Material appearance modeling & rendering

3DV Tutorial:

Methods for photographic radiometry,

modelling of light transport and material appearance

Jeppe Revall Frisvad

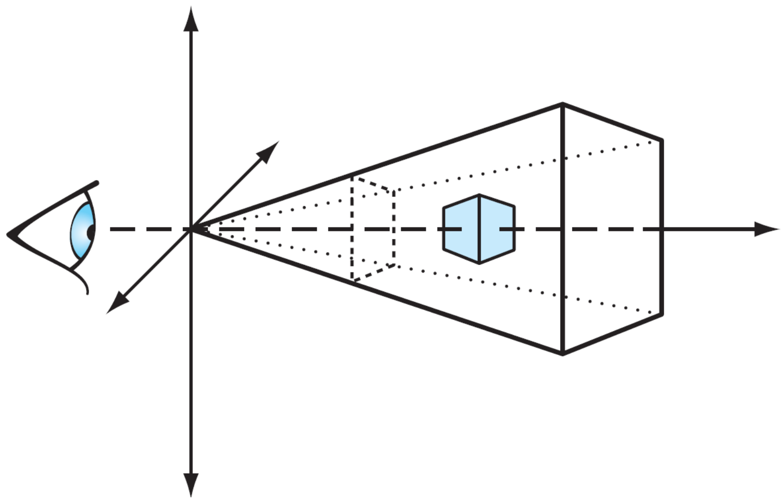
September 2018

Building blocks of realistic rendering

- Think of the experiment: “taking a picture”.
- What do we need to model it?
  - Camera (light sensor).
  - Light (light transport).
  - Objects (scene geometry).
  - Luminaires (light sources).
  - Materials (light scattering and absorption).
- To render an image, we need mathematical models for these elements.
- We can use very simple models:
  - Pinhole camera (perspective, lines from a point).
  - Ray optics (light propagates along straight lines).
  - Mathematical primitives (planes, spheres, triangles).
  - Practical lights (point, directional, spot).
  - Lambertian materials (constant reflectance).
- If we desire a high level of realism, more complicated models are required.

Camera (light sensor)

- Basic camera components: light sensitive area, processing unit, digital storage.
- The simplest camera model is a rectangle, which models the light sensitive area (the chip/film), placed in front of an eye point where light is gathered.



- We can use this modified pinhole camera model in two ways:
  - Follow rays from the eye point through the rectangle and onwards (ray tracing).
  - Project the geometry on the image plane and find the geometry that ends up in the rectangle (rasterization).

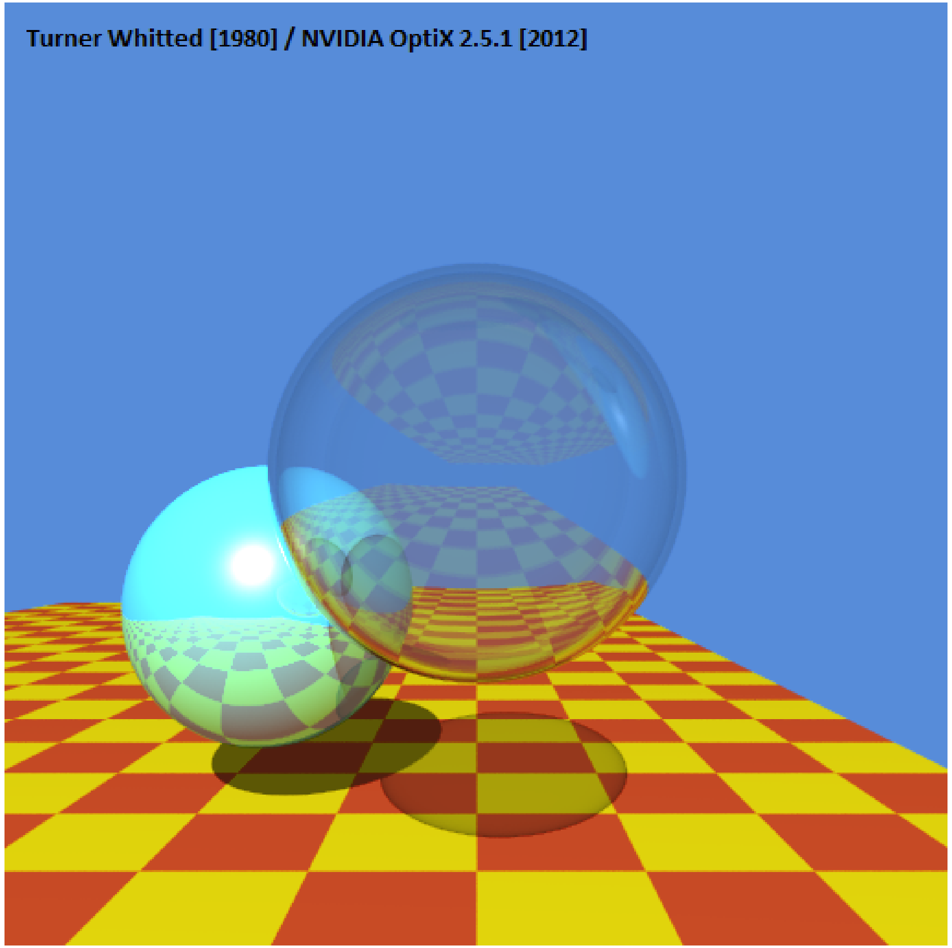
Ray tracing motivation

- 74 minutes in the original ray tracing paper of Turner Whitted [1980].
- More than 300 frames per second using GPU ray tracing in 2012.
- That's a 6 orders of magnitude speed-up! (Beats Moore's law.)

$$74 \cdot 60/(1/300) = 1{,}332{,}000$$

$$2^{(2012-1980)/2} = 2^{16} = 65{,}536$$

- Ray tracing is now widely spread in production rendering.
- Latest GPUs support this with dedicated ray tracing cores!

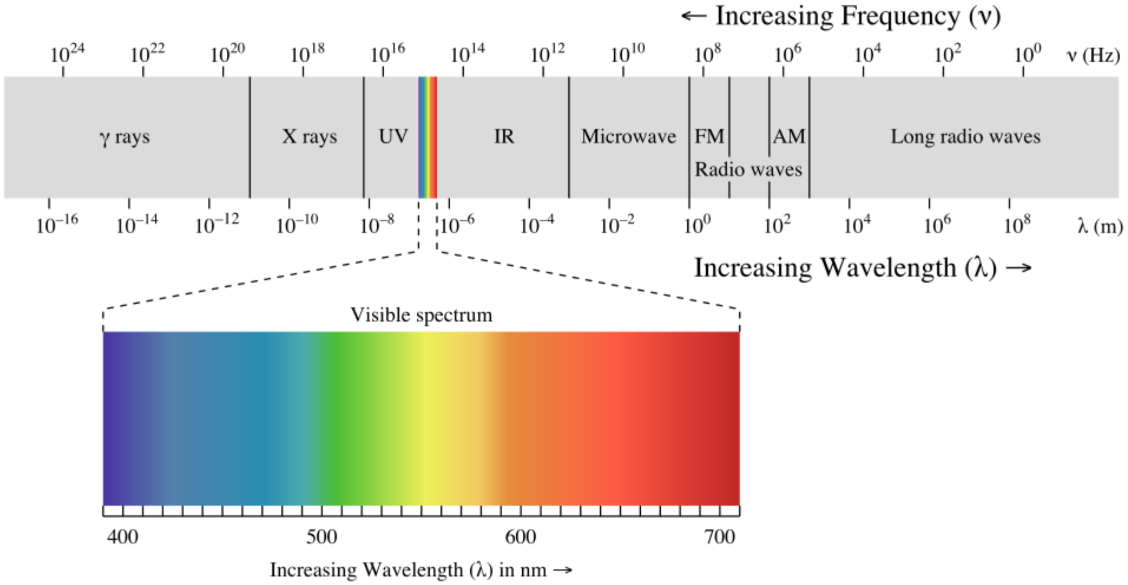


References

- Whitted, T. An improved illumination model for shaded display. *Communications of the ACM 23*(6), pp. 343–349, June 1980.
- 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 29*(4), pp. 66:1–66:13, July 2010.

Light (light transport)

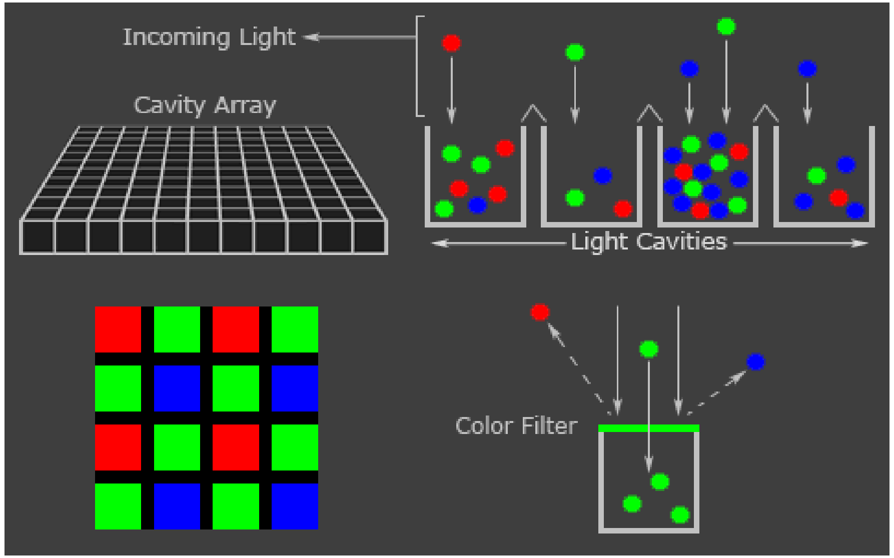
- Visible light is electromagnetic waves of wavelengths (λ) from 380 nm to 780 nm.



- Electromagnetic waves propagate as *rays of light* for $\lambda \to 0$.
- Rays of light follow the path of least time (Fermat): straight line segments.
- The parametrisation of a straight line in 3D is therefore a good, simple model for light transport: $\boldsymbol{r}(t) = \boldsymbol{o} + t\vec{\omega} \quad , \quad t \in [t_{\min}, t_{\max}] \quad , \quad t_{\max} > t_{\min} > 0\,.$

The light sensitive Charge-Coupled Device (CCD) chip

- A CCD chip is an array of light sensitive cavities.
- A digital camera therefore has a resolution $W \times H$ measured in number of pixels.
- A pixel corresponds to a small area on the chip.
- Several light sensitive cavities contribute to each pixel because the light measurement is divided into red, green, and blue.
- Conversion from this colour pattern to an RGB image is called demosaicing.

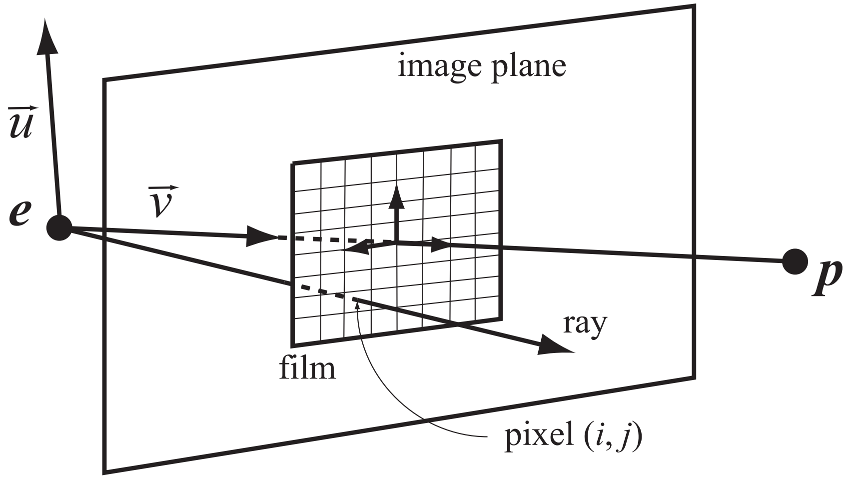


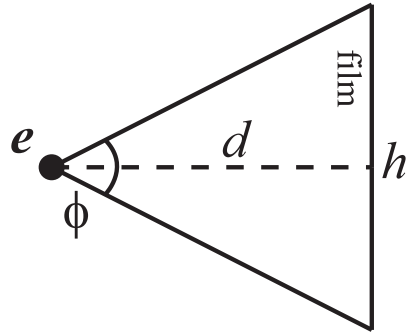
Ray generation

- Camera description:

Extrinsic parameters		Intrinsic parameters	
$\boldsymbol{e}$	Eye point	ϕ	Vertical field of view
$\boldsymbol{p}$	View point	d	Camera constant
$\vec{u}$	Up direction	W, H	Camera resolution

- Sketch of ray generation:





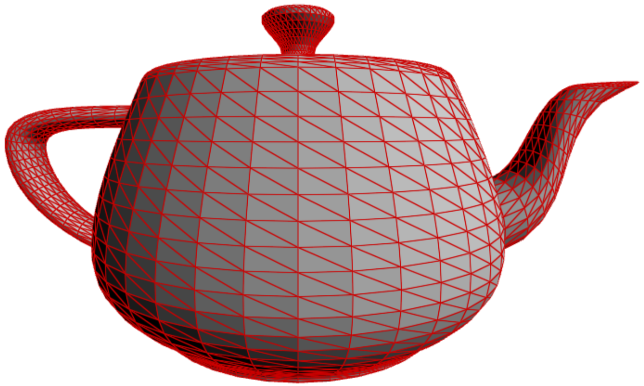
- Given pixel index (i, j), we find the direction $\vec{\omega}$ of a ray through that pixel.

Objects (scene geometry)

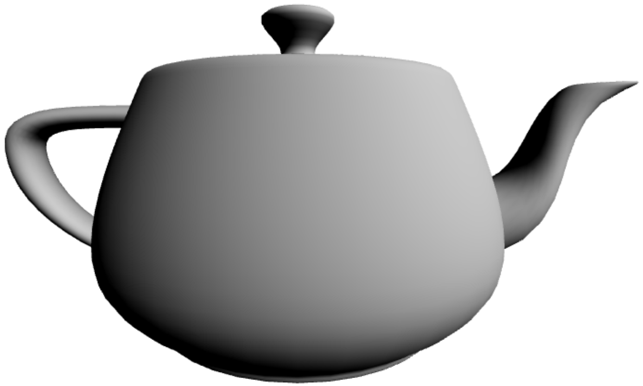
- Surface geometry is often modelled by a collection of triangles, where some of them share edges (a triangle mesh).
- Triangles provide a discrete representation of an arbitrary surface.

