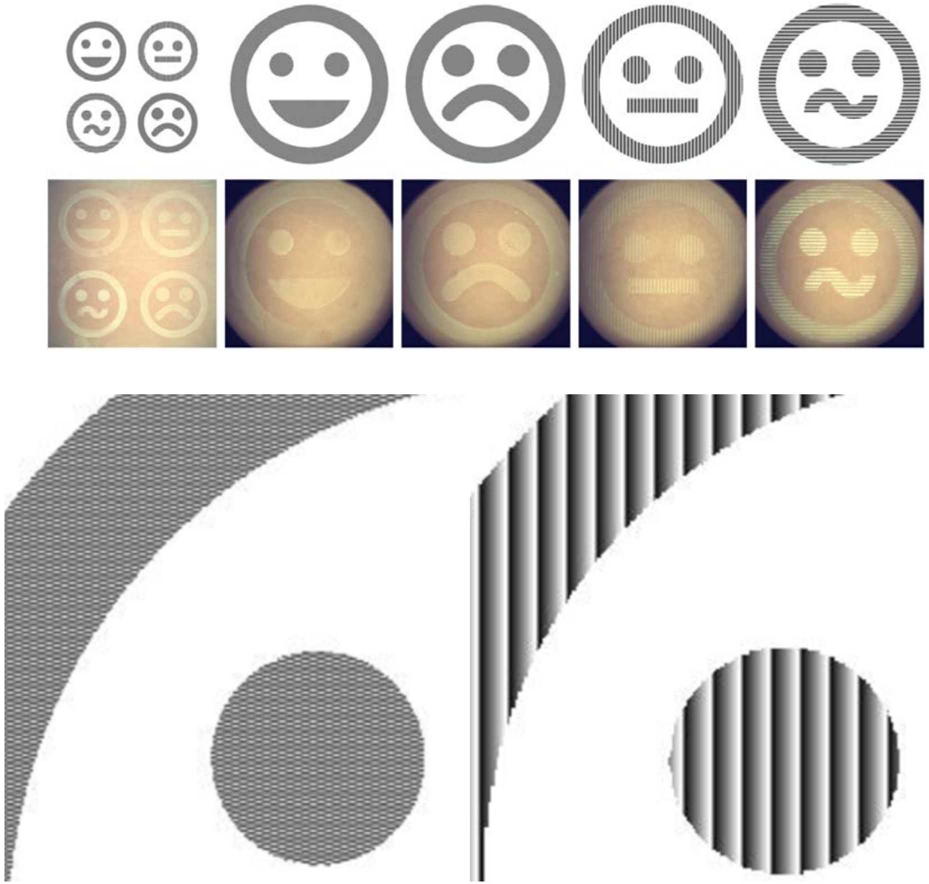
BRDF printing and 3D printer ground noise



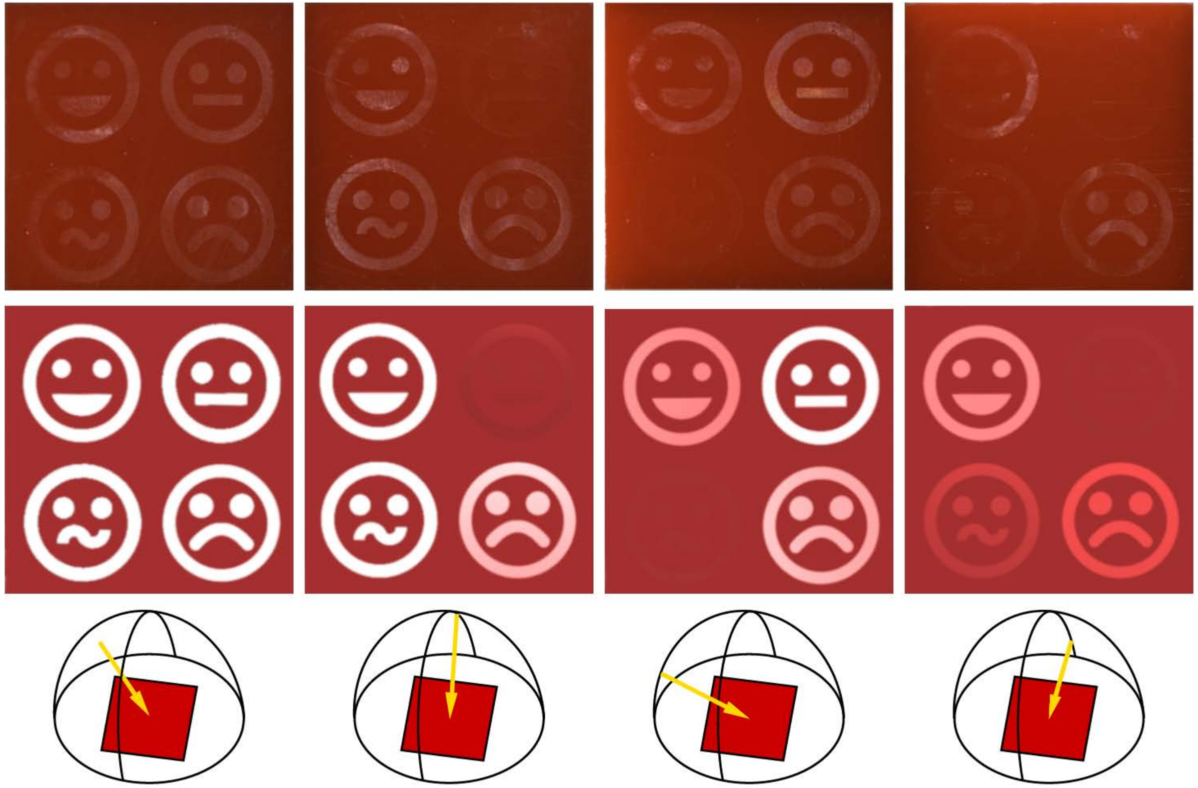
(a) input (b) input BRDF (c) measured (d) printed BRDF (e) input + ground (f) model BRDF

$$\mathrm{ground}(\boldsymbol{p}) = \frac{2}{3}\mathrm{noise}\left(\frac{\boldsymbol{p}}{50\ \mu m}\right) + \frac{1}{9}\mathrm{noise}\left(\frac{\boldsymbol{p}}{25\ \mu m}\right) + \frac{1}{12}\mathrm{noise}\left(\frac{\boldsymbol{p}}{2\ \mu m}\right)$$

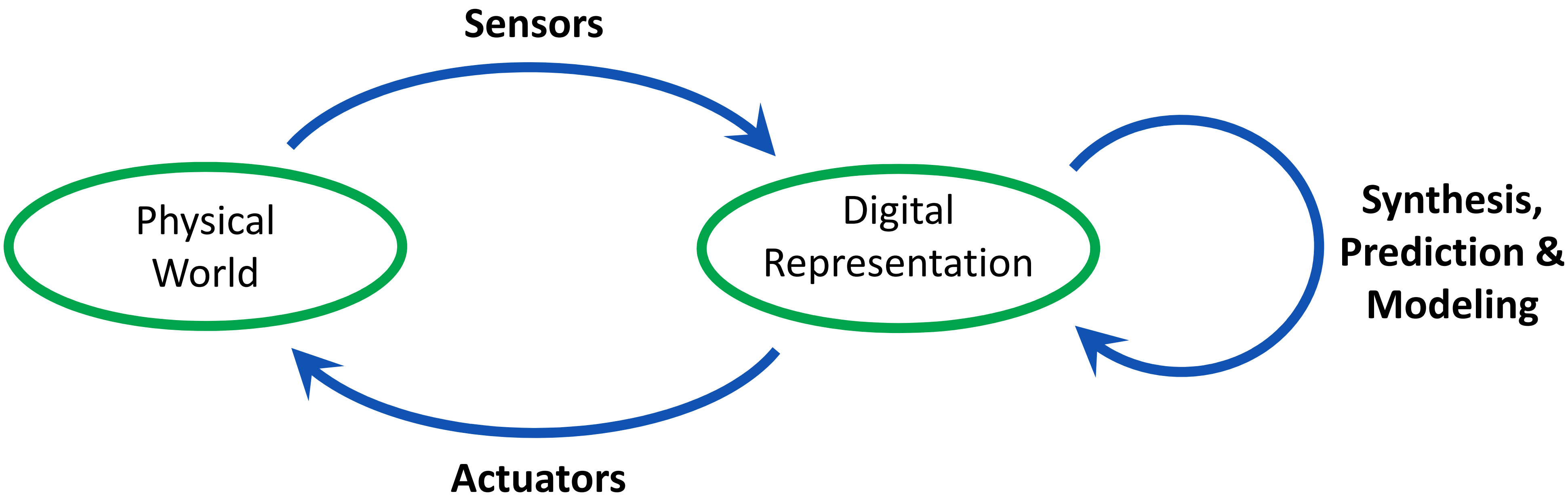
Anisotropic smileys (ridges and sinusoids)

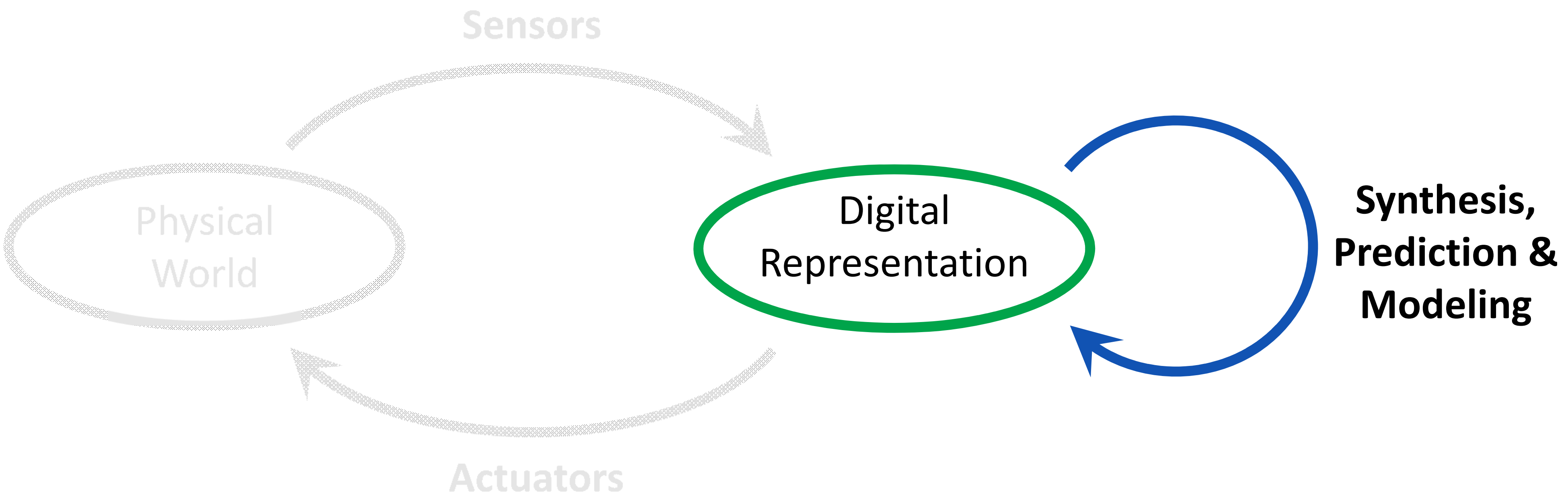


Input texture and microscope images.