  Teapot example:

wireframe



faces

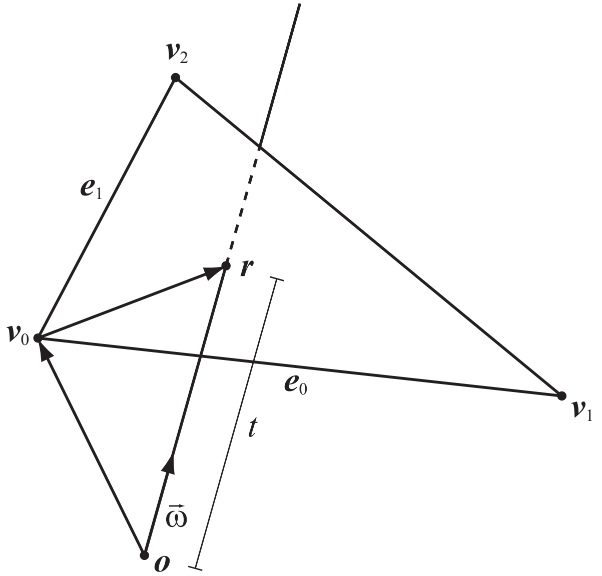


shaded

- Triangles are useful as they are defined by only three vertices.

  And ray-triangle intersection is simple.

Ray-triangle intersection



- Ray: $\boldsymbol{r}(t) = \boldsymbol{o} + t\,\vec{\omega},\;\; t \in [t_{\min}, t_{\max}]$.
- Triangle: $\boldsymbol{v}_0,\;\; \boldsymbol{v}_1,\;\; \boldsymbol{v}_2$.
- Edges and 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:

$$\begin{aligned} \boldsymbol{r}(u, v, w) &= u\boldsymbol{v}_0 + v\boldsymbol{v}_1 + w\boldsymbol{v}_2 = (1 - v - w)\boldsymbol{v}_0 + v\boldsymbol{v}_1 + w\boldsymbol{v}_2 \\ &= \boldsymbol{v}_0 + v\boldsymbol{e}_0 - w\boldsymbol{e}_1\,. \end{aligned}$$

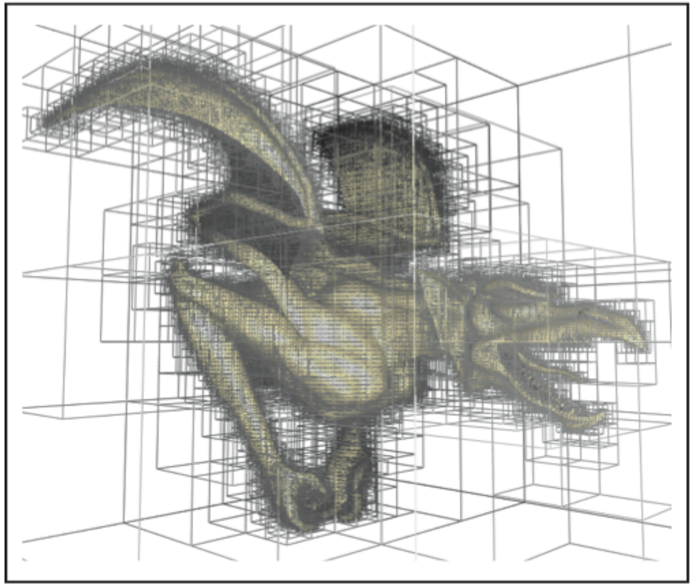
- The ray intersects the triangle's plane at $t' = \dfrac{(\boldsymbol{v}_0 - \boldsymbol{o}) \cdot \boldsymbol{n}}{\vec{\omega} \cdot \boldsymbol{n}}$.
- Find $\boldsymbol{r}(t') - \boldsymbol{v}_0$ and decompose it into portions along the edges $\boldsymbol{e}_0$ and $\boldsymbol{e}_1$ to get v and w. Then check

$$v \geq 0 \quad , \quad w \geq 0 \quad , \quad v + w \leq 1\;.$$

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 will still take more than 15 minutes.
- This is prohibitive. We need to find the relevant triangles.

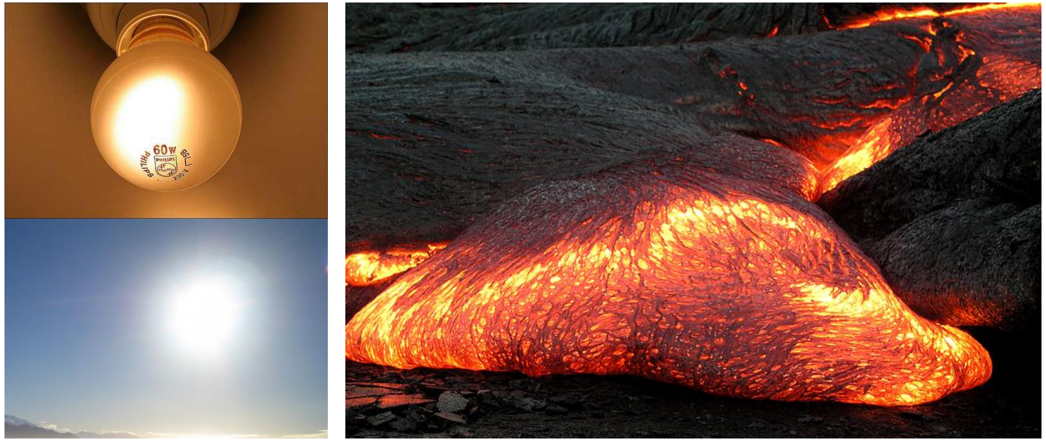
- Spatial data structures offer logarithmic complexity instead of linear.
- A million tests become twenty operations $\left(\log_2 10^6 \approx 20\right)$.
- 15 minutes become 20 milliseconds.



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

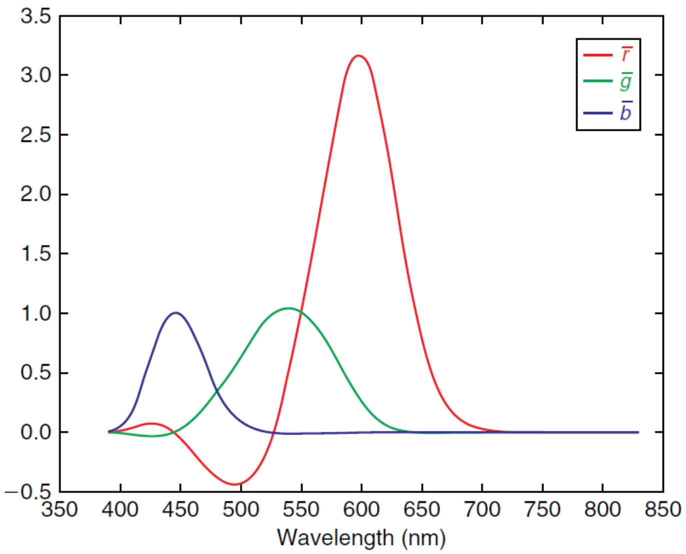
Luminaires (light sources)

- A light source is described by a spectrum of light $L_{e,\lambda}(\boldsymbol{x}, \vec{\omega}_o)$ which is emitted from each point on the emissive object.
- A simple model is a light source that from each point emits the same amount of light in all directions and at all wavelengths, $L_{e,\lambda} = \mathrm{const.}$
- The spectrum of heat-based light sources can be estimated using Planck's law of radiation. Examples:

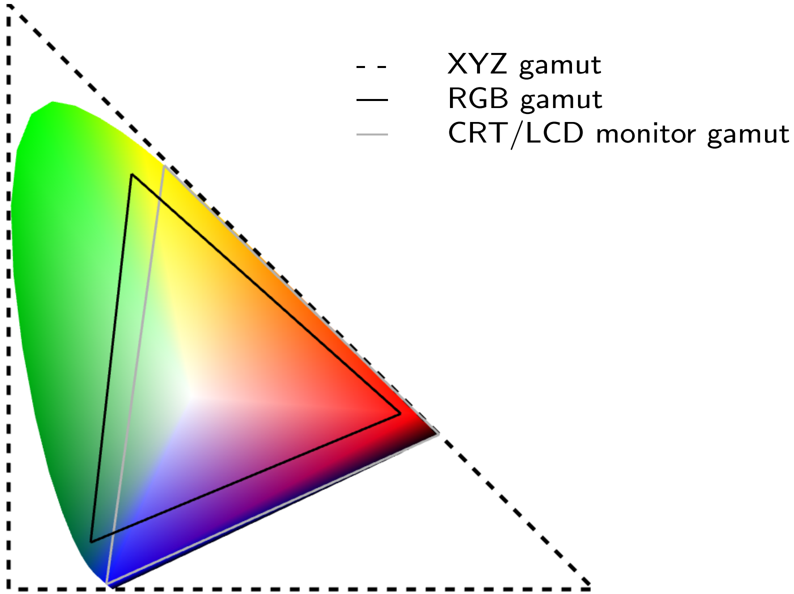


- The surface geometry of light sources is modelled in the same way as other geometry in the scene.

Colorimetry (spectrum to RGB)



CIE color matching functions



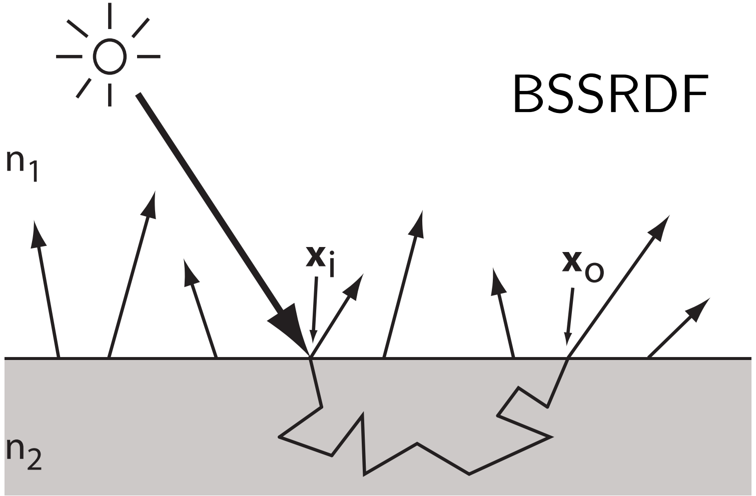
The chromaticity diagram

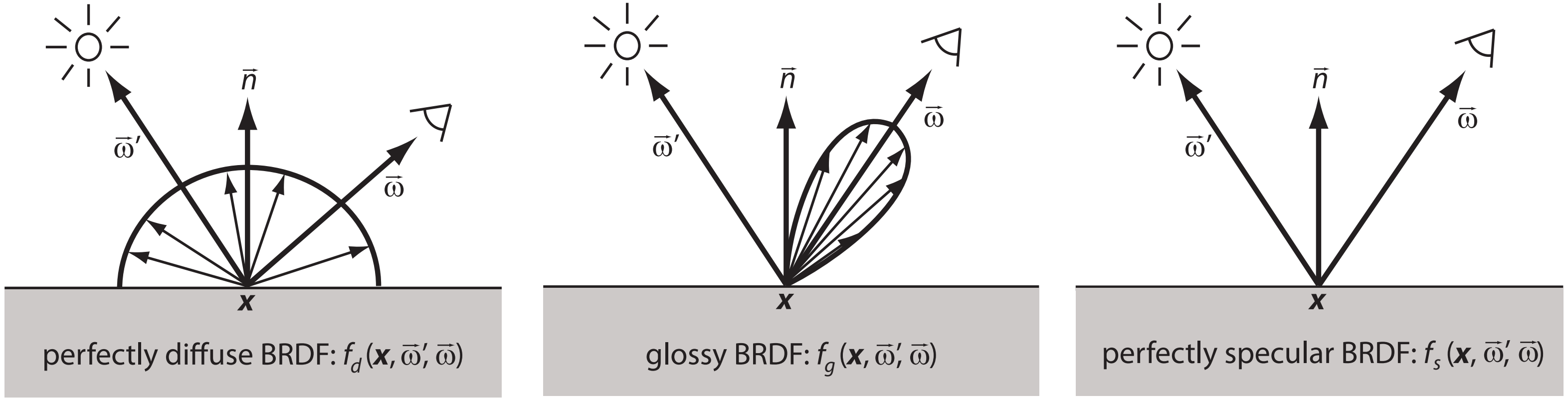
$$R = \int_{\mathscr{V}} C(\lambda)\bar{r}(\lambda)\,\mathrm{d}\lambda \quad , \quad G = \int_{\mathscr{V}} C(\lambda)\bar{g}(\lambda)\,\mathrm{d}\lambda \quad , \quad B = \int_{\mathscr{V}} C(\lambda)\bar{b}(\lambda)\,\mathrm{d}\lambda \ ,$$

where $\mathscr{V}$ is the interval of visible wavelengths and $C(\lambda)$ is the spectrum that we want to transform to RGB.

Materials (light scattering and absorption)

- Optical properties (index of refraction, $n(\lambda) = n'(\lambda) + i\, n''(\lambda)$).
- Reflectance distribution functions, $S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o)$.



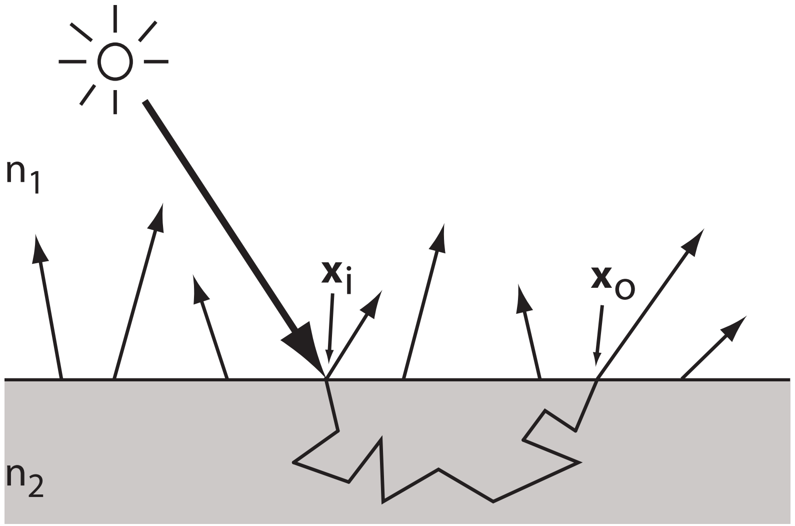


Subsurface scattering

- Behind light reflecting from surfaces [Nicodemus et al. 1977]:

$$\frac{\mathrm{d}L_r(\boldsymbol{x}_o, \vec{\omega}_o)}{\mathrm{d}\Phi_i(\boldsymbol{x}_i, \vec{\omega}_i)} = S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o)\,.$$

- $\mathrm{d}L_r$ is an element of reflected radiance.
- $\mathrm{d}\Phi_i$ is an element of incident flux.
- S (the BSSRDF) is the factor of proportionality.