Top row: 3D printed
Middle row: computed
Bottom row: direction of incidence


Sensors
Physical World
Digital Representation
Synthesis, Prediction & Modeling
Actuators


Sensors
Physical World
Digital Representation
Synthesis, Prediction & Modeling
Actuators

Digital Representation

Synthesis, Prediction & Modeling

Building models using particle composition and Lorenz-Mie theory

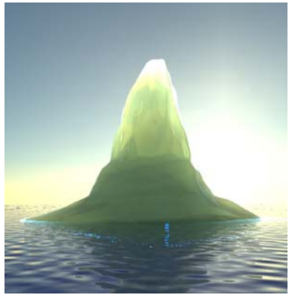
Particles:
Refractive index, concentration, size distribution, density

Host medium:
Refractive index, density, viscosity, surface tension

Global parameters:
temperature, gravity, pressure

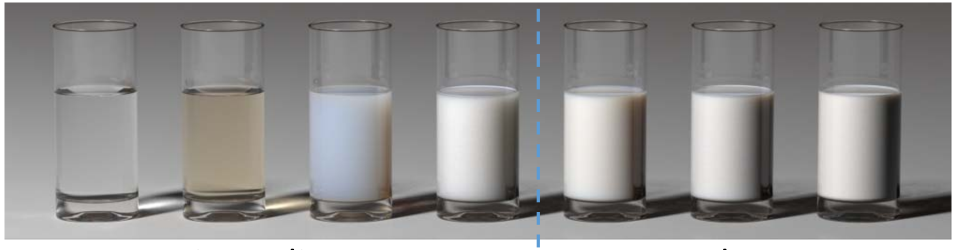
Model → Rendering & Simulation → Digital Prototype

algae in sea ice



[Frisvad 2008]

milk: water, vitamins, protein and fat particles



ingredients | products

[Frisvad et al. 2007a]

Light-material interaction in a volume

- Some light is absorbed.
- Some light scatters away (out-scattering).
- Some light scatters back into the line of sight (in-scattering).

  (absorption + out-scattering = extinction)

- Historical origins:

Bouguer [1729, 1760]	A measure of light. Exponential extinction.
Lambert [1760]	Cosine law of perfectly diffuse reflection and emission.
Lommel [1887]	Testing Lambert's cosine law for scattering volumes.
	Describing isotropic in-scattering mathematically.
Chwolson [1889]	A theory for subsurface light diffusion (similar to Lommel's).
Schuster [1905]	Scattering in foggy atmospheres (plane-parallel media).
	Reinventing the theory in astrophysics.
King [1913]	General equation which includes anisotropic scattering (phase function).
Chandrasekhar [1950]	The first definitive text on radiative transfer.

Radiative transfer and scattering properties

- We follow a ray of light passing through a scattering medium.
- The parameters describing the medium are

  σ_a the absorption coefficient $[\mathrm{m}^{-1}]$
  σ_s the scattering coefficient $[\mathrm{m}^{-1}]$
  σ_t the extinction coefficient $[\mathrm{m}^{-1}]$ $(\sigma_t = \sigma_a + \sigma_s)$
  p the phase function $[\mathrm{sr}^{-1}]$
  ε the emission properties $[\mathrm{W sr^{-1} m^{-3}}]$ (radiance per meter).

- The radiative transfer equation (RTE)

$$
\begin{aligned}
(\vec{\omega} \cdot \nabla) L(\boldsymbol{x}, \vec{\omega}) \quad = \quad & -\sigma_t(\boldsymbol{x}) L(\boldsymbol{x}, \vec{\omega}) \\
& + \sigma_s(\boldsymbol{x}) \int_{4\pi} p(\boldsymbol{x}, \vec{\omega}', \vec{\omega}) L(\boldsymbol{x}, \vec{\omega}') \, \mathrm{d}\omega' \\
& + \varepsilon(\boldsymbol{x}, \vec{\omega}) \quad ,
\end{aligned}
$$

where L is radiance at the position $\boldsymbol{x}$ along the ray in the direction $\vec{\omega}$.

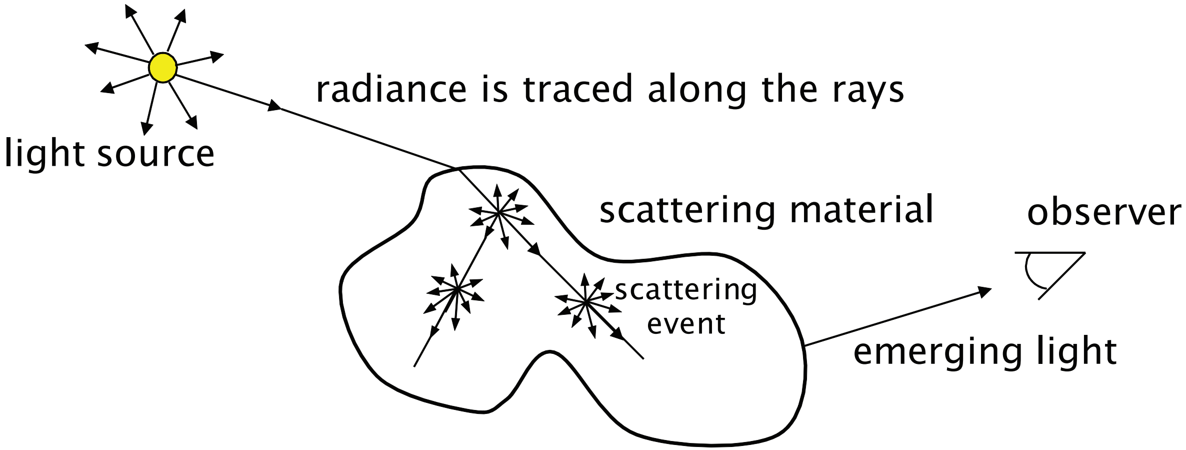
Computing appearance from scattering properties

- Prediction requires solving the radiative transfer equation:

$$(\vec{\omega}\cdot\nabla)L(\boldsymbol{x},\vec{\omega}) = -\sigma_t(\boldsymbol{x})L(\boldsymbol{x},\vec{\omega}) + \sigma_s(\boldsymbol{x})\int_{4\pi} p(\boldsymbol{x},\vec{\omega}',\vec{\omega})L(\boldsymbol{x},\vec{\omega}')\,\mathrm{d}\omega' + \varepsilon(\boldsymbol{x},\vec{\omega})\,.$$

- The solution method of choice today:

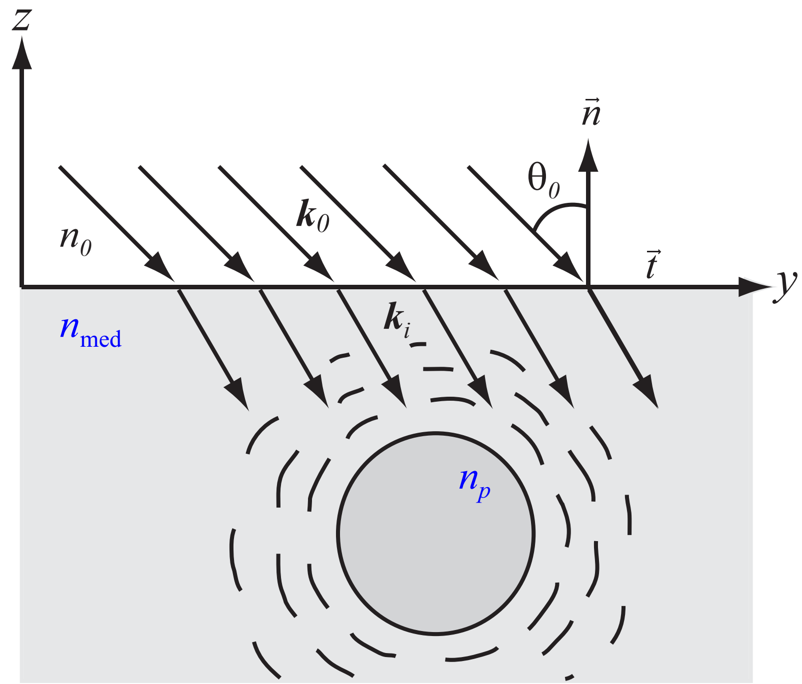
  Stochastic ray tracing (Monte Carlo integration).



- How do we compute input scattering properties from the particle composition of a material?

Scattering of a plane wave by a spherical particle

- A plane wave scattered by a spherical particle gives rise to a spherical wave.
- The components of a spherical wave are spherical functions.
- To evaluate these spherical functions, we use spherical harmonic expansions.
- Coefficients in these spherical harmonic expansions are referred to as Lorenz-Mie coefficients a_n and b_n.



- Lorenz [1890] and Mie [1908] derived formal expressions for a_n and b_n using the spherical Bessel functions j_n and y_n.
- These expressions are written more compactly if we use the Riccati-Bessel functions: $\psi_n(z) = z\, j_n(z) \quad , \quad \zeta_n(z) = z(j_n(z) - i\, y_n(z))\,,$ where z is (in general) a complex number.

The Lorenz-Mie coefficients (a_n and b_n)

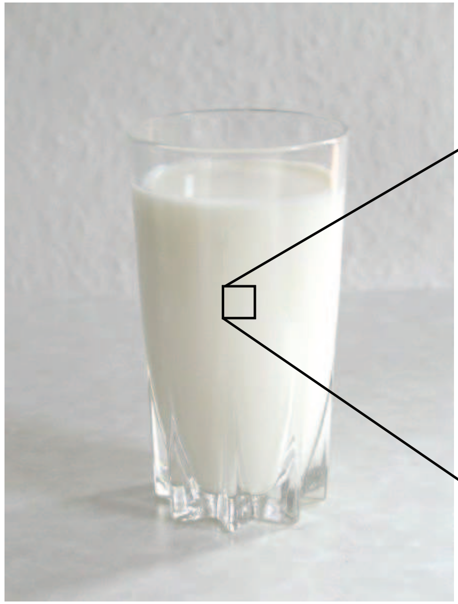
- Using the Riccati-Bessel functions ψ_n and ζ_n, the expressions for the Lorenz-Mie coefficients are

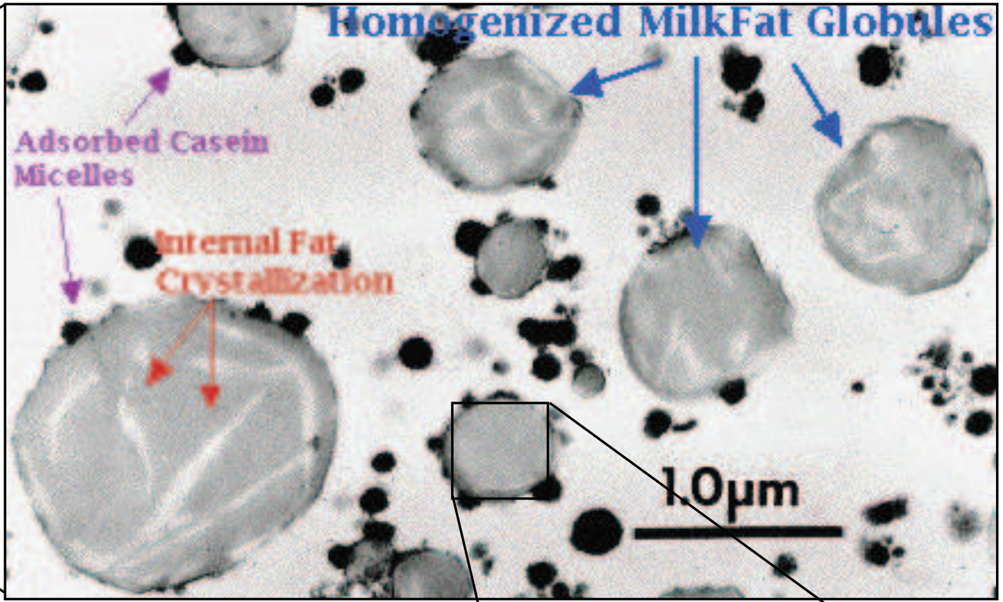
$$\begin{aligned} a_n &= \frac{n_{\text{med}}\psi_n'(y)\psi_n(x) - n_p\psi_n(y)\psi_n'(x)}{n_{\text{med}}\psi_n'(y)\zeta_n(x) - n_p\psi_n(y)\zeta_n'(x)} \\ b_n &= \frac{n_p\psi_n'(y)\psi_n(x) - n_{\text{med}}\psi_n(y)\psi_n'(x)}{n_p\psi_n'(y)\zeta_n(x) - n_{\text{med}}\psi_n(y)\zeta_n'(x)} \ . \end{aligned}$$

  - Primes $'$ denote derivative.
  - n_{med} and n_p are the refractive indices of the host medium and the particle respectively.
  - x and y are called size parameters.
- If r is the particle radius and λ is the wavelength *in vacuo*, then x and y are defined by

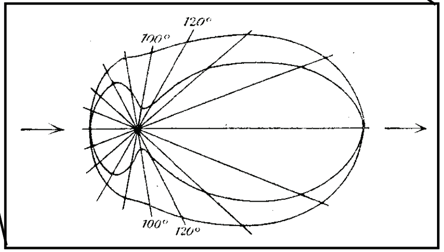
$$x = \frac{2\pi r n_{\text{med}}}{\lambda} \quad , \quad y = \frac{2\pi r n_p}{\lambda} \ .$$