- Using the definition of radiance $L = \dfrac{\mathrm{d}^2\Phi}{\cos\theta\,\mathrm{d}A\,\mathrm{d}\omega}$, we have

$$L_r(\boldsymbol{x}_o, \vec{\omega}_o) = \int_A \int_{2\pi} S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) L_i(\boldsymbol{x}_i, \vec{\omega}_i) \cos\theta_i\,\mathrm{d}\omega_i\,\mathrm{d}A_i\,.$$

References

- Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W., and Limperis, T. Geometrical considerations and nomenclature for reflectance. Tech. rep., National Bureau of Standards (US), 1977.

The rendering equation

- When rendering surfaces, the equation we evaluate is [Kajiya 1986, Jensen et al. 2001]

$$L_o(\mathbf{x}_o, \vec{\omega}_o) = L_e(\mathbf{x}_o, \vec{\omega}_o) + \int_A \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o) L_i(\mathbf{x}_i, \vec{\omega}_i) \cos\theta_i \, \mathrm{d}\omega_i \, \mathrm{d}A_i \quad ,$$

where

L_o is outgoing radiance,
L_e is emitted radiance,
L_i is incoming radiance,
$\mathbf{x}_i, \mathbf{x}_o$ are the surface positions where light is incoming and outgoing,
$\vec{\omega}_o$ is the direction of the outgoing radiance,
$\vec{\omega}_i$ is the direction toward the light source,
S is the bidirectional scattering-surface reflectance distribution function (BSSRDF),
$\mathrm{d}\omega_i$ is an element of solid angle,
θ_i is the angle between $\vec{\omega}_i$ and the surface normal $\vec{n}$ at $\mathbf{x}_i$, such that $\cos\theta_i = \vec{\omega}_i \cdot \vec{n}$.

References

- Kajiya, J. The Rendering Equation. *Computer Graphics (Proceedings of ACM SIGGRAPH 86) 20*(4), pp. 143–150, 1986.
- Jensen, H. W., Marschner, S., Levoy, M., and Hanrahan, P. A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH 2001*, pp. 511–518, 2001.

Monte Carlo integration

- The law of large numbers:

$$\Pr\left\{\frac{1}{N}\sum_{i=1}^{N} f(X_i) \to E\{f(X)\}\right\} = 1 \quad \text{for} \quad N \to \infty \ .$$

"It is certain that the estimator goes to the expected value as the number of samples goes to infinity."

- Approximating an arbitrary integral by stochastic sampling:

$$F = \int_A f(x)\,\mathrm{d}x = \int_A \frac{f(x)}{\mathrm{pdf}(x)}\,\mathrm{pdf}(x)\,\mathrm{d}x = E\left\{\frac{f(X)}{\mathrm{pdf}(X)}\right\}.$$

Using the law of large numbers, we have the Nth estimator

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{\mathrm{pdf}(X_i)} \ ,$$

where X_i are sampled on A and $\mathrm{pdf}(x) > 0$ for all $x \in A$.

An estimator for rendering

- The equation for reflected radiance:

$$L_r(\boldsymbol{x}_o, \vec{\omega}_o) = \int_A \int_{2\pi} S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) L_i(\boldsymbol{x}_i, \vec{\omega}_i) \cos\theta_i \,\mathrm{d}\omega_i \,\mathrm{d}A_i \,.$$

- The Monte Carlo estimator:

$$L_{r,N,M}(\boldsymbol{x}_o, \vec{\omega}_o) = \frac{1}{NM} \sum_{p=1}^{M} \sum_{q=1}^{N} \frac{S(\boldsymbol{x}_{i,p}, \vec{\omega}_{i,q}; \boldsymbol{x}_o, \vec{\omega}_o) L_i(\boldsymbol{x}_{i,p}, \vec{\omega}_{i,q}) \cos\theta_i}{\mathrm{pdf}(\boldsymbol{x}_{i,p}) \mathrm{pdf}(\vec{\omega}_{i,q})} \quad .$$

- Common direction sampling pdf (cosine-weighted hemisphere):

$$\mathrm{pdf}(\vec{\omega}_{i,q}) = \frac{\vec{\omega}_{i,q} \cdot \vec{n}_i}{\pi} = \frac{\cos\theta_i}{\pi} \quad .$$

- Common area sampling pdf (triangle mesh):

$$\mathrm{pdf}(\boldsymbol{x}_{i,p}) = \mathrm{pdf}(\triangle)\mathrm{pdf}(\boldsymbol{x}_{i,p,\triangle}) = \frac{A_\triangle}{A_\ell} \frac{1}{A_\triangle} = \frac{1}{A_\ell} \quad .$$

Tracing rays through a scene

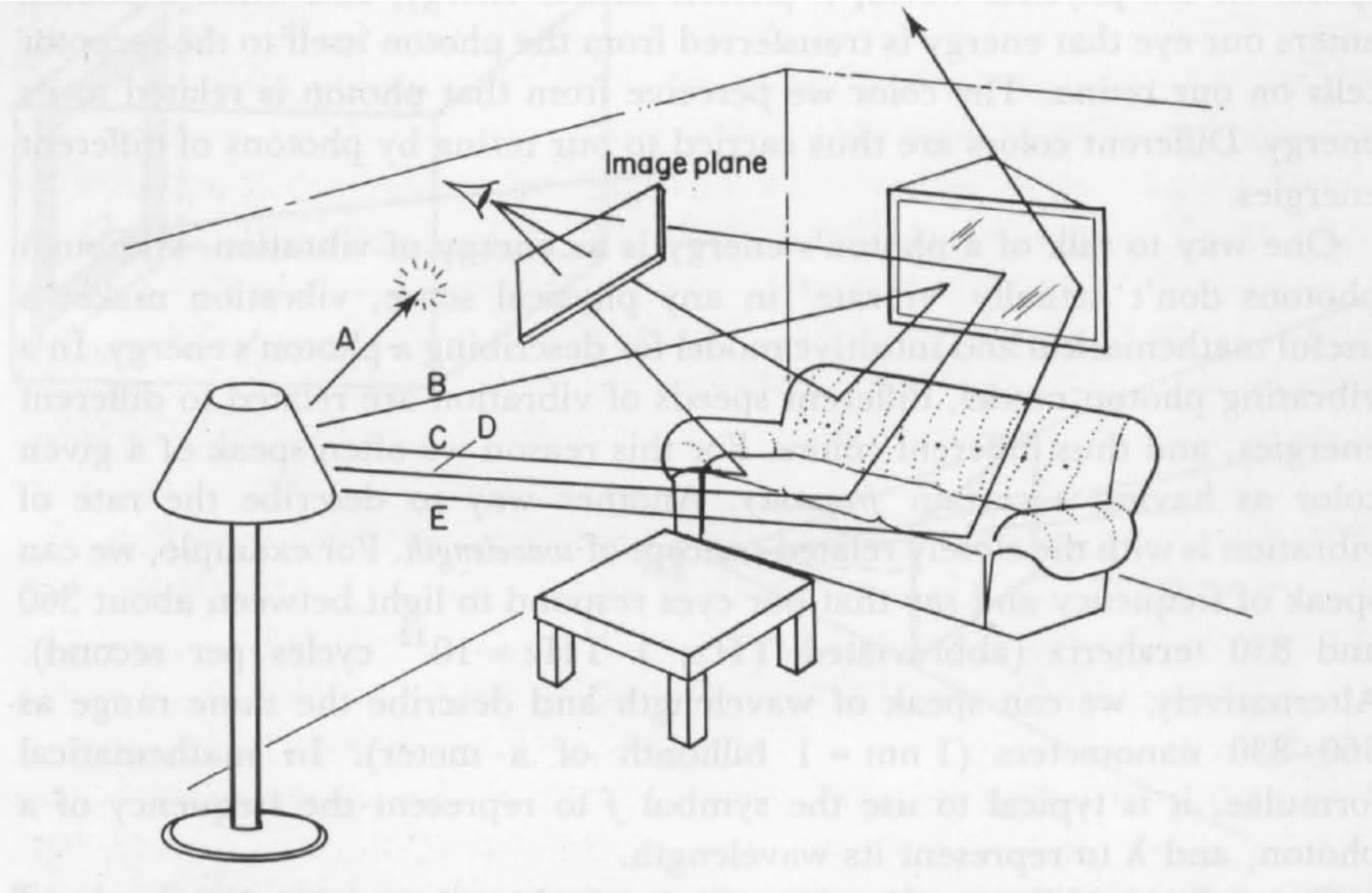


Fig. 5. Some light rays (like A and E) never reach the image plane at all. Others follow simple or complicated routes.

References

- Glassner, A. An Overview of Ray Tracing. In *An Introduction to Ray Tracing*, Chapter 1, pp. 1–17, Morgan Kaufmann, 1989.

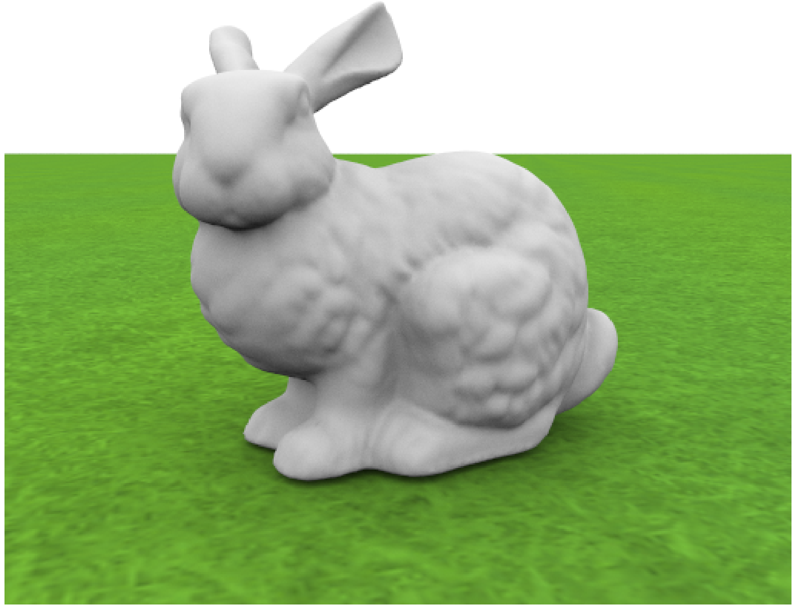
Progressive unidirectional path tracing

1. Generate rays from the eye through pixel positions
2. Trace the rays and evaluate the rendering equation for each ray.
3. Randomize the position within the pixel area to Monte Carlo integrate (measure) the radiance arriving in a pixel.

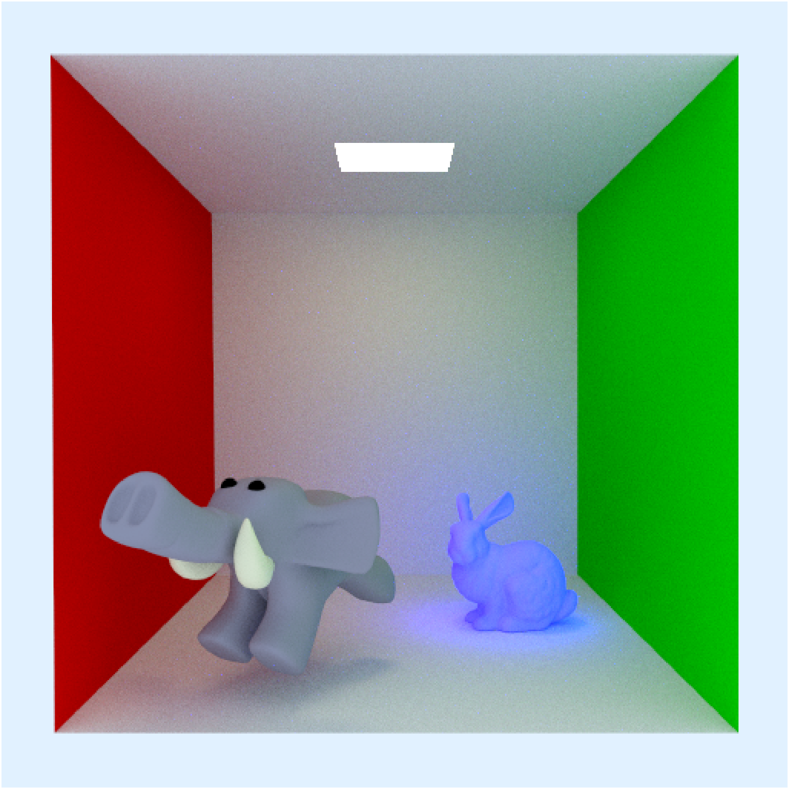
- Path tracing is the idea of using $N = 1$ in the Monte Carlo estimators for the reflected radiance to generate a path instead of a tree.
- Noise is reduced by progressive updates of the measurement.
- Update the rendering result in a pixel L_j after rendering a new frame with result L_{new} using
$$L_{j+1} = \frac{L_{\text{new}} + jL_j}{j+1} \, .$$
- Progressive (stop and go) rendering is convenient for several reasons:
  - No need to start over.
  - Result can be stored and refined later if need be.
  - Convergence can be inspected during progressive updates.

Sampling a cosine-weighted hemisphere (ambient occlusion)

- Material:
$$S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) = \frac{\rho_d(\boldsymbol{x}_o)}{\pi}\delta(\boldsymbol{x}_o - \boldsymbol{x}_i)\,.$$
- Sampler:
$$\mathrm{pdf}(\vec{\omega}_{i,q}) = \frac{\vec{\omega}_{i,q} \cdot \vec{n}_i}{\pi} = \frac{\cos\theta_i}{\pi}\,.$$
- Estimator:
$$\begin{aligned} &L_{r,N}(\boldsymbol{x}_o, \vec{\omega}_o) \\ &= \frac{1}{N}\sum_{q=1}^{N} \frac{\rho_d(\boldsymbol{x}_o)}{\pi} \frac{L_i(\boldsymbol{x}_o, \vec{\omega}_{i,q})\cos\theta_i}{\mathrm{pdf}(\vec{\omega}_{i,q})} \\ &= \rho_d(\boldsymbol{x}_o)\frac{1}{N}\sum_{q=1}^{M} L_i(\boldsymbol{x}_o, \vec{\omega}_{i,q})\,. \end{aligned}$$



Sampling a triangle mesh (area lights, soft shadows)



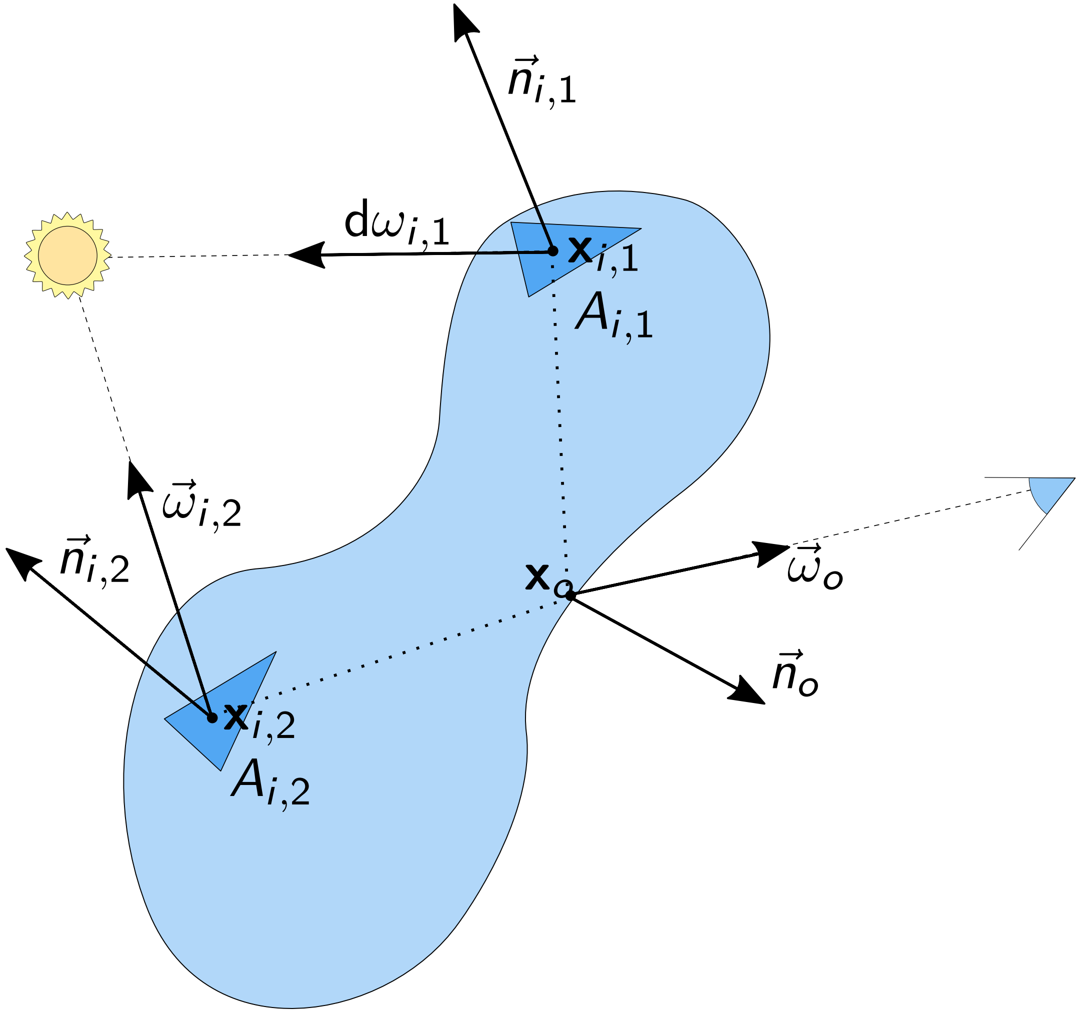
- Material:
  $$S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) = \frac{\rho_d(\boldsymbol{x}_o)}{\pi}\delta(\boldsymbol{x}_o - \boldsymbol{x}_i)\,.$$
- Sampler:
  $$\vec{\omega}_{i,q} = \frac{\boldsymbol{x}_{\ell,q} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{\ell,q} - \boldsymbol{x}_i\|}$$
  $$\mathrm{pdf}(\boldsymbol{x}_{\ell,q}) = \mathrm{pdf}(\triangle)\mathrm{pdf}(\boldsymbol{x}_{\ell,q,\triangle}) = \frac{A_\triangle}{A_\ell}\frac{1}{A_\triangle}\,.$$
- Estimator:
  $$L_{r,N}(\boldsymbol{x}_o, \vec{\omega}_o)$$
  $$= \frac{\rho_d(\boldsymbol{x}_o)}{\pi}\frac{1}{N}\sum_{q=1}^{N} L_e(\boldsymbol{x}_{\ell,q}, -\vec{\omega}_{i,q})V(\boldsymbol{x}_{\ell,q}, \boldsymbol{x}_o)\frac{\cos\theta_i \cos\theta_\ell}{\|\boldsymbol{x}_{\ell,q} - \boldsymbol{x}_i\|^2}A_\ell\,.$$

Sampling for subsurface scattering

- Material:
  $S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) = \dots\ .$

- Sampler:

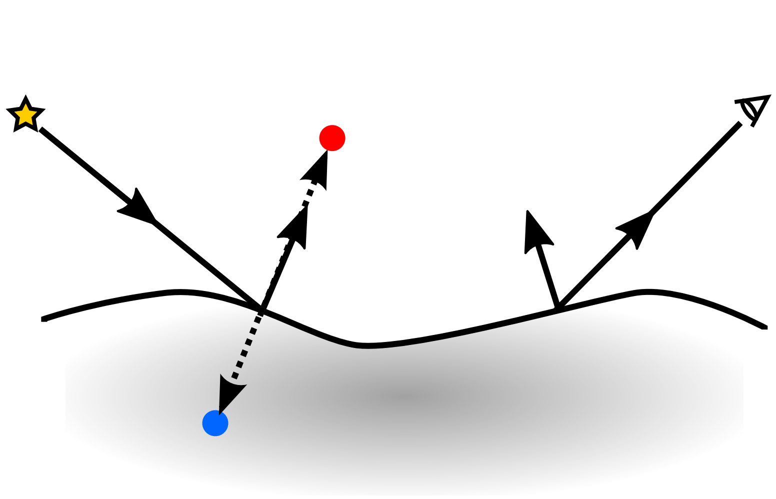
$$\begin{aligned}
\mathrm{pdf}(\vec{\omega}_{i,q}) &= \frac{\vec{\omega}_{i,q}\cdot\vec{n}_i}{\pi} = \frac{\cos\theta_i}{\pi}\,.\\
\mathrm{pdf}(\boldsymbol{x}_{i,p}) &= \mathrm{pdf}(\triangle)\mathrm{pdf}(\boldsymbol{x}_{i,p,\triangle})\\
&= \frac{A_\triangle}{A_\ell}\frac{1}{A_\triangle} = \frac{1}{A_\ell}\,.
\end{aligned}$$



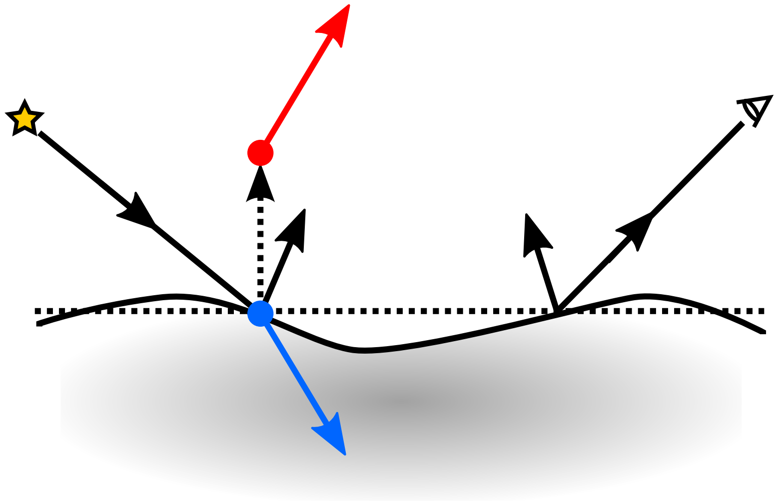
References

- Frisvad, J. R., Hachisuka, T., and Kjeldsen, T. K. Directional dipole model for subsurface scattering. *ACM Transactions on Graphics 34*(1), pp. 5:1–5:12, November 2014. Presented at SIGGRAPH 2015.
- Dal Corso, A., and Frisvad, J. R. Point cloud method for rendering BSSRDFs. Technical Report, Technical University of Denmark, 2018.

Analytical models for subsurface scattering



standard dipole



directional dipole

$$S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) = T_{12}(\vec{\omega}_i)(S_1 + S_d(\|\boldsymbol{x}_o - \boldsymbol{x}_i\|))\,T_{21}(\vec{\omega}_o)\,.$$

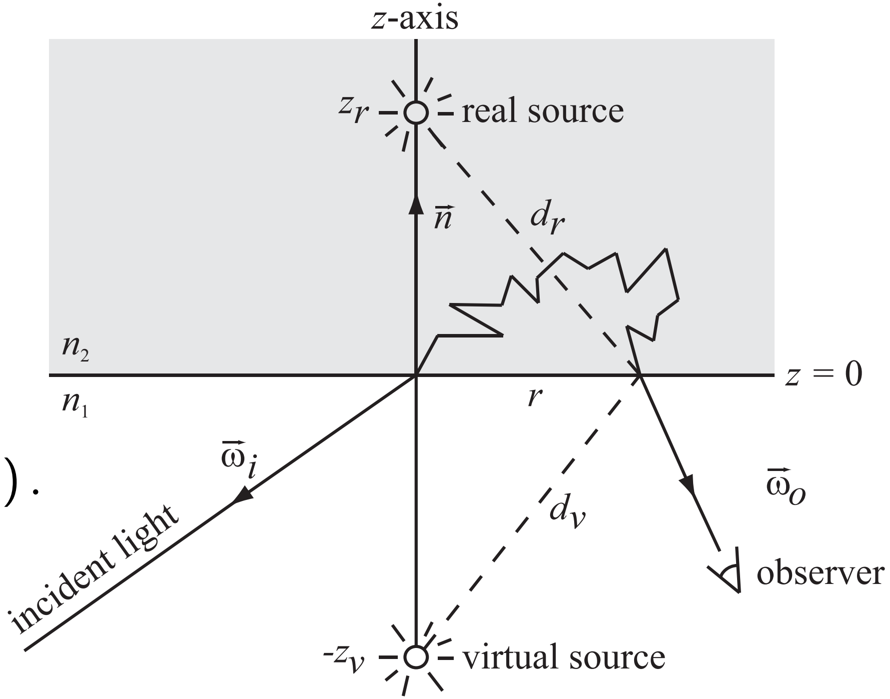
$$S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) = T_{12}(\vec{\omega}_i)(S_{\delta E} + S_d(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o))\,T_{21}(\vec{\omega}_o)\,.$$

- Directions ($\vec{\omega}_i$, $\vec{\omega}_o$) also require surface normals ($\vec{n}_i$, $\vec{n}_o$) to get angles (θ_i, θ_o).
- T_{12} and T_{21} are Fresnel transmittances.
- S_1 and $S_{\delta E}$ are fully directional (depend on $\boldsymbol{x}_i$, $\vec{\omega}_i$, $\boldsymbol{x}_o$, $\vec{\omega}_o$, and normals).

Diffusive part of the standard dipole

$$S_d(r) = \frac{\alpha'}{4\pi^2}\left(\frac{(z_r(1+\sigma_{\text{tr}}d_r)e^{-\sigma_{\text{tr}}d_r}}{d_r^3} + \frac{z_v(1+\sigma_{\text{tr}}d_v)e^{-\sigma_{\text{tr}}d_v}}{d_v^3}\right).$$

- Distances:
  - $z_r = \Lambda$.
  - $z_v = \Lambda + 4AD$.
  - $d_r(r) = \sqrt{r^2 + z_r^2}$.
  - $d_v(r) = \sqrt{r^2 + z_v^2}$.
- Optical properties ($\eta = n_2/n_1$, σ_s, σ_a, g):
  - Reduced scattering coefficient: $\sigma_s' = \sigma_s(1-g)$.
  - Reduced extinction coefficient: $\sigma_t' = \sigma_s' + \sigma_a$.
  - Reduced scattering albedo: $\alpha' = \sigma_s'/\sigma_t'$.
  - Transport mean free path: $\Lambda = 1/\sigma_t'$.
  - Diffusion coefficient: $D = \Lambda/3$.
  - Transport coefficient: $\sigma_{\text{tr}} = \sqrt{\sigma_a/D}$.
  - Reflection parameter: $A(\eta)$ (ratio of polynomial fits).