From particles to appearance





Courtesy of University of Guelph



Lorenz–Mie theory provides the link

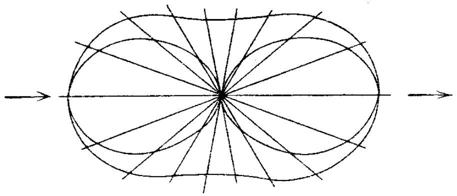
Scattering by spherical particles

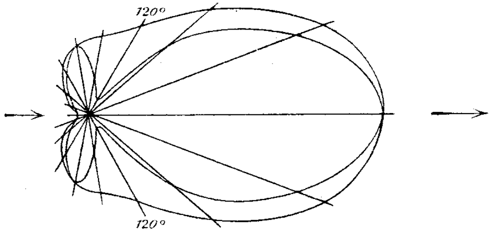
- The Lorenz-Mie theory:

$$
\begin{aligned}
p(\theta) &= \frac{|S_1(\theta)|^2 + |S_2(\theta)|^2}{2|k|^2 C_s} \\
S_1(\theta) &= \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \left(a_n \pi_n(\cos\theta) + b_n \tau_n(\cos\theta) \right) \\
S_2(\theta) &= \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \left(a_n \tau_n(\cos\theta) + b_n \pi_n(\cos\theta) \right) \quad .
\end{aligned}
$$

- a_n and b_n are the Lorenz-Mie coefficients.
- π_n and τ_n are spherical functions associated with the Legendre polynomials.

small particle





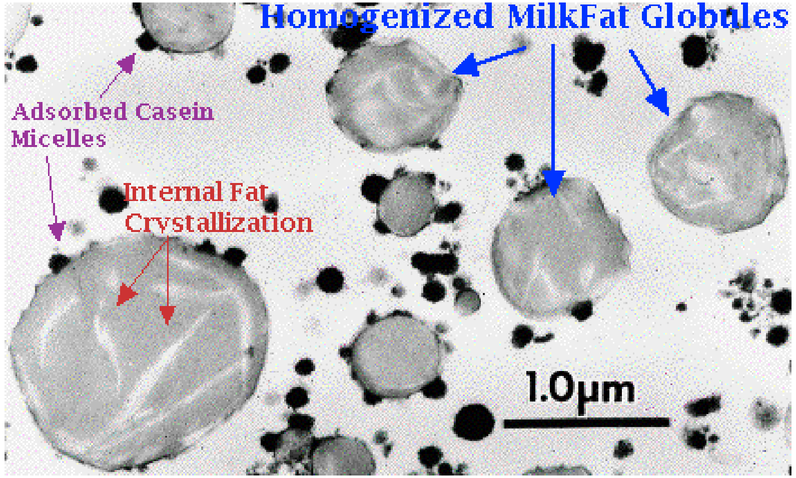
large particle

Quantity of scattering

- Lorenz-Mie theory continued:

  The scattering and extinction cross sections of a particle:

$$
\begin{aligned}
C_s &= \frac{\lambda^2}{2\pi|n_{\mathrm{med}}|^2}\sum_{n=1}^{\infty}(2n+1)\left(|a_n|^2+|b_n|^2\right)\\
C_t &= \frac{\lambda^2}{2\pi}\sum_{n=1}^{\infty}(2n+1)\mathrm{Re}\left(\frac{a_n+b_n}{{n_{\mathrm{med}}}^2}\right) .
\end{aligned}
$$



Bulk optical properties of a material

- Input is the desired volume fraction of a component v and a representative number density distribution $\hat{N}$. We have

$$\hat{v} = \frac{4\pi}{3} \int_{r_{\min}}^{r_{\max}} r^3 \hat{N}(r)\, \mathrm{d}r \quad ,$$

  and then the desired distribution is $N = \hat{N} v / \hat{v}$.
- Use this to find the bulk properties σ_s (and σ_t likewise)

$$\sigma_s = \int_{r_{\min}}^{r_{\max}} C_s(r) N(r)\, \mathrm{d}r \quad .$$

Computing scattering properties

- Input needed for computing scattering properties:
  - Particle composition (volume fractions, particle shapes).
  - Refractive index for host medium n_{med}.
  - Refractive index for each particle type n_p.
  - Size distribution for each particle type (N).

- Lorenz-Mie theory uses a series expansion. How many terms should we include?
- Number of terms to sum $M = \left\lceil |x| + p|x|^{1/3} + 1 \right\rceil$.
  - Empirically justified [Wiscombe 1980, Mackowski et al. 1990].
  - Theoretically justified [Cachorro and Salcedo 1991].
  - For a maximum error of 10^{-8}, use $p = 4.3$.

- Code for evaluating the expansions in the Lorenz-Mie theory is available online [Frisvad et al. 2007]: https://people.compute.dtu.dk/jerf/code/

Particle contents (examples)

- Natural water
  - Refractive index of host: saline water.
  - Mineral and alga contents: user input in volume fractions.
  - Refractive indices of mineral and algae: empirical formulae.
  - Shape of mineral and algal particles: spheres.
  - Size distributions: power laws.
- Icebergs
  - Refractive index of host: pure ice.
  - Brine and air contents: depend on temperature, salinity, and density.
  - Refractive index of brine: empirical formula, measured absorption spectrum.
  - Shape of brine pores and air pockets: closed cylinders and ellipsoids.
  - Size distributions: power laws.
- Milk
  - Refractive index of host: water + dissolved vitamin B2.
  - Fat and protein contents: user input in wt.-%.
  - Refractive index of milk fat and casein: measured spectra.
  - Shape of fat globules and casein micelles: spheres and a volume to surface area ratio.
  - Size distributions: log-normal with mean depending on fat content and homogenization pressure.

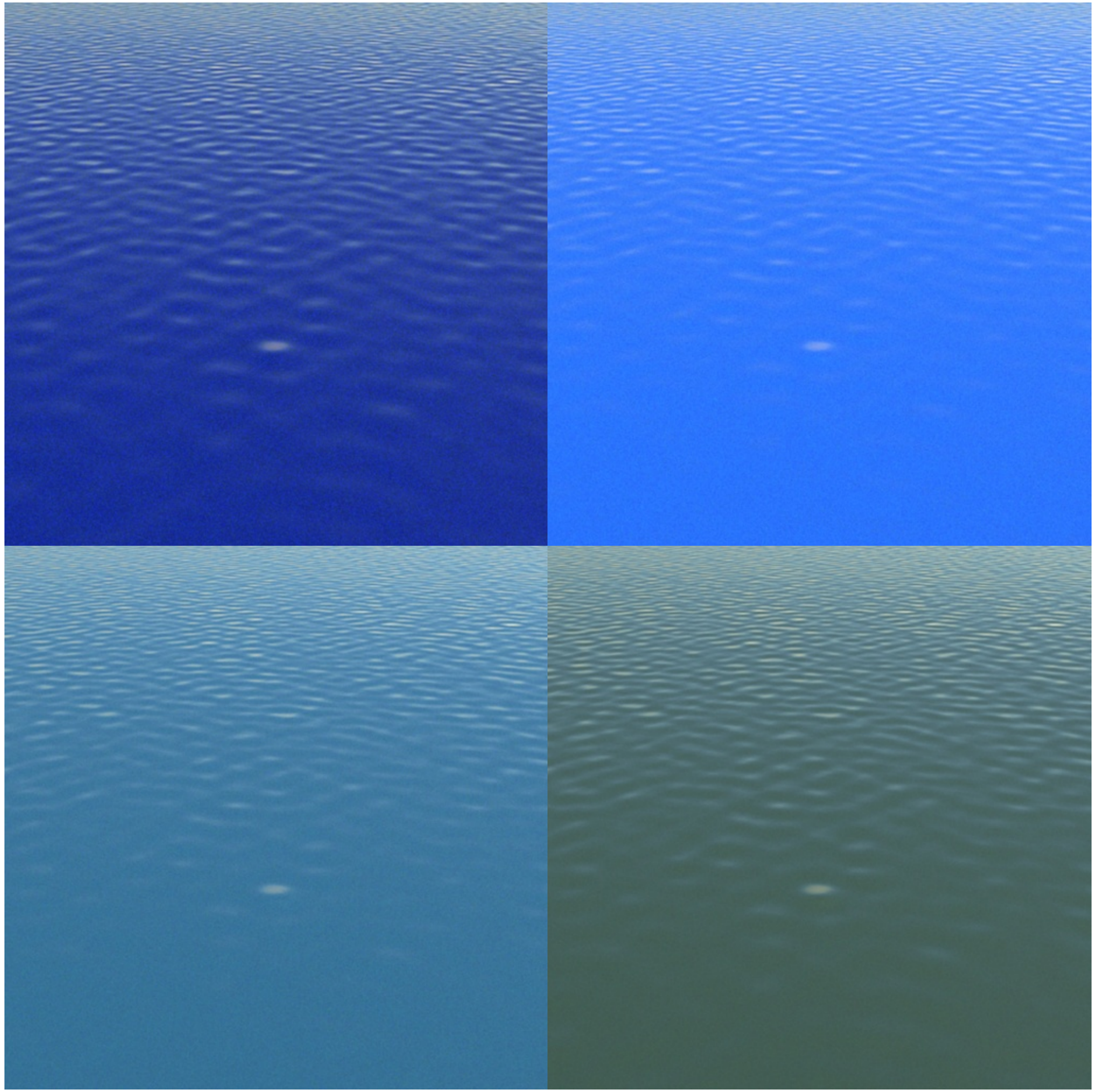
Case study: natural waters

- Glacial melt water with rock flour mixing with purer water from melted snow to give Lake Pukaki in New Zealand its beautiful bright blue colour.

Oceanic and coastal waters

Cold Atlantic

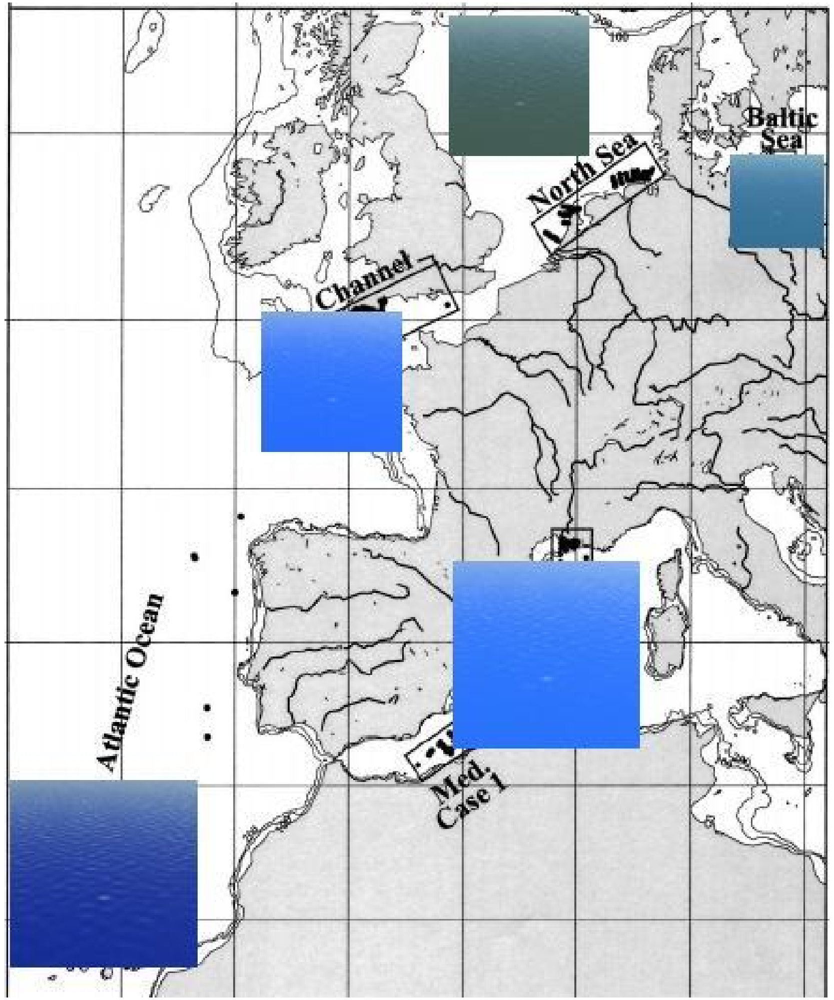
Mediterranean



Baltic

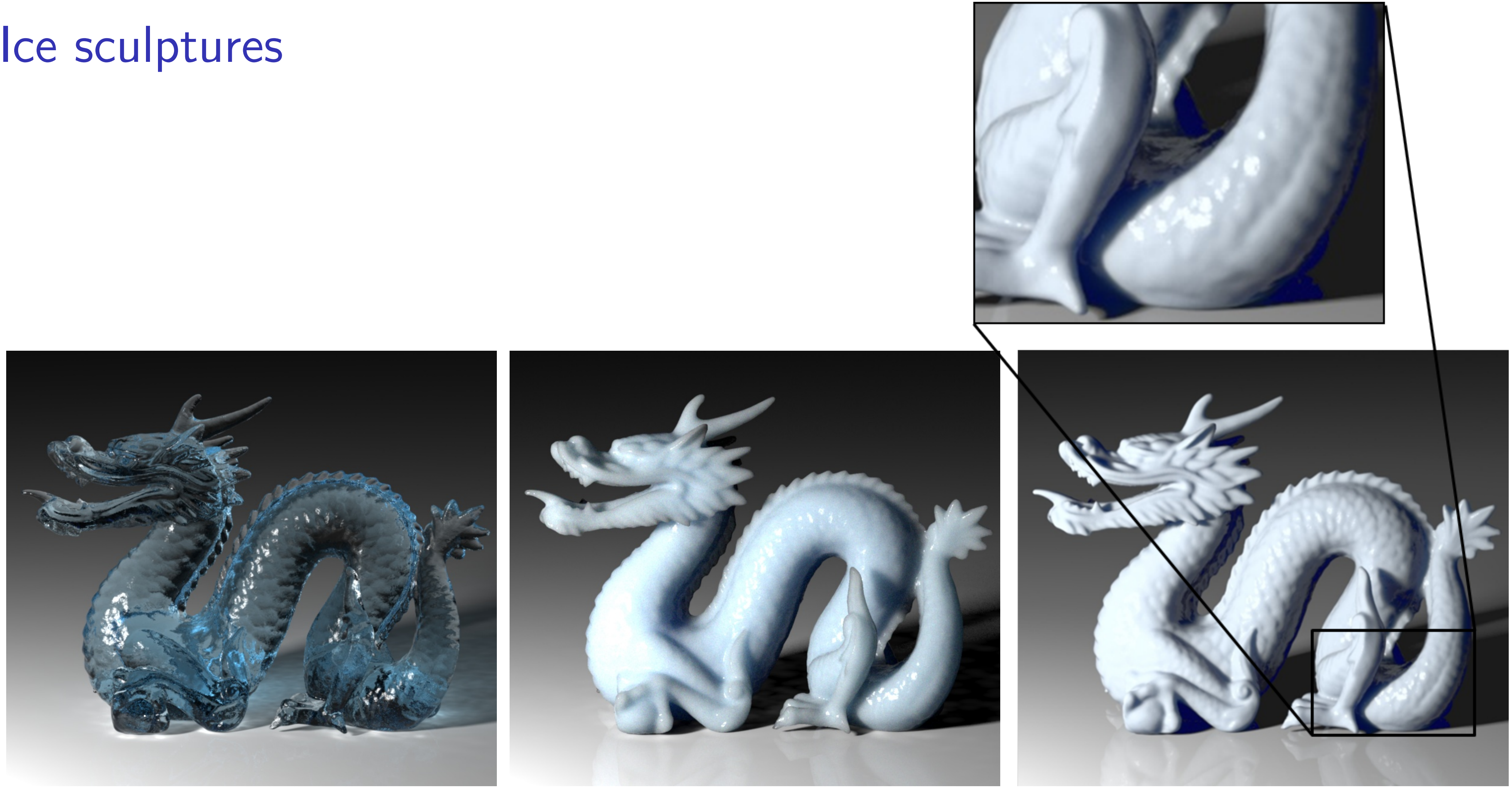
North Sea

Oceanic and coastal waters



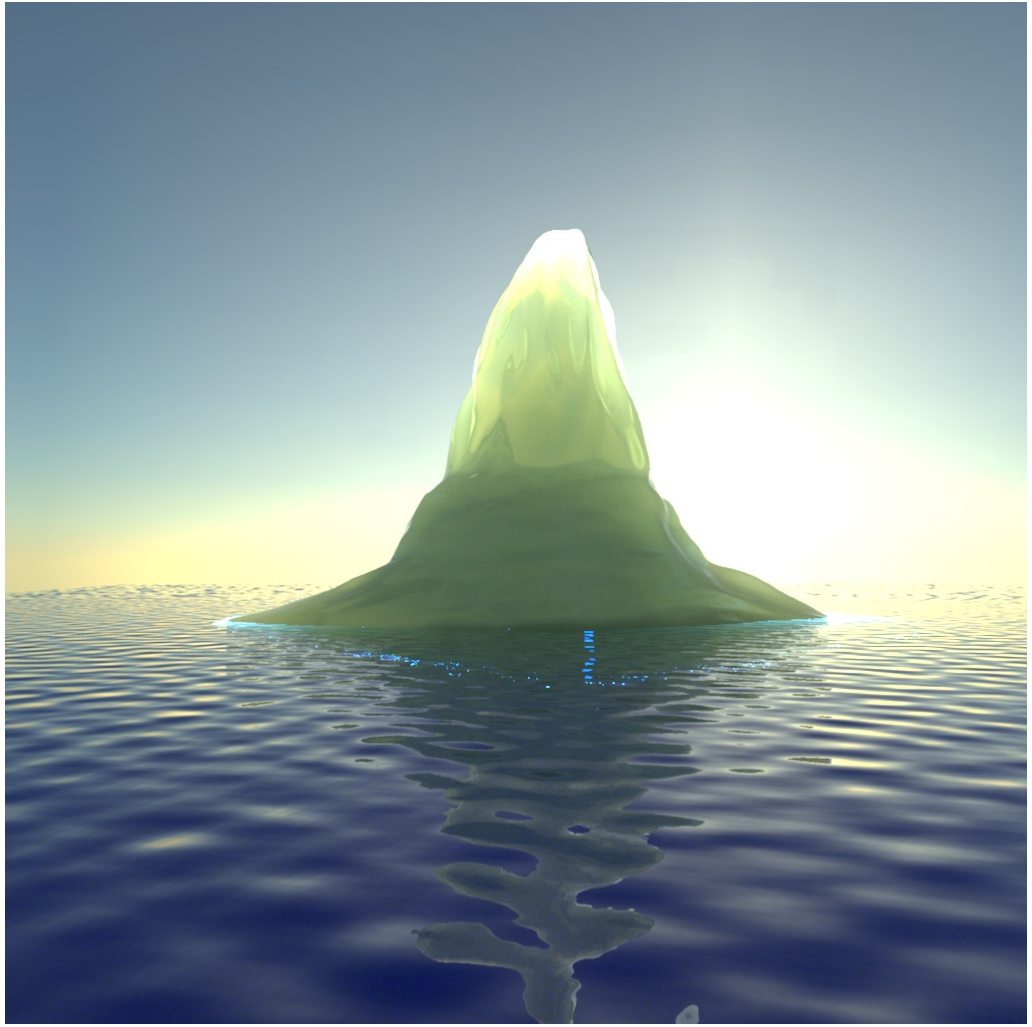
Case study: icebergs

Ice sculptures



pure ice compacted snow white ice

Algal ice



Green algae on iceberg.
(Photo courtesy Gerhard Dieckmann)

Case study: milk

skimmed

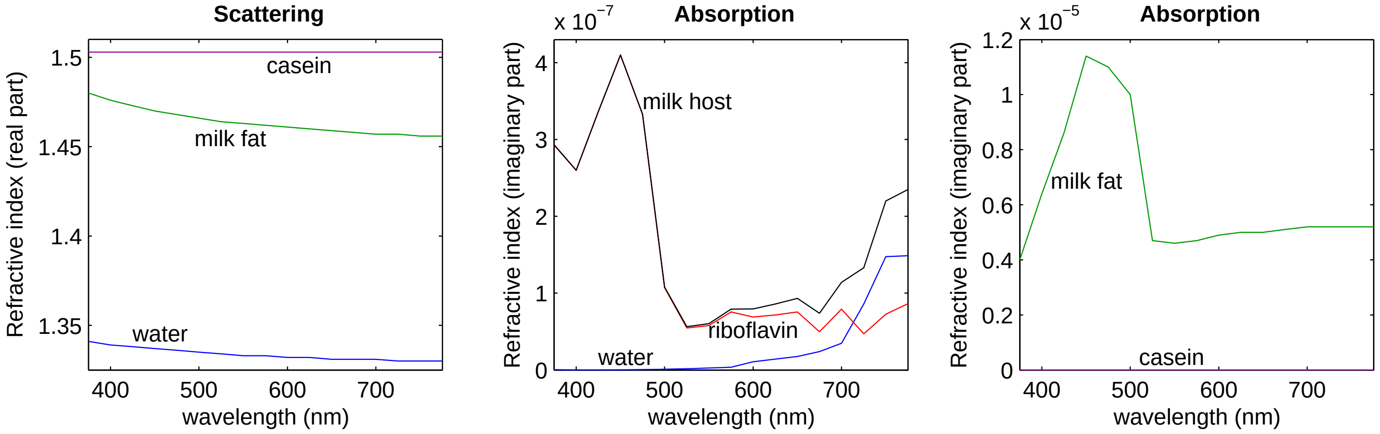
low fat

whole

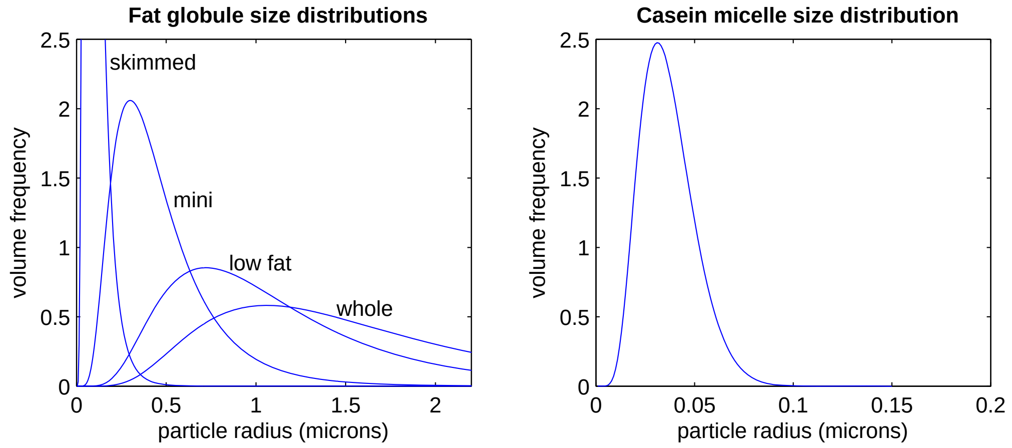
- Refractive index of host: water + dissolved vitamin B2.
- Fat and protein contents: user input in wt.-%.
- Refractive index of milk fat and casein: measured spectra.
- Shape of fat globules and casein micelles: spheres and a volume to surface area ratio.
- Size distributions: log-normal with mean depending on fat content and homogenization pressure.