Diffusive part of the directional dipole

$$S_d(\mathbf{x}_i, \vec{\omega}_i; \mathbf{x}_o) = S'_d(\mathbf{x}_o - \mathbf{x}_i, \vec{\omega}_{12}, d_r) - S'_d(\mathbf{x}_o - \mathbf{x}_v, \vec{\omega}_v, d_v) \ \ ,$$

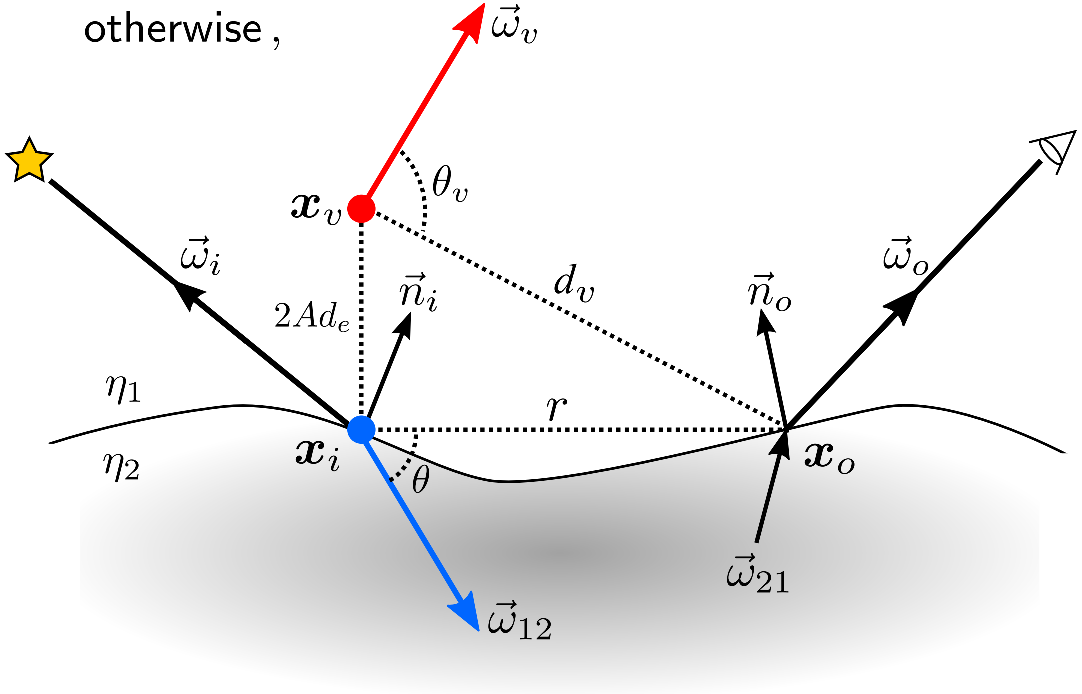
- Real source:
  - $\vec{\omega}_{12} = \eta^{-1}((\vec{\omega}_i \cdot \vec{n}_i)\vec{n}_i - \vec{\omega}_i) - \vec{n}_i\sqrt{1 - \eta^{-2}(1 - (\vec{\omega}_i \cdot \vec{n}_i)^2)}\,.$
  - $d_r^2 = \begin{cases} \|\mathbf{x}_o - \mathbf{x}_i\|^2 + D\mu_0(D\mu_0 - 2d_e\cos\beta) & \text{for } \mu_0 > 0 \\ \|\mathbf{x}_o - \mathbf{x}_i\|^2 + 1/(3\sigma_t)^2 & \text{otherwise}\,, \end{cases}$

    with $\mu_0 = \cos\theta_0 = -\vec{n}_o \cdot \vec{\omega}_{12}$

    and $\cos\beta = -\sqrt{\frac{r^2 - (\mathbf{x}\cdot\omega_{12})^2}{r^2 + d_e^2}}\,.$
- Virtual source:
  - Modified normal:

    $\vec{n}_i^* = \frac{\mathbf{x}_o - \mathbf{x}_i}{\|\mathbf{x}_o - \mathbf{x}_i\|} \times \frac{\vec{n}_i \times (\mathbf{x}_o - \mathbf{x}_i)}{\|\vec{n}_i \times (\mathbf{x}_o - \mathbf{x}_i)\|}\,,$

    or $\vec{n}_i^* = \vec{n}_i$ if $\mathbf{x}_o = \mathbf{x}_i$.
  - $\mathbf{x}_v = \mathbf{x}_i + 2Ad_e\vec{n}_i^*\,, \ \ d_v = |\mathbf{x}_v - \mathbf{x}_i|\,, \ \ \vec{\omega}_v = \vec{\omega}_{12} - 2(\vec{\omega}_{12} \cdot \vec{n}_i^*)\vec{n}_i^*\,.$

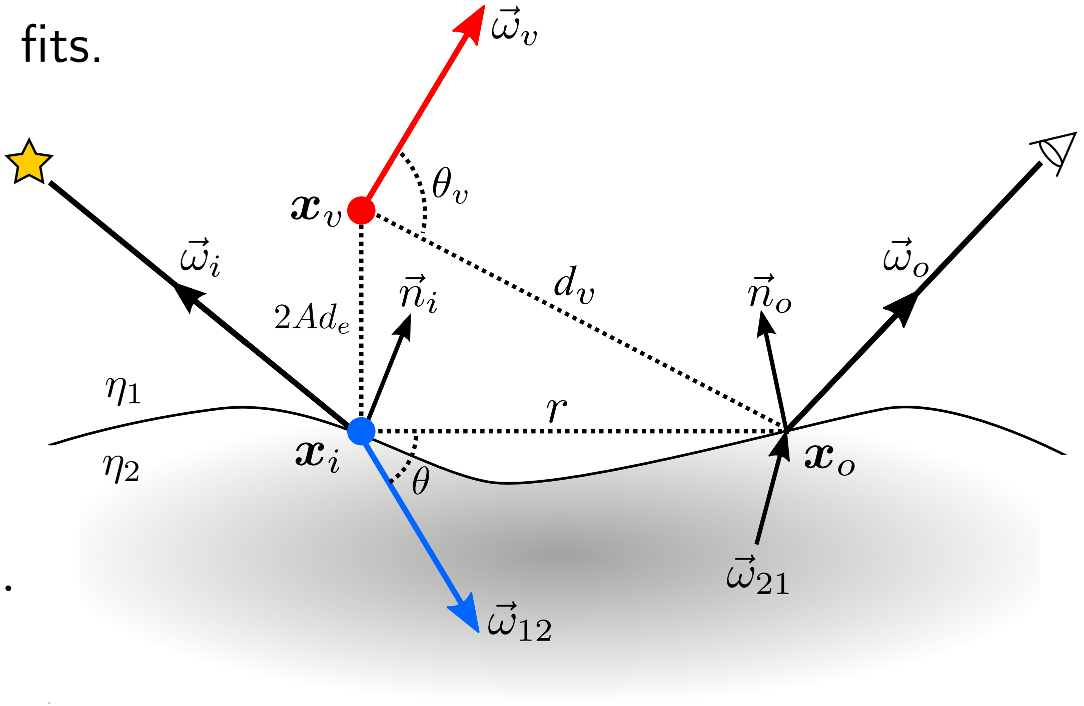


Directional subsurface scattering when disregarding the boundary

$$S'_d(\mathbf{x}, \vec{\omega}_{12}, r) = \frac{1}{4C_\phi(1/\eta)} \frac{1}{4\pi^2} \frac{e^{-\sigma_{\mathrm{tr}} r}}{r^3} \Bigg[C_\phi(\eta) \left(\frac{r^2}{D} + 3(1 + \sigma_{\mathrm{tr}} r)\, \mathbf{x} \cdot \vec{\omega}_{12} \right)$$
$$- C_{\mathbf{E}}(\eta) \left(3D(1 + \sigma_{\mathrm{tr}} r)\, \vec{\omega}_{12} \cdot \vec{n}_o - \left((1 + \sigma_{\mathrm{tr}} r) + 3D \frac{3(1+\sigma_{\mathrm{tr}} r) + (\sigma_{\mathrm{tr}} r)^2}{r^2} \mathbf{x} \cdot \vec{\omega}_{12} \right) \mathbf{x} \cdot \vec{n}_o \right) \Bigg],$$

where $C_\phi(\eta)$ and $C_{\mathbf{E}}(\eta)$ are polynomial fits.

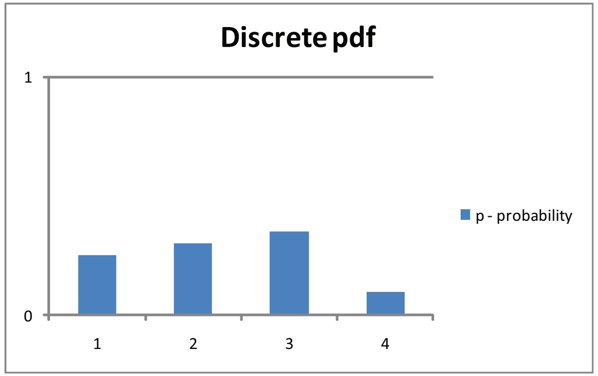
- Additional dependencies:
  - Normal: $\vec{n}_o$.
  - Optical properites: η, D, σ_{tr} .
- Note the exponential term: $e^{-\sigma_{\mathrm{tr}} r}$.
- Normal incidence: $\vec{\omega}_{12} \cdot \vec{n}_o = \pm 1$.
- Plane (half-space): $\mathbf{x} \cdot \vec{n}_o \approx 0$.
- normal incidence on plane: $\mathbf{x} \cdot \vec{\omega}_{12} \approx 0$.
- $r \to \|\mathbf{x}_o - \mathbf{x}_i\|$ for $\|\mathbf{x}_o - \mathbf{x}_i\| \to \infty$.

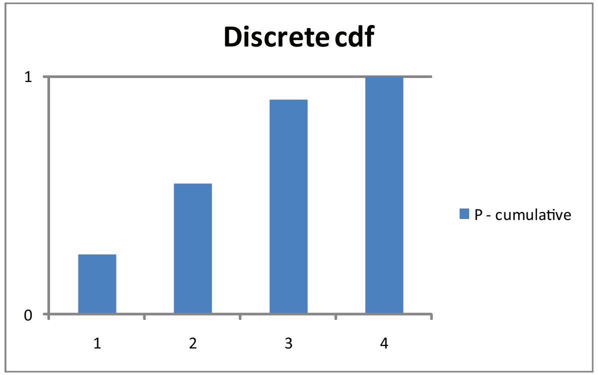


Rejection control

- The exponential attenuation $e^{-\sigma_{\mathrm{tr}} d}$ appears in all analytical BSSRDFs and $d \to \|\boldsymbol{x}_o - \boldsymbol{x}_i\|$ for $\|\boldsymbol{x}_o - \boldsymbol{x}_i\| \to \infty$.
- We should exploit this.
- Russian roulette:

  sample $\xi \in [0, 1]$ uniformly;
  if $(\xi < P_1)$
      call event 1;
      divide by p_1;
  else if $(\xi < P_2)$
      call event 2;
      divide by p_2;
  else if $(\xi < P_3)$
      ...
  else if $(\xi < P_4)$
      ...





- When sampling $\boldsymbol{x}_i$, use Russian roulette with $p_1(\boldsymbol{x}_i) = P_1(\boldsymbol{x}_i) = e^{-\sigma_{\mathrm{tr}}\|\boldsymbol{x}_o - \boldsymbol{x}_i\|}$ to accept or reject a sample.

Progressive rendering of subsurface scattering

- The equation for reflected radiance:

$$L_r(\mathbf{x}_o, \vec{\omega}_o) = \int_A \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o) L_i(\mathbf{x}_i, \vec{\omega}_i) \cos\theta_i \, \mathrm{d}\omega_i \, \mathrm{d}A_i \,.$$

- Initialize a frame by storing samples of transmitted light.
- For each sample:
  - Sample a point $(\mathbf{x}_{i,p})$ on the surface of the scattering material by sampling a random triangle and then a random point in the triangle: $\mathrm{pdf}(\mathbf{x}_{i,p}) = 1/A_\ell$.
  - Sample a ray direction $\vec{\omega}_{i,q}$ using a cosine-weighted hemisphere and trace they ray to collect incident light L_i: $\mathrm{pdf}(\vec{\omega}_{i,q}) = \cos\theta_i/\pi$.
  - Use $\vec{\omega}_{i,q}$ to find the direction of the transmitted/refracted ray and the Fresnel transmittance T_{12}.
  - Store the transmitted radiance: $L_t = \dfrac{T_{12} L_i \cos\theta_i}{\mathrm{pdf}(\mathbf{x}_{i,p})\mathrm{pdf}(\vec{\omega}_{i,q})} = T_{12} L_i \pi A_\ell$.

Progressive rendering of subsurface scattering

- The equation for reflected radiance:

$$L_r(\boldsymbol{x}_o, \vec{\omega}_o) = \int_A \int_{2\pi} S(\boldsymbol{x}_i, \vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) L_i(\boldsymbol{x}_i, \vec{\omega}_i) \cos\theta_i \, \mathrm{d}\omega_i \, \mathrm{d}A_i \, .$$

- For a ray hitting $\boldsymbol{x}_o$ with direction $-\vec{\omega}_o$:
  - Compute Fresnel transmittance T_{21} of the ray refracting from inside to $\vec{\omega}_o$.
  - Loop through the NM samples using exponential distance attenuation as the probability of acceptance in a Russian roulette (rejection control).
  - Use L_t of accepted samples and T_{21} together with the analytical expression for S to Monte Carlo integrate the rendering equation.
  - The Monte Carlo estimator for the diffusive part is (we use $N = 1$):

$$\begin{aligned} L_{d,N,M}(\boldsymbol{x}_o, \vec{\omega}_o) &= \frac{1}{NM} \sum_{p=1}^{M} \sum_{q=1}^{N} \frac{S(\boldsymbol{x}_{i,p}, \vec{\omega}_{i,q}; \boldsymbol{x}_o, \vec{\omega}_o) L_i(\boldsymbol{x}_{i,p}, \vec{\omega}_{i,q}) \cos\theta_i}{\mathrm{pdf}(\boldsymbol{x}_{i,p}) \mathrm{pdf}(\vec{\omega}_{i,q})} \\ &= \frac{T_{21}}{NM} \sum_{p=1}^{M} \sum_{q=1}^{N} \frac{S_d(\boldsymbol{x}_{i,p}, \vec{\omega}_{i,q}; \boldsymbol{x}_o, \vec{\omega}_o) T_{12} L_i(\boldsymbol{x}_{i,p}, \vec{\omega}_{i,q}) \pi A_\ell}{e^{-\sigma_{\mathrm{tr}} \|\boldsymbol{x}_o - \boldsymbol{x}_{i,p}\|}} \left[\xi < e^{-\sigma_{\mathrm{tr}} \|\boldsymbol{x}_o - \boldsymbol{x}_{i,p}\|} \right] . \end{aligned}$$

The input challenge

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

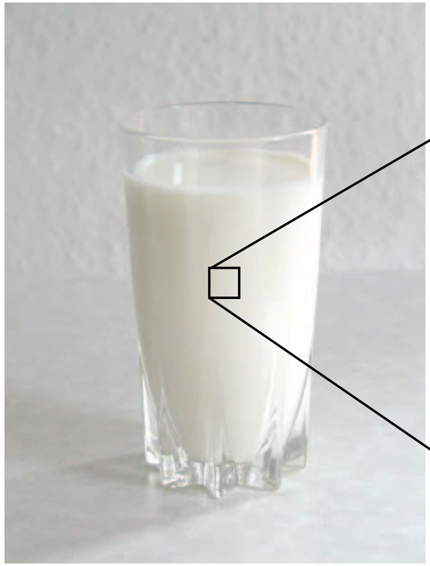
- How do we get plausible input parameters?
  - Modelling (example: light scattering by particles).
  - Measuring (example: diffuse reflectance spectroscopy).