Measurements used for the milk model

- Refractive indices:



- Particle size distributions:



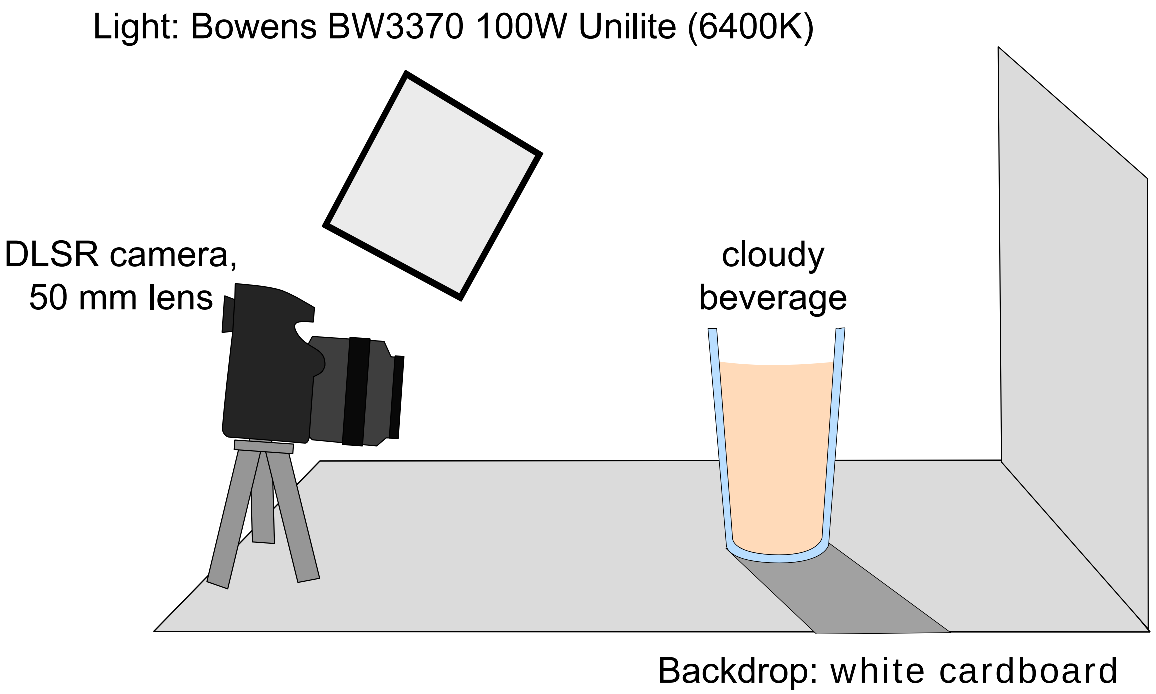
Predicting appearance based on a content declaration

water vitamin B2 protein fat skimmed low fat whole

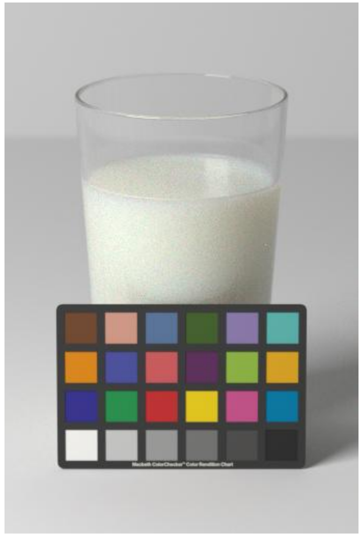
- Vitamin B2 content: 0.17 mg / 100 g
- Protein content: 3 g / 100 g
- Fat content: 0.1 g (skimmed), 1.5 g (low fat), 3.5 g (whole) / 100 g
- Homogenization pressure: 0 MPa (model: [0, 50] MPa)

Predicting appearance

Scene



organic low fat milk



rendering

photograph

- Digital scene modeled by hand to match physical scene (as best we could)

Case study: cloudy apple juice

> *The visual appearance of a cloudy drink is a decisive factor for consumer acceptance. [Beveridge 2002]*

Let us see if we can use Lorenz-Mie theory to create an appearance model useful for:

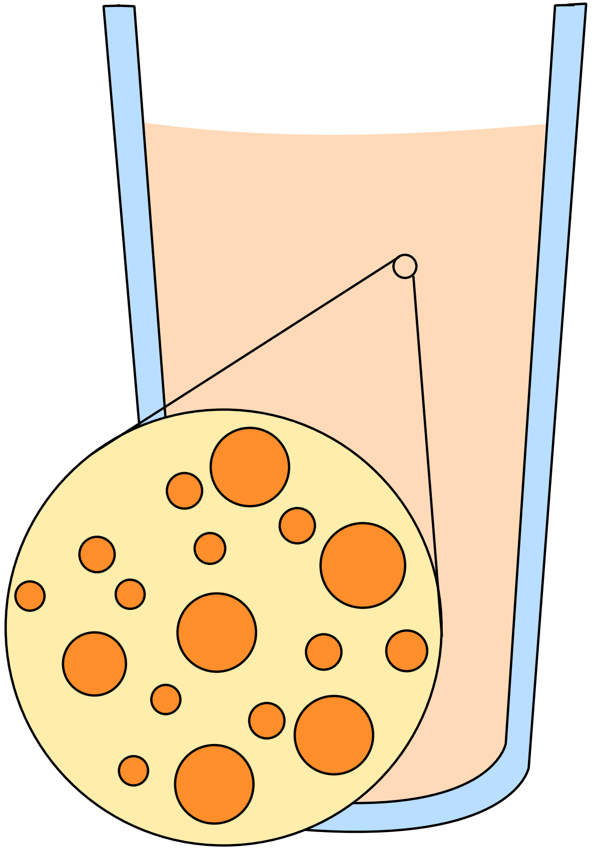
- predicting the visual effect of modifying production parameters;
- analyzing a given product with cameras.

Apple juice appearance model

- Host medium is water with dissolved solids (mostly sugars).
- Particles are browned apple flesh.

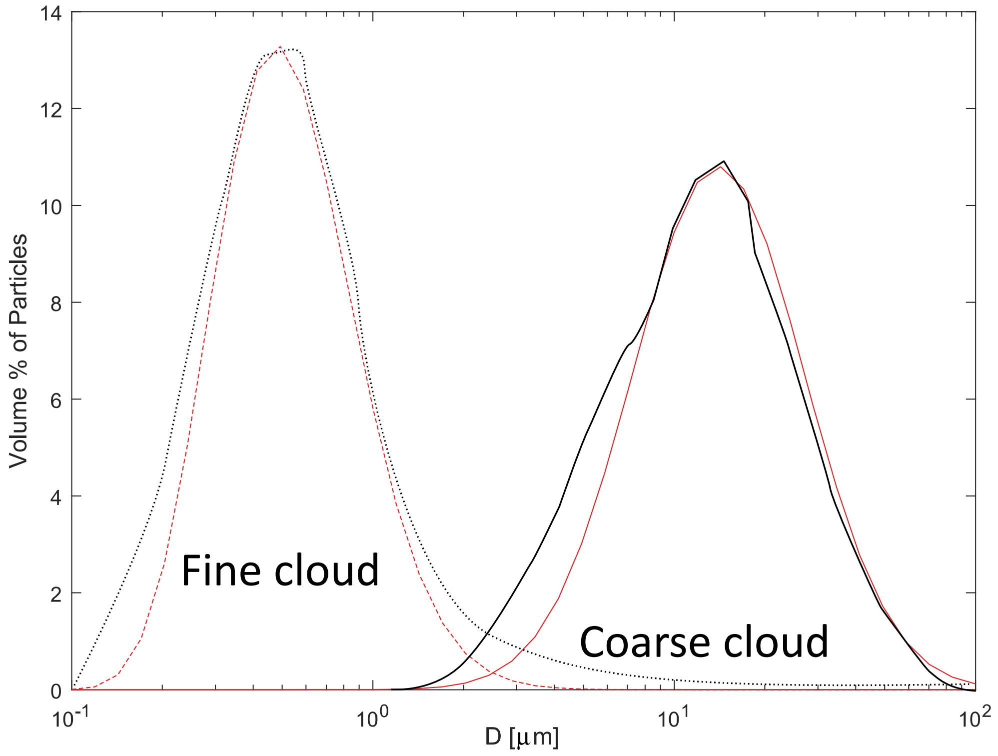
- Optical properties given by complex indices of refraction: $n = n' + i\,n''$.

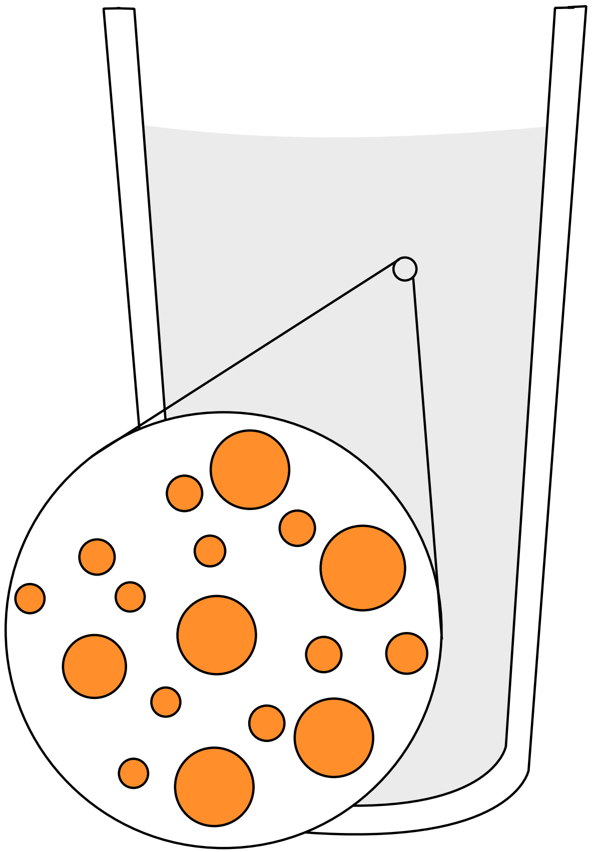
- We can relate these refractive indices to production parameters:
  - Particle concentration.
  - Storage time.
  - Handling of apples.
  - ...



Apple juice appearance model

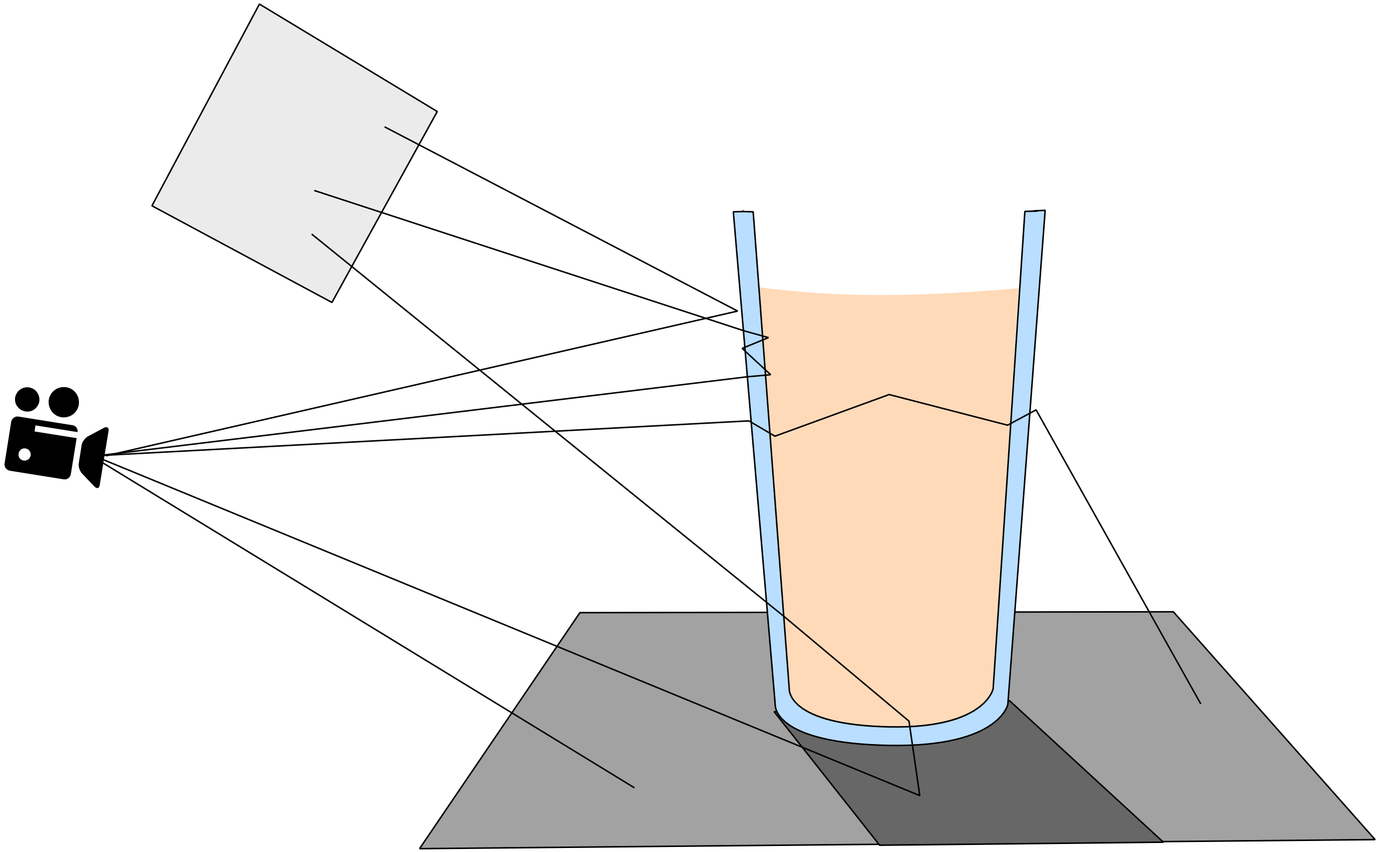
- We use a bimodal particle size distribution $\hat{N}$ from Zimmer et al. [1994], scaled to the desired volume concentration v of particles ($N = \hat{N}v/\hat{v}$).