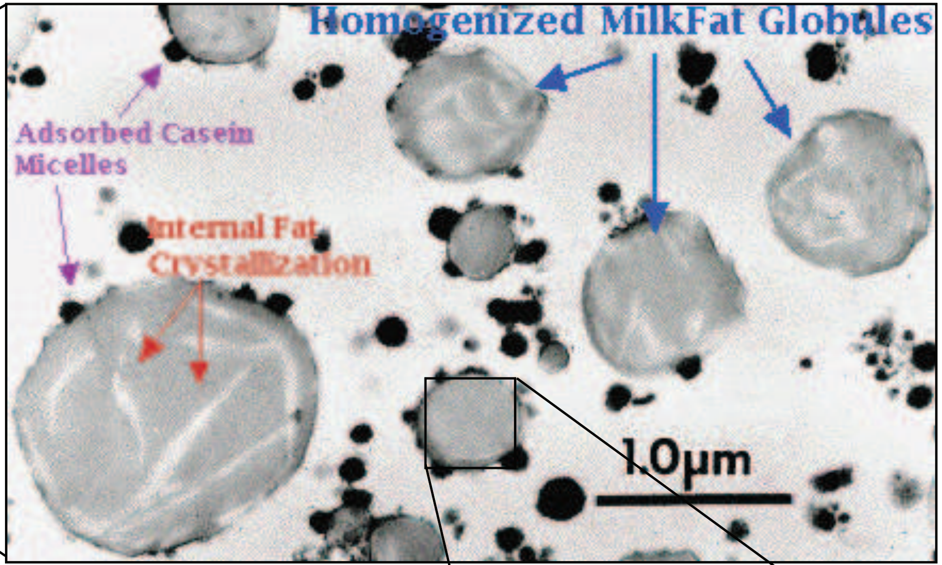
- Suppose we would like to go beyond visual comparison.
- How do we assess the appearance produced by a given set of input parameters?
  - Full digitization of a scene.
  - Reference photographs from known camera positions.
  - Pixelwise comparison of renderings with photographs.

Light-material interaction in a volume

- Events as we move along a ray traveling through a translucent material:
  - 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)
- The optical properties of such a medium are
  - n index of refraction ($n = n' + i\,n''$): speed of light (n') and absorption (n'').
  - σ_s scattering coefficient [m^{-1}]: amount of scattering as we move along a ray.
  - p phase function [sr^{-1}]: directional distribution of the scattered light.
  - ε emission properties [$\mathrm{W\,sr^{-1}m^{-3}}$] (radiance per meter).
- And then surface roughness. Other properties are derived quantities. For example:
  - σ_a absorption coefficient [m^{-1}]: $\sigma_a = 4\pi n''/\lambda$.
  - σ_t extinction coefficient [m^{-1}]: $\sigma_t = \sigma_a + \sigma_s$.
  - g asymmetry parameter: $g = \int_{4\pi} p(\vec{\omega}_i, \vec{\omega}_o)(\vec{\omega}_i \cdot \vec{\omega}_o)\,\mathrm{d}\omega_i$.
- How to compute scattering properties from the particle composition of a material?

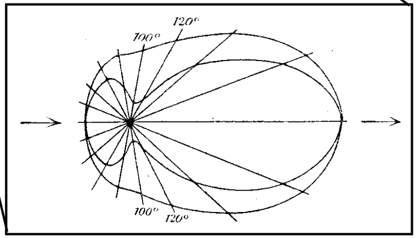
Modelling appearance using light scattering by particles





Courtesy of University of Guelph

Lorenz–Mie theory provides the link



References

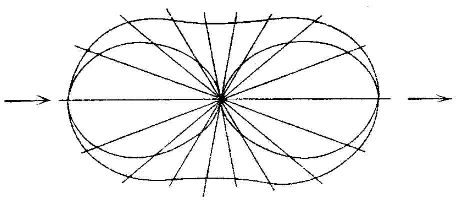
- Lorenz, L. Lysbevægelser i og uden for en af plane Lysbølger belyst Kugle. *Det kongelige danske Videnskabernes Selskabs Skrifter*. 6. Række, naturvidenskabelig og mathematisk Afdeling VI. pp. 1–62, 1890.
- Mie, G. Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. *Annalen der Physik 25*(3), pp. 377–445. IV. Folge. 1908.

Scattering by spherical particles

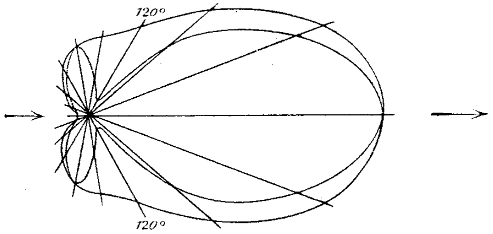
- The Lorenz-Mie theory:

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

- a_n and b_n are Lorenz-Mie coefficients (of particle size, refractive indices, and wavelength).
- π_n and τ_n are spherical functions associated with the Legendre polynomials.



small particle



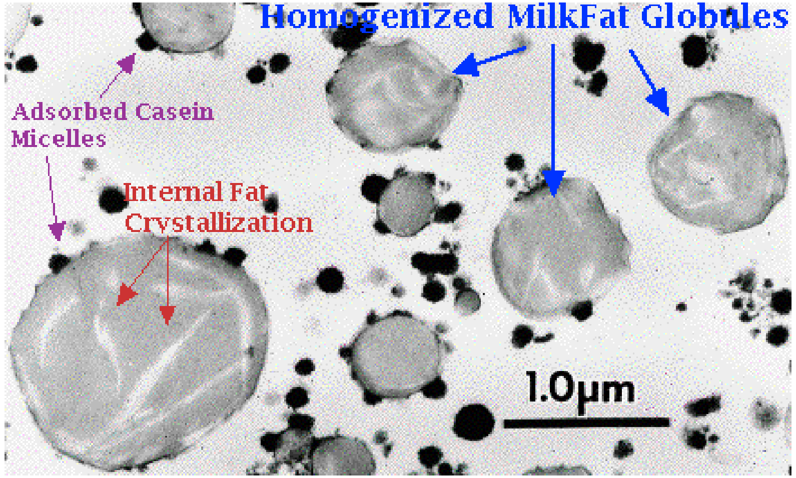
large particle

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

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

  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)

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

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]: http://people.compute.dtu.dk/jerf/code/

References

- Frisvad, J. R., Christensen, N. J., and Jensen, H. W. Computing the scattering properties of participating media using Lorenz-Mie theory. *ACM Transactions on Graphics 26*(3), pp. 60:1–60:10, July 2007.

Case study: milk

skimmed

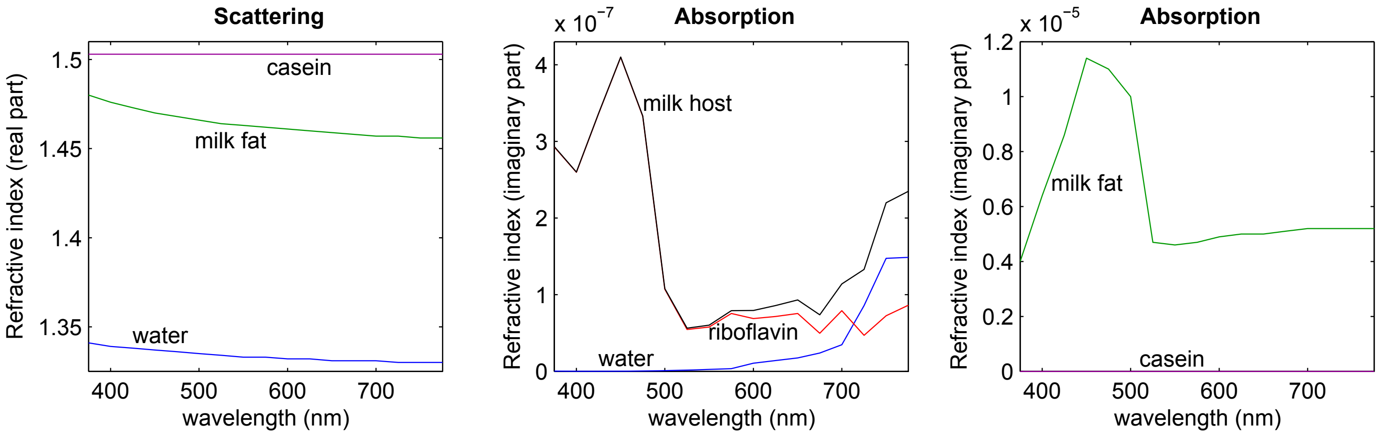
low fat

whole

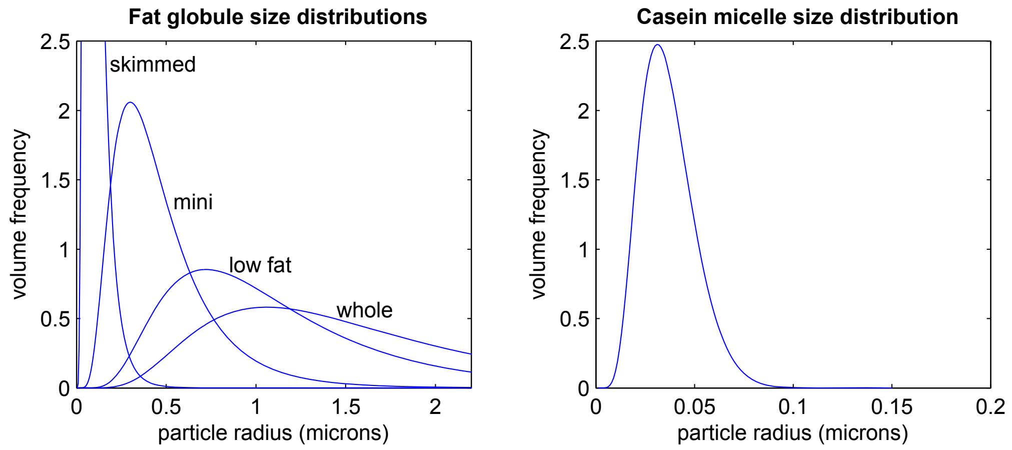
- Refractive index of host: water + dissolved vitamin B2.
- Fat and protein contents: user input in wt.-%.
- Refractive index of milk fat and casein: measured spectra.
- Shape of fat globules and casein micelles: spheres and a volume to surface area ratio.
- Size distributions: log-normal with mean depending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

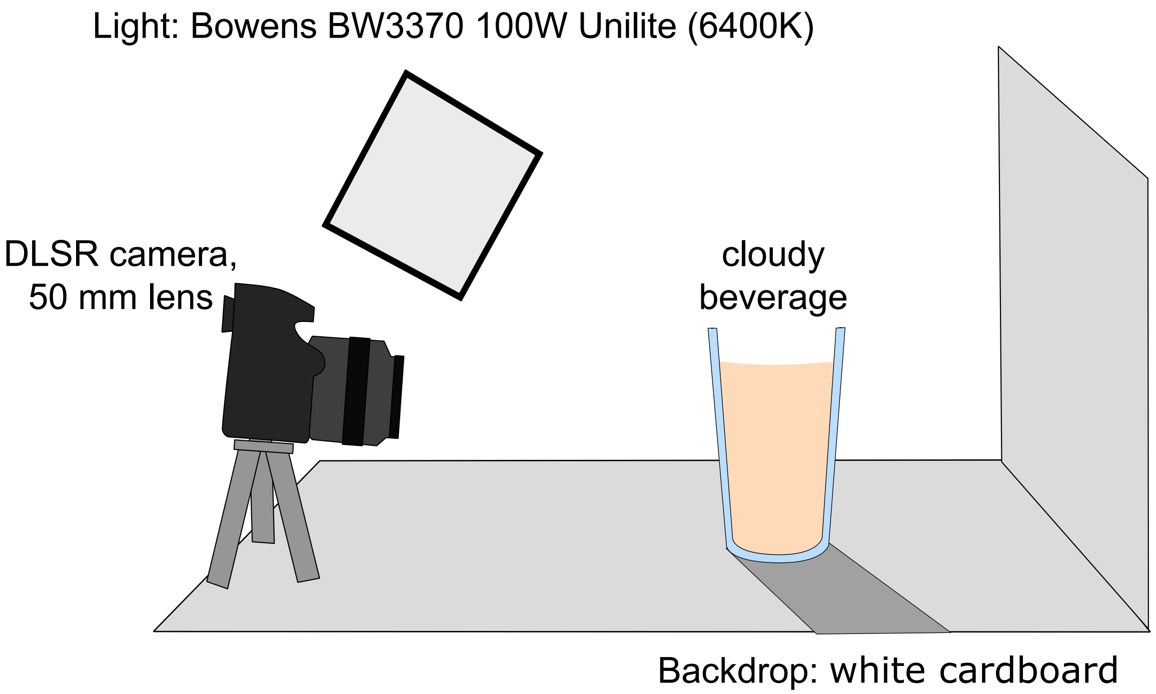
- Vitamin B2 content: 0.17 mg / 100 g
- Protein content: 3 g / 100 g
- Fat content: 0.1 g (skimmed), 1.5 g (low fat), 3.5 g (whole) / 100 g
- Homogenization pressure: 0 MPa (model: [0, 50] MPa)

References

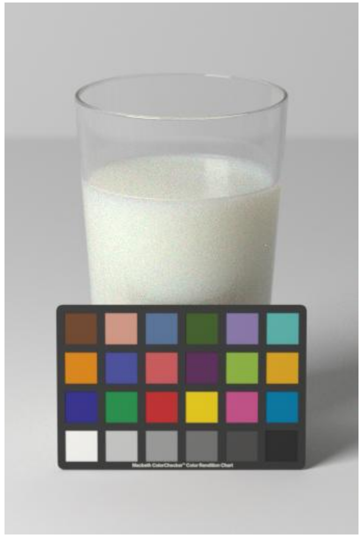
- Frisvad, J. R., Christensen, N. J., and Jensen, H. W. Computing the scattering properties of participating media using Lorenz-Mie theory. *ACM Transactions on Graphics 26*(3), pp. 60:1–60:10, July 2007.