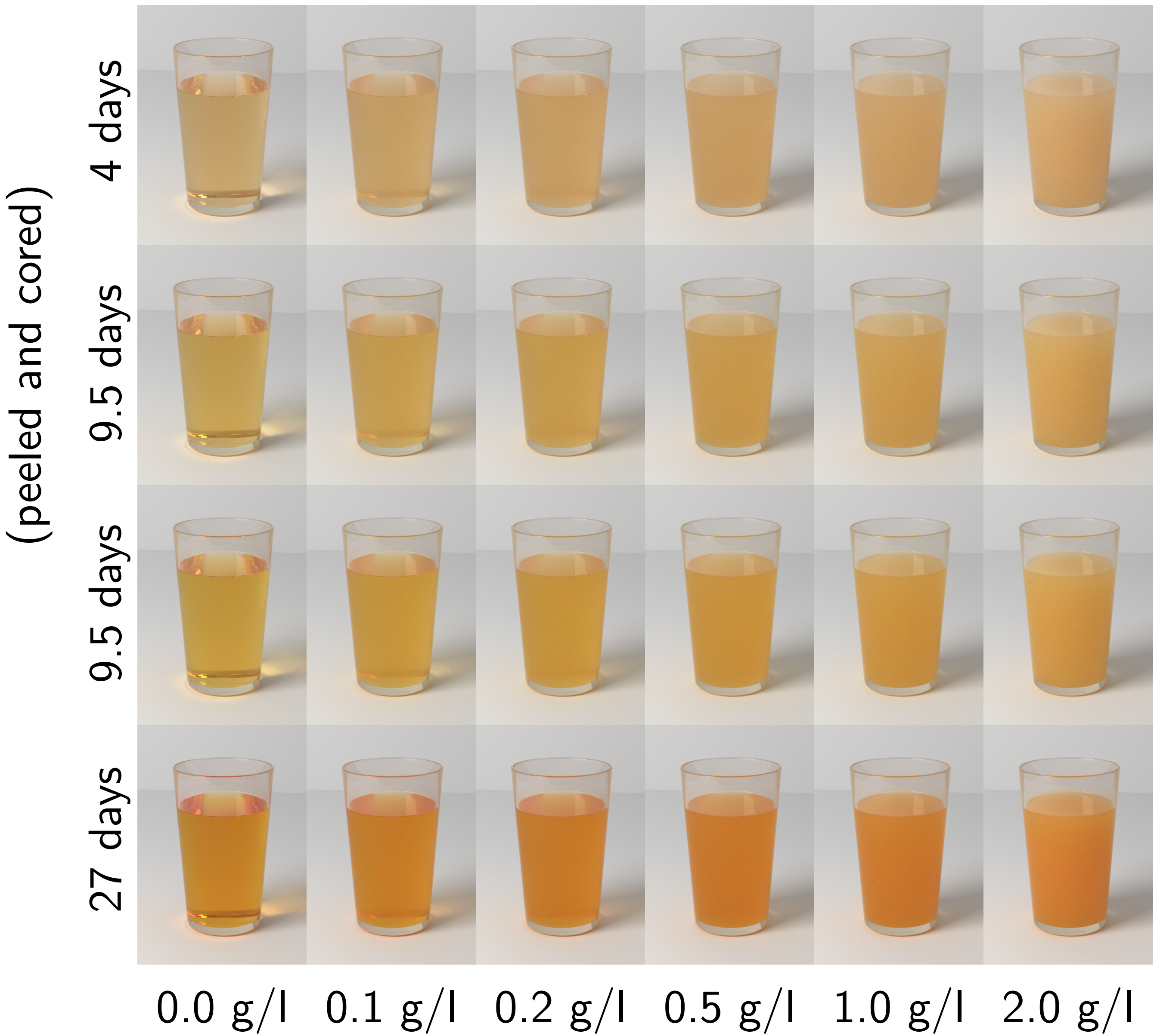
Rendering

- We can neither use single scattering nor diffusion theory.
- Thus, we use progressive unidirectional path tracing (Monte Carlo).
- Accounting for refractive indices using different interfaces.



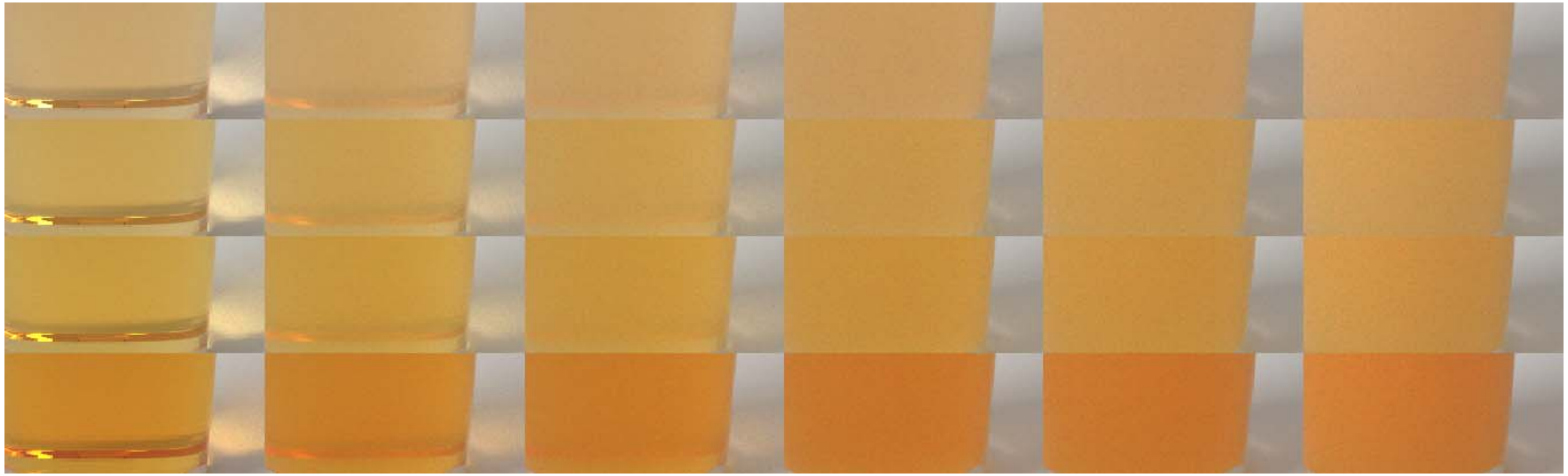
Results

- Varying particle concentration v (horizontally).
- Varying storage time and handling (vertically).



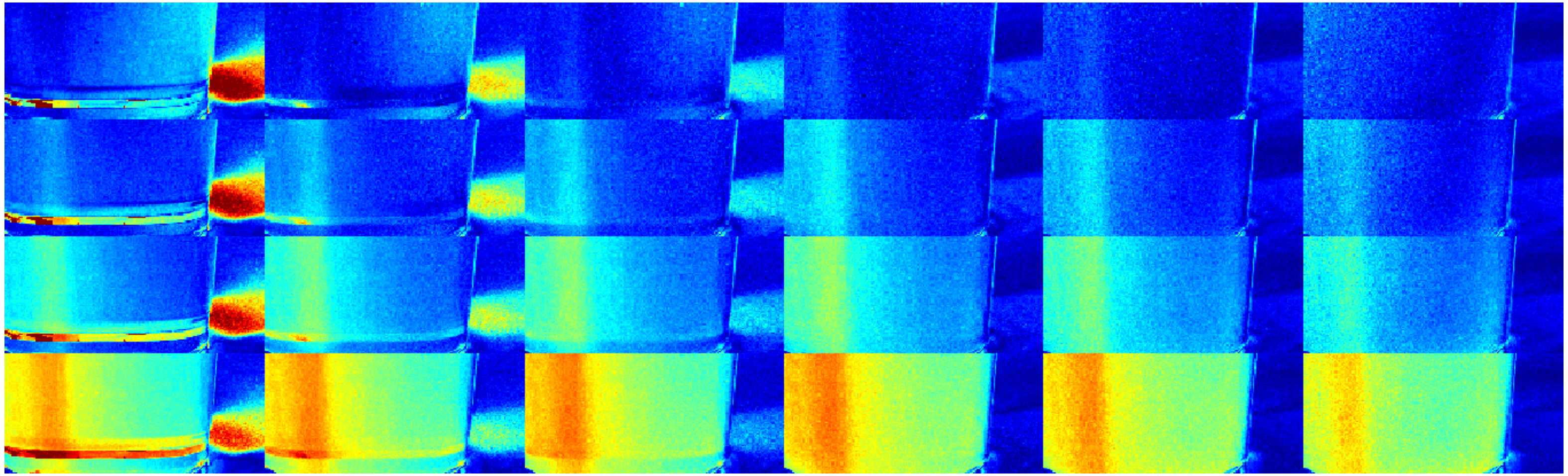
Patch-based quantitative comparison

Reference



	0.0 g/L	0.1 g/L	0.2 g/L	0.5 g/L	1.0 g/L	2.0 g/L
4 days						
9.5 days, peeled						
9.5 days						
27 days						

Patch-based quantitative comparison



4 days
9.5 days, peeled
9.5 days
27 days

0.0 g/L
0.1 g/L
0.2 g/L
0.5 g/L
1.0 g/L
2.0 g/L

Visual comparison - MAM 2016 rendering

rendering

photograph

Visual comparison - EPJH 2019 rendering

rendering

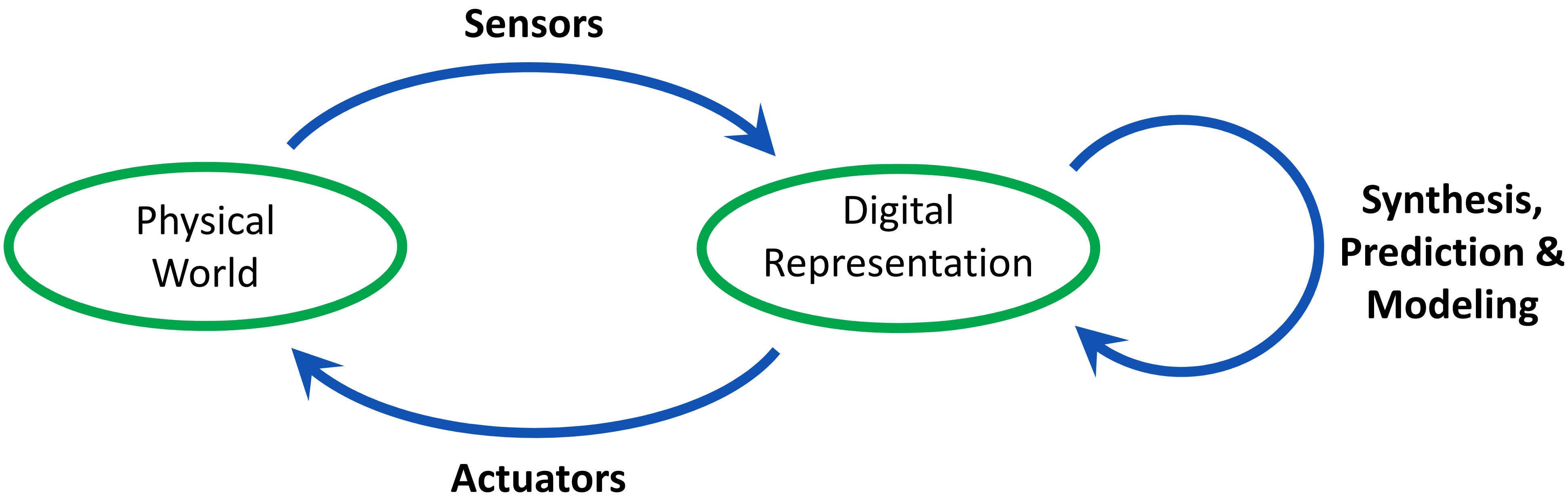
photograph

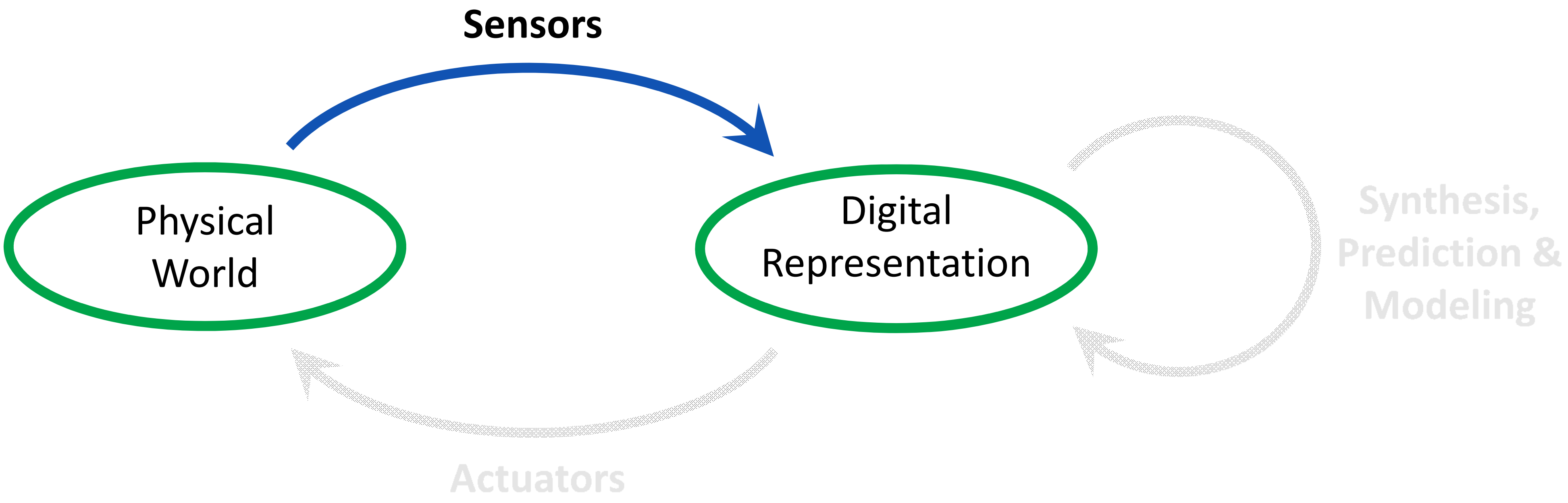
The input challenge

- Light transport simulation has come a long way, but renderings can only be as realistic/accurate as the input parameters permit.

- How do we get plausible input parameters?
  - Modeling (example: light scattering by particles).
  - Measuring (example: diffuse reflectance spectroscopy).

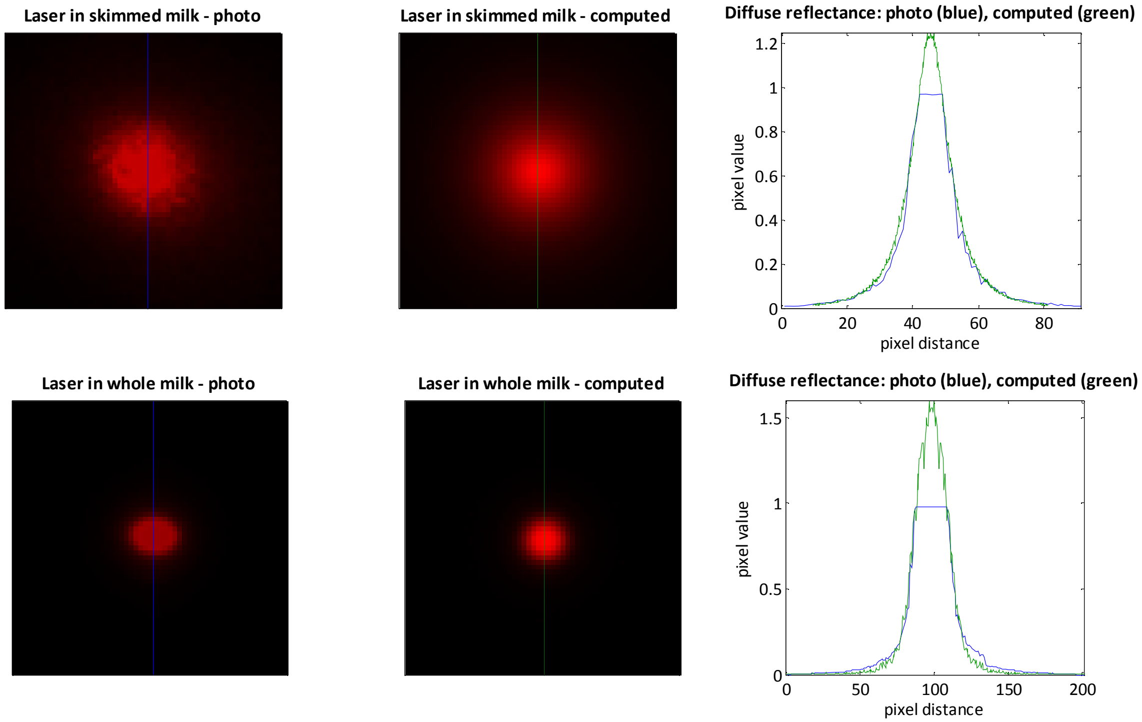
- Suppose we would like to go beyond visual comparison.
- How do we assess the appearance produced by a given set of input parameters?
  - Full digitization of a scene.
  - Reference photographs from known camera positions.
  - Pixelwise comparison of renderings with photographs.

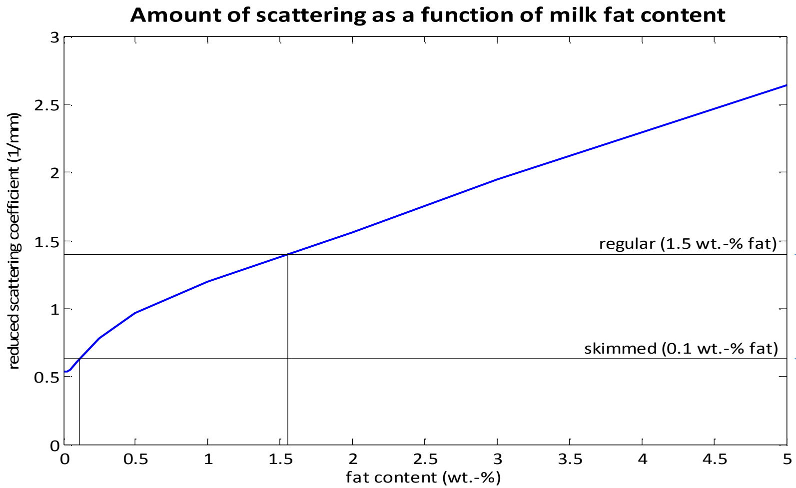

Sensors
Physical World
Digital Representation
Synthesis, Prediction & Modeling
Actuators


Sensors
Physical World
Digital Representation
Synthesis, Prediction & Modeling
Actuators

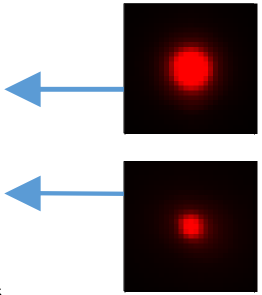
Simplistic model validation

- Camera
- Tripod
- Laser pointer
- Cup (use black cup)

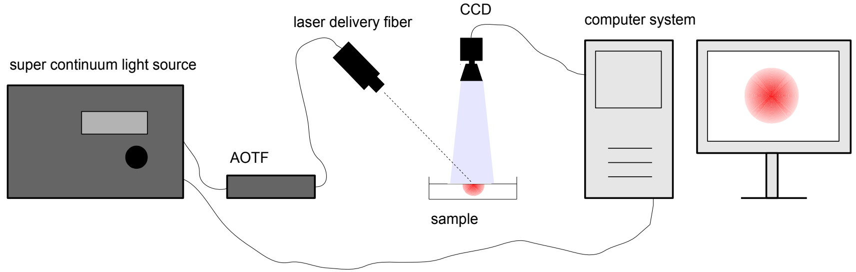




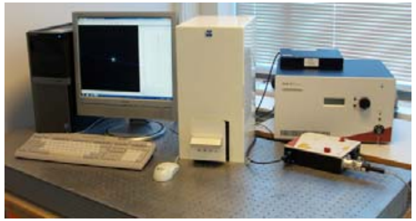
Captured images used for estimating the reduced scattering coefficient:



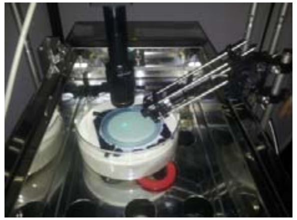
Measuring scattering properties using diffuse reflectance spectroscopy



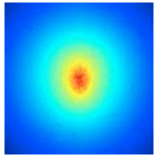
lab setup



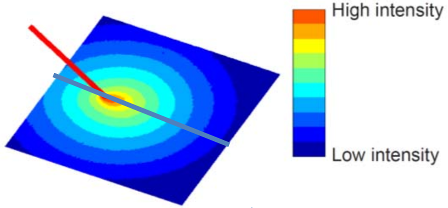
in situ setup



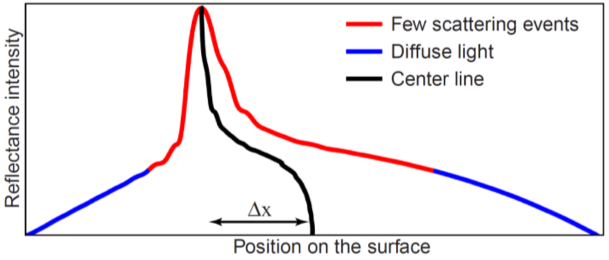
sample image
(log transformed, false colours)



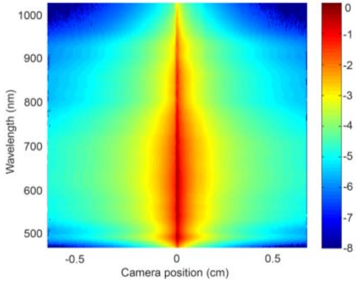
Infer optical properties using an analytic subsurface scattering model



extract profile

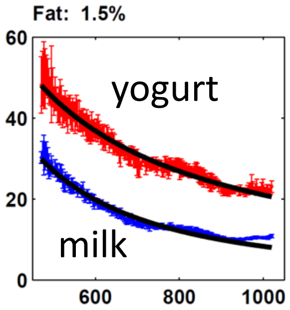


spectroscopy

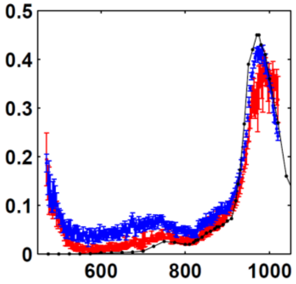


oblique incidence reflectometry

reduced scattering [1/cm]



absorption [1/cm]



wavelength [nm]

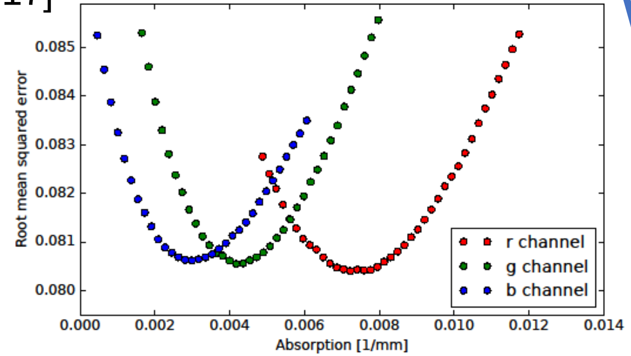
wavelength [nm]

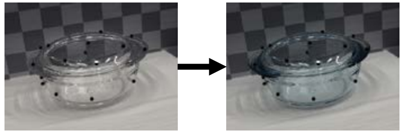
- Proper version of the simplistic approach used for validation of the milk model.