Predicting appearance

Scene



organic low fat milk



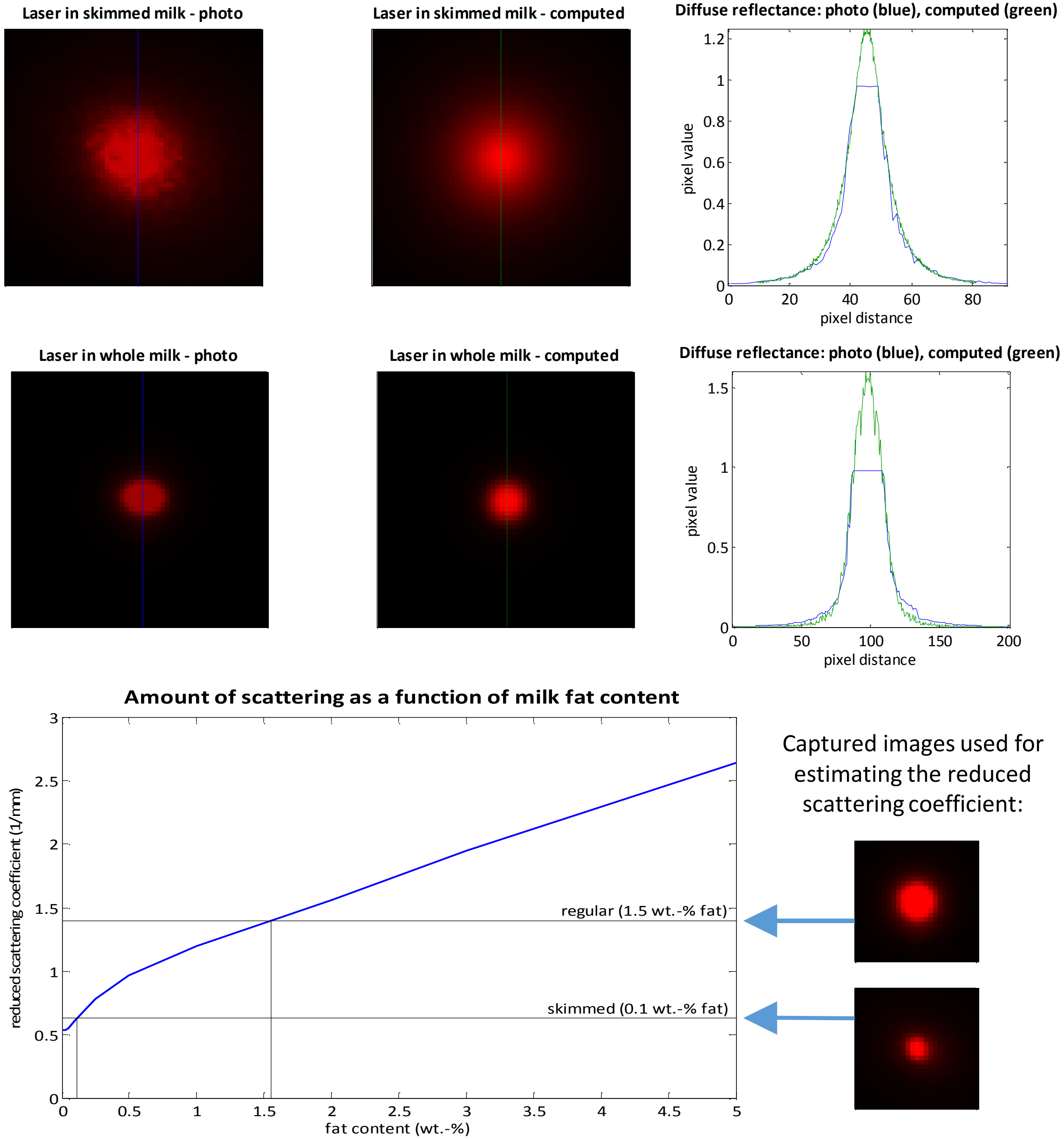
rendering

photograph

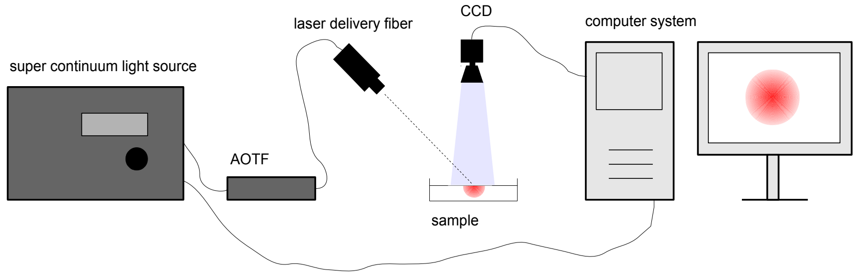
- Digital scene modeled by hand to match physical scene (as best we could)

Simplistic model validation

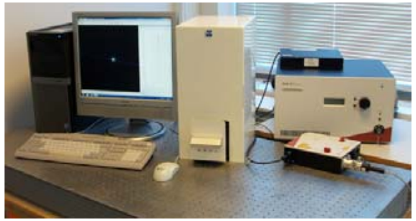
- Camera
- Tripod
- Laser pointer
- Cup (use black cup)



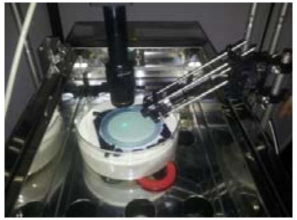
Measuring scattering properties using diffuse reflectance spectroscopy



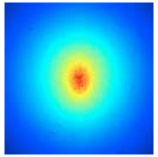
lab setup

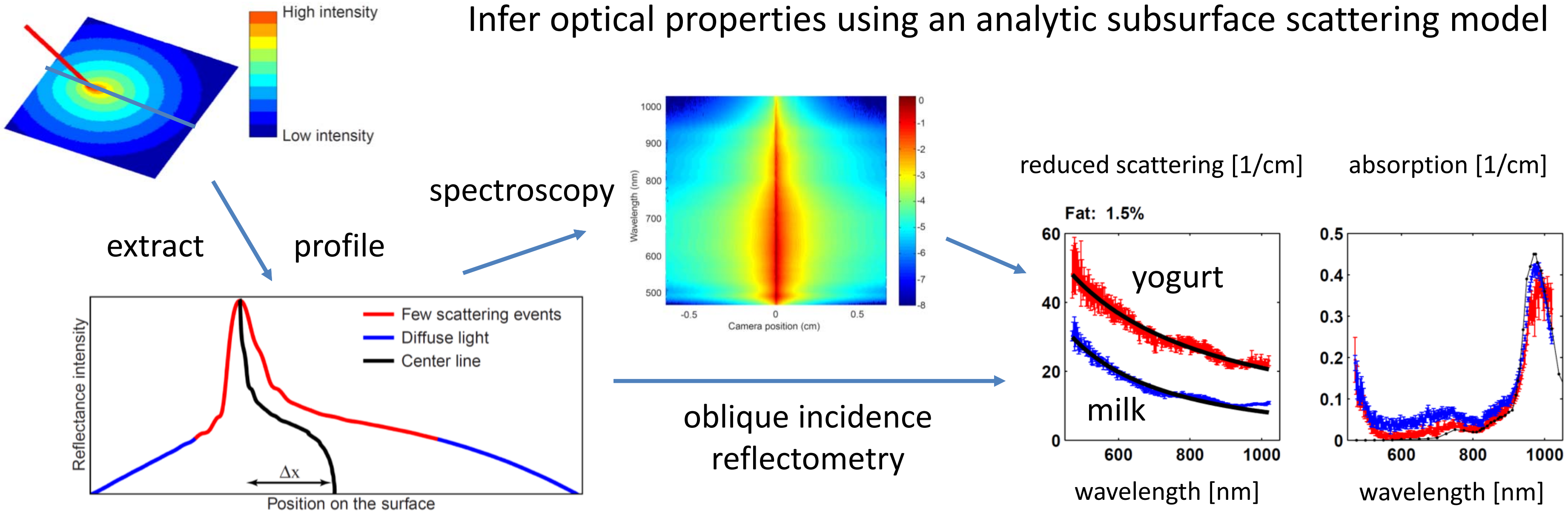


in situ setup



sample image
(log transformed,
false colours)

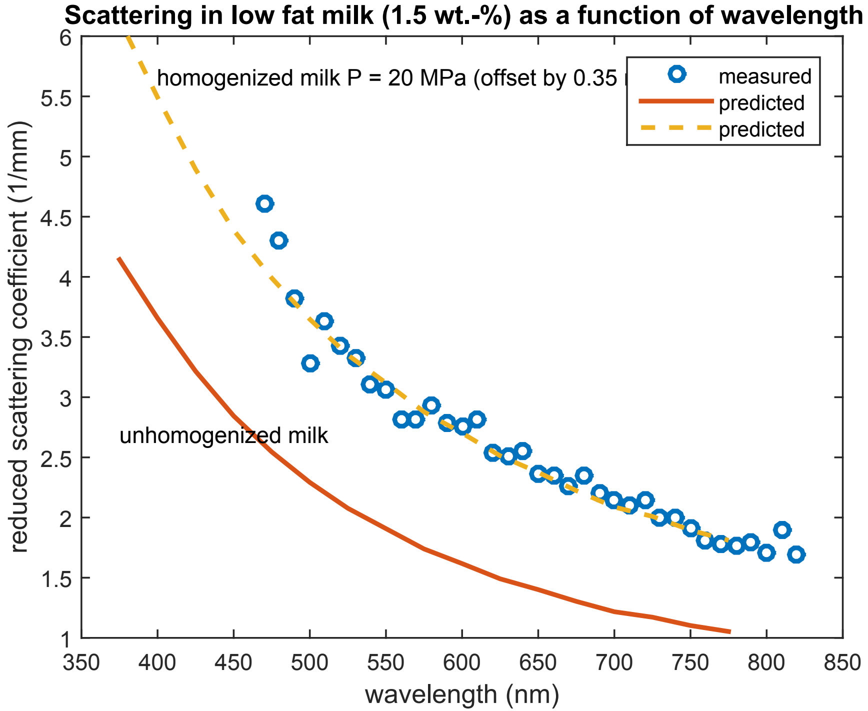


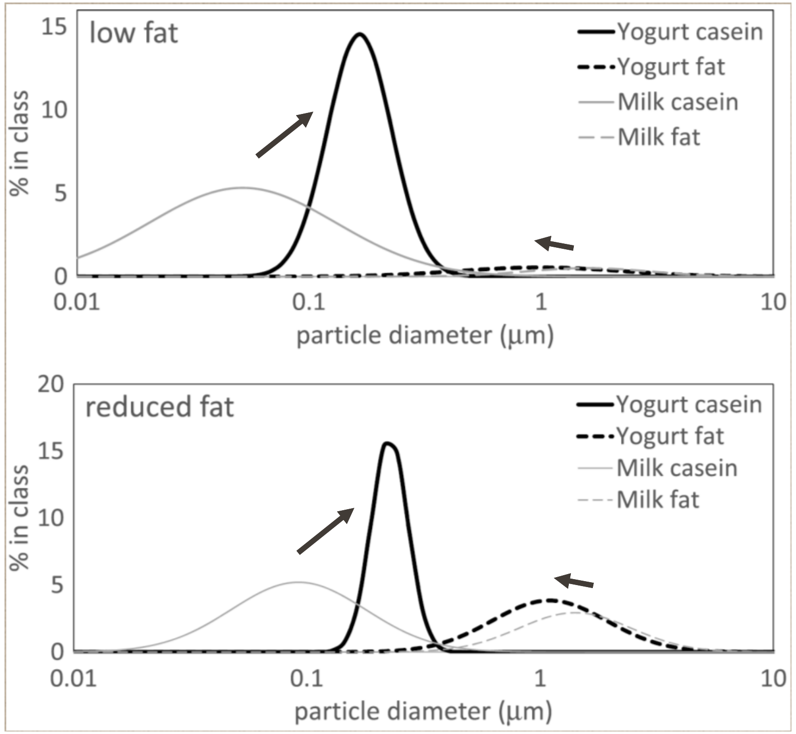


References

- Abildgaard, O. H. A., Kamran, F., Dahl, A. B., Skytte, J. L., Nielsen, F. D., Thomsen, C. L., Andersen, P. E., Larsen, R., and Frisvad, J. R. Non-invasive assessment of dairy products using spatially resolved diffuse reflectance spectroscopy. *Applied Spectroscopy 69*(9), pp. 1096–1105, September 2015.

Using measured scattering properties for product analysis





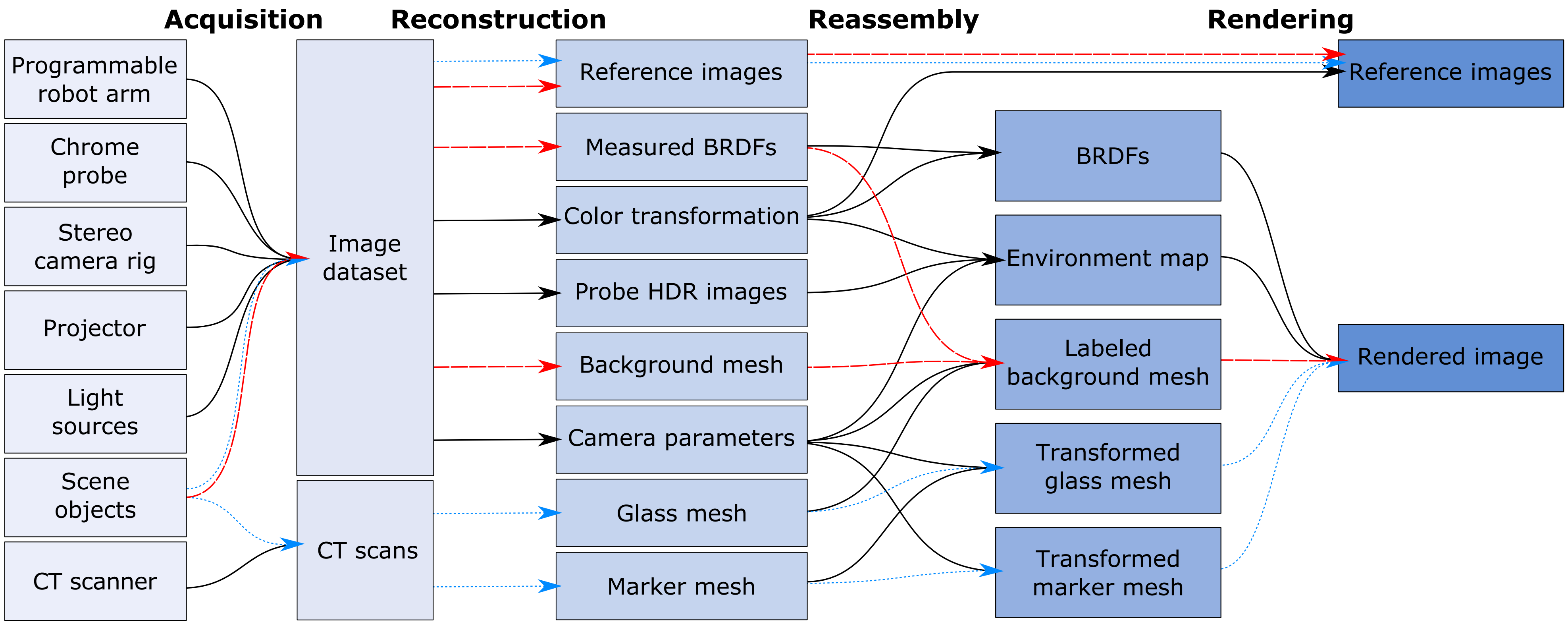
inferring
milk homogenization
(pressure)

inferring
milk fermentation
(apparent particle size distribution)

References

- Abildgaard, O. H. A., Frisvad, J. R., Falster, V., Parker, A., Christensen, N. J., Dahl, A. B., and Larsen, R. Noninvasive particle sizing using camera-based diffuse reflectance spectroscopy. *Applied Optics 55*(14), pp. 3840–3846, May 2016.