Research examples

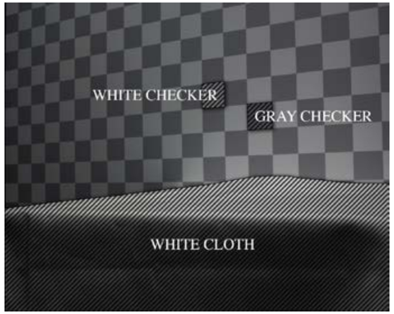
Acquiring input for appearance models

- Geometry from structured light and CT scans
  [Eiriksson *et al.* 2016; Stets *et al.* 2017; Guerra *et al.* 2019]
- Reflectance properties from sparse sampling and PCA reconstruction
  [Nielsen *et al.* 2015; Xu *et al.* 2016; Stets *et al.* 2017; Nielsen *et al.* 2017; Lyngby *et al.* 2019]
- Scattering properties from directional subsurface scattering
  [Abildgaard *et al.* 2015; Abildgaard *et al.* 2016]
- Material properties from analysis-by-synthesis
  [Dal Corso *et al.* 2016; Stets *et al.* 2017]
- Environment mapping
  [Stets *et al.* 2017]

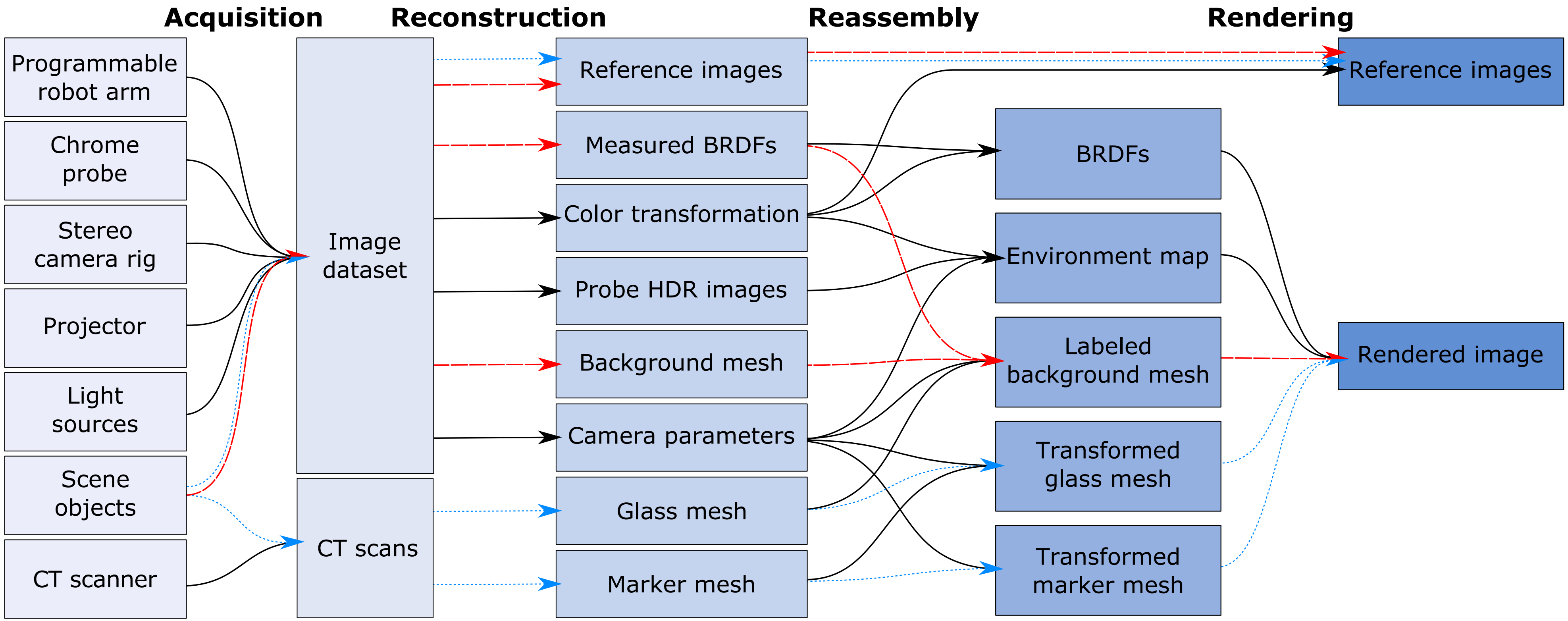




White checker | Grey checker | White cloth



Multimodal digitization pipeline

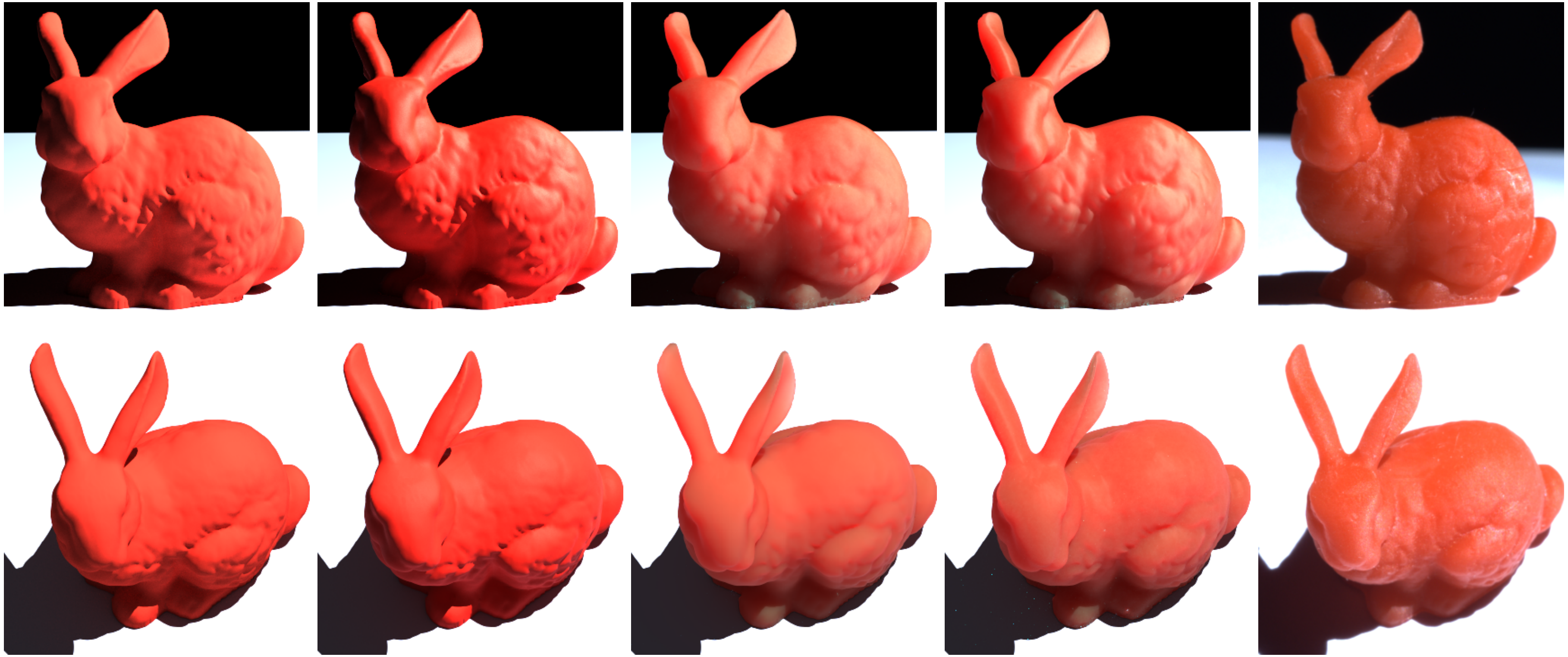


[pipeline video] [overview video]

- Data available at **https://eco3d.compute.dtu.dk/pages/transparency**

Analysis by synthesis

We can compare rendered images with photographs to estimate material parameters (model fitting) or to find deviations from expected appearance:

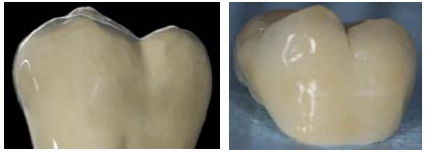


Rendered images using different material appearance models. References.

Research examples

Physical to digital material appearance.
Models: milk, juice, teeth, icebergs, rainbows, fur, glass

[Larsen *et al.* 2012]



render photo

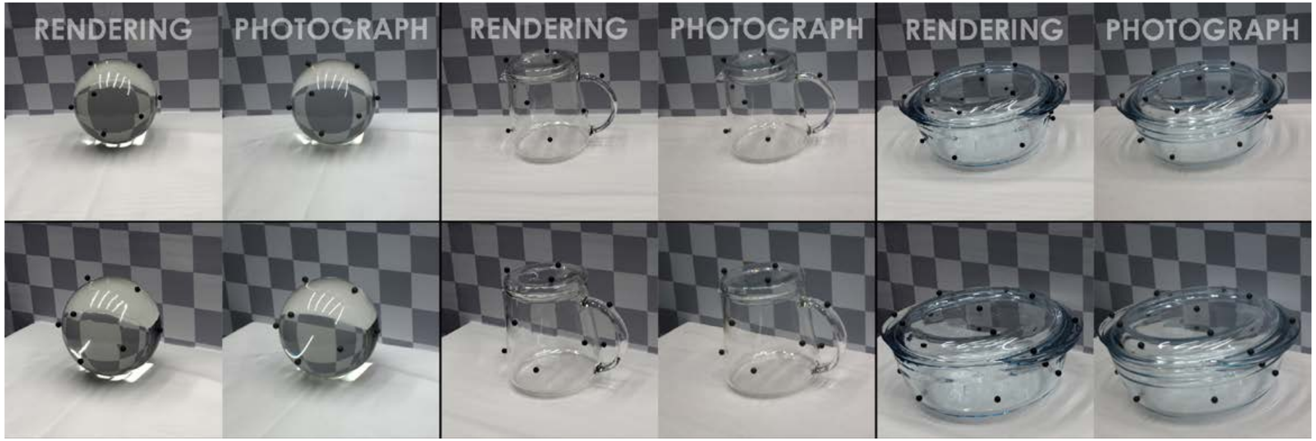
[Frisvad *et al.* 2005]

[Andersen *et al.* 2016]

algae in sea ice

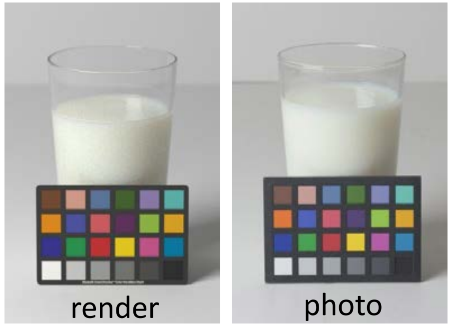
[Frisvad 2008]

[Stets *et al.* 2017]

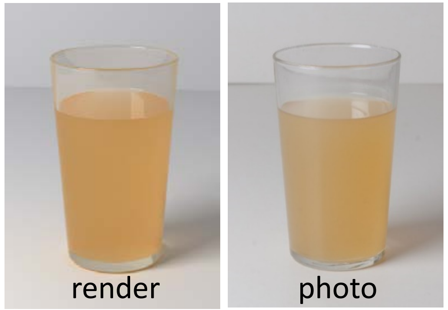


[Dal Corso *et al.* 2016]

organic low fat milk



unfiltered apple juice



Thank you for your attention

A. Luongo, V. Falster, M. B. Doest, M. M. Ribo, E. R. Eiriksson, D. B. Pedersen, and J. R. Frisvad. Microstructure control in 3D printing with digital light processing. *Computer Graphics Forum*, 2019. To appear.