Multimodal digitization pipeline



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

References

- Stets, J. D., Dal Corso, A., Nielsen, J. B., Lyngby, R. A., Jensen, S. H. N., Wilm, J., Doest, M. B., Gundlach, C., Eiriksson, E. R., Conradsen, K., Dahl, A. B., Brentzen, J. A., Frisvad, J. R., and Aans, H. Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering. *Applied Optics 56*(27), pp. 7679–7690, September 2017.

Geometry & Appearance Digitization

Acquisition → Reconstruction → Reassembly → Rendering

Acknowledgement
Thanks to Jonthan Dyssel Stets for letting me use his "Geometry & Appearance Digitization" slides.

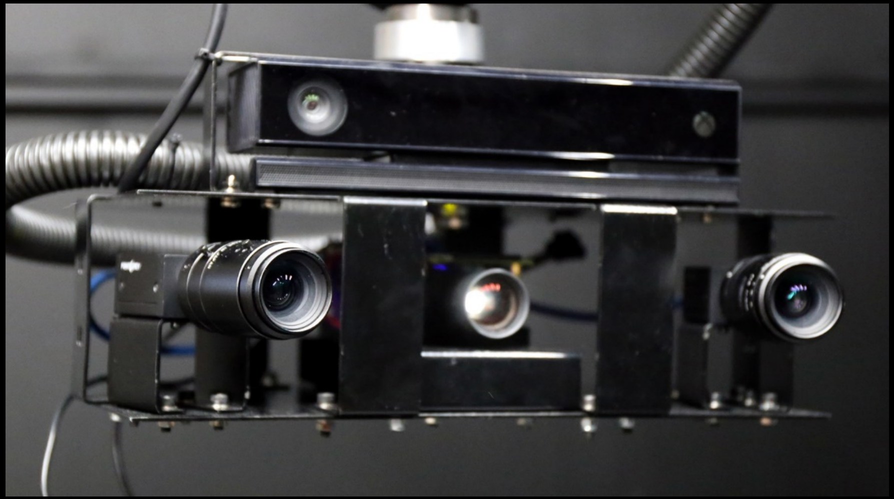
Geometry & Appearance Digitization

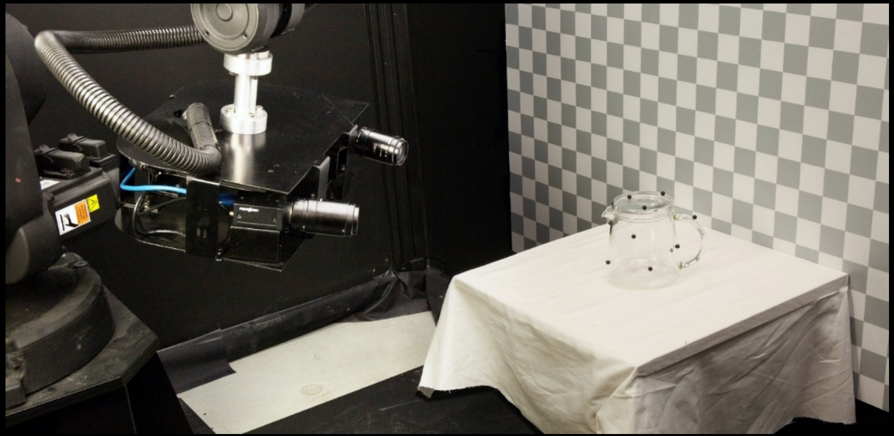
Sphere

Bowl

Teapot

Geometry & Appearance Digitization





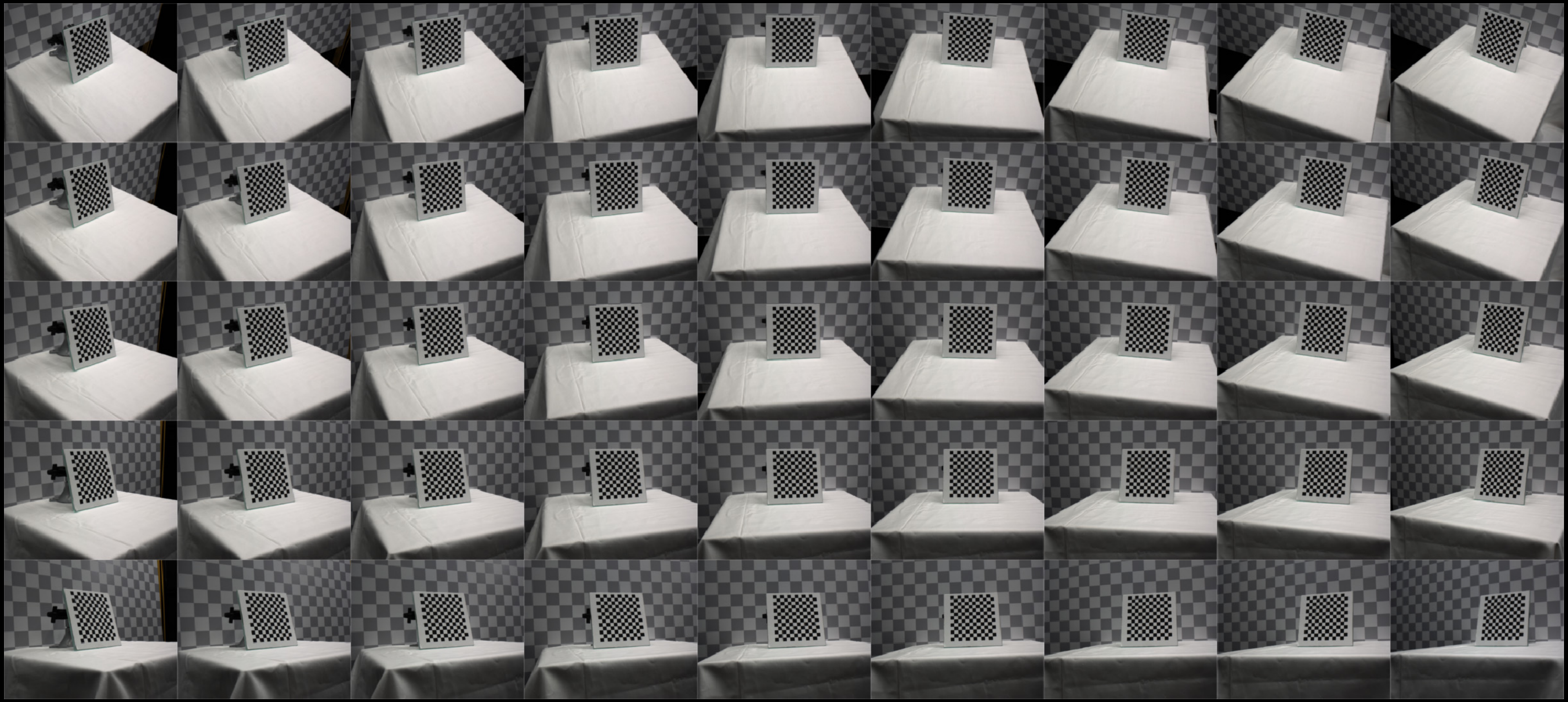
Instruments

- Robot repeatability
- Illumination control
- Stereo camera setup
- Structured light scanner

Scene

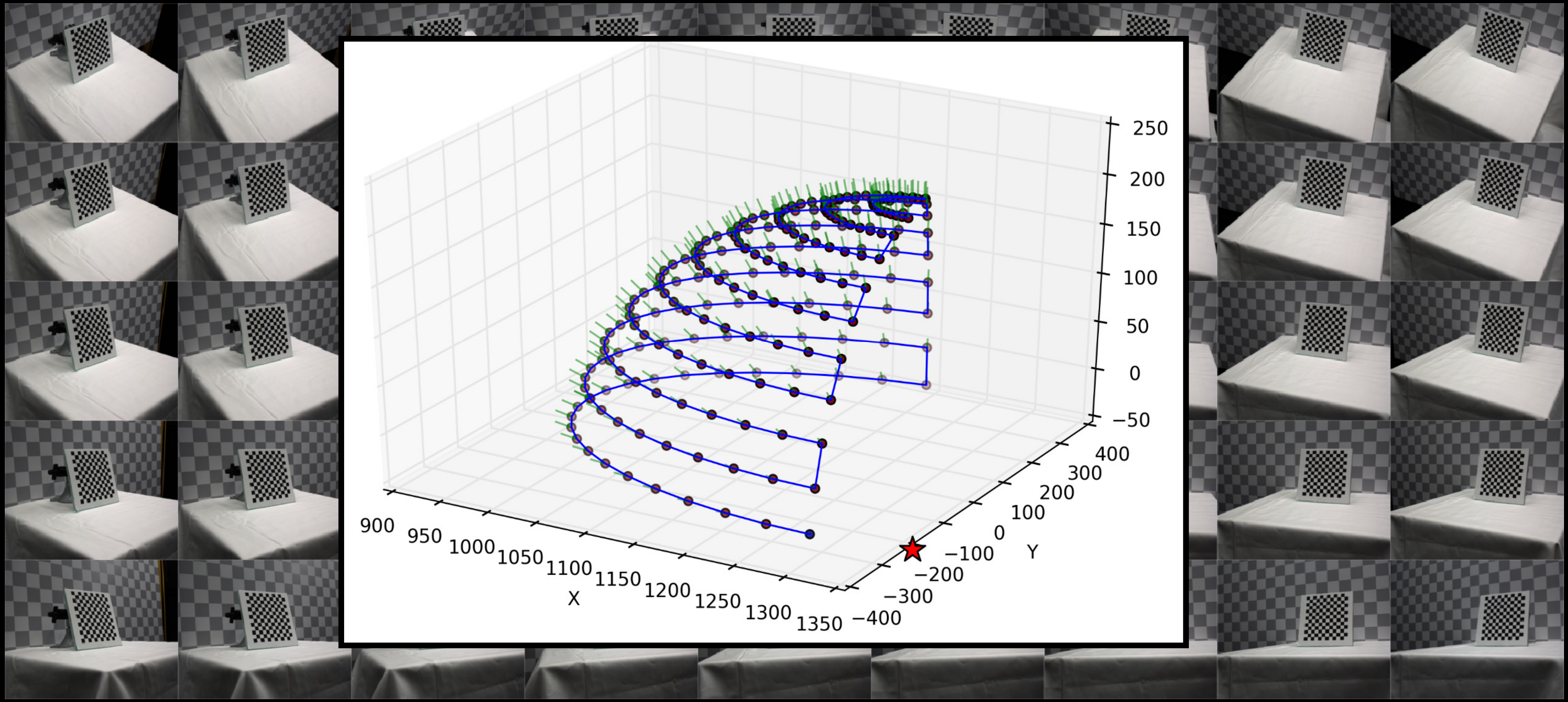
- Checkered backdrop
- Table with cloth
- Glass object

Geometry & Appearance Digitization

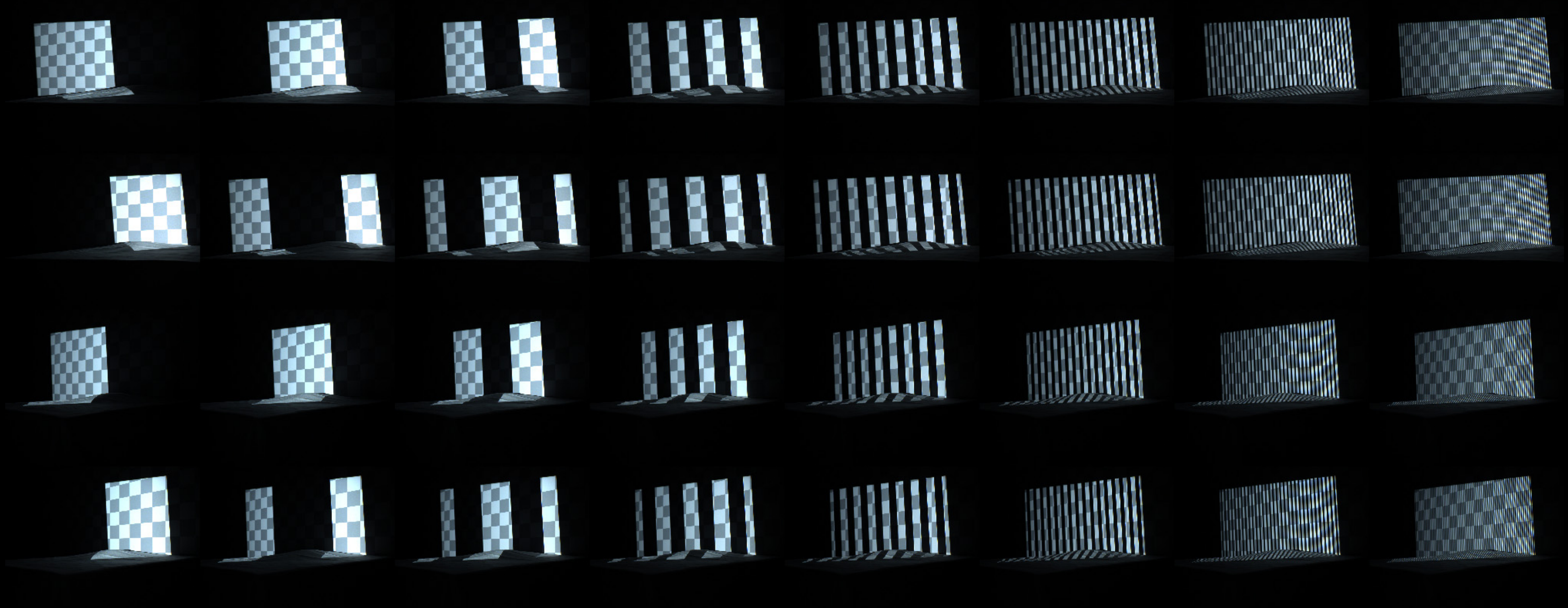


Geometry & Appearance Digitization

Geometry & Appearance Digitization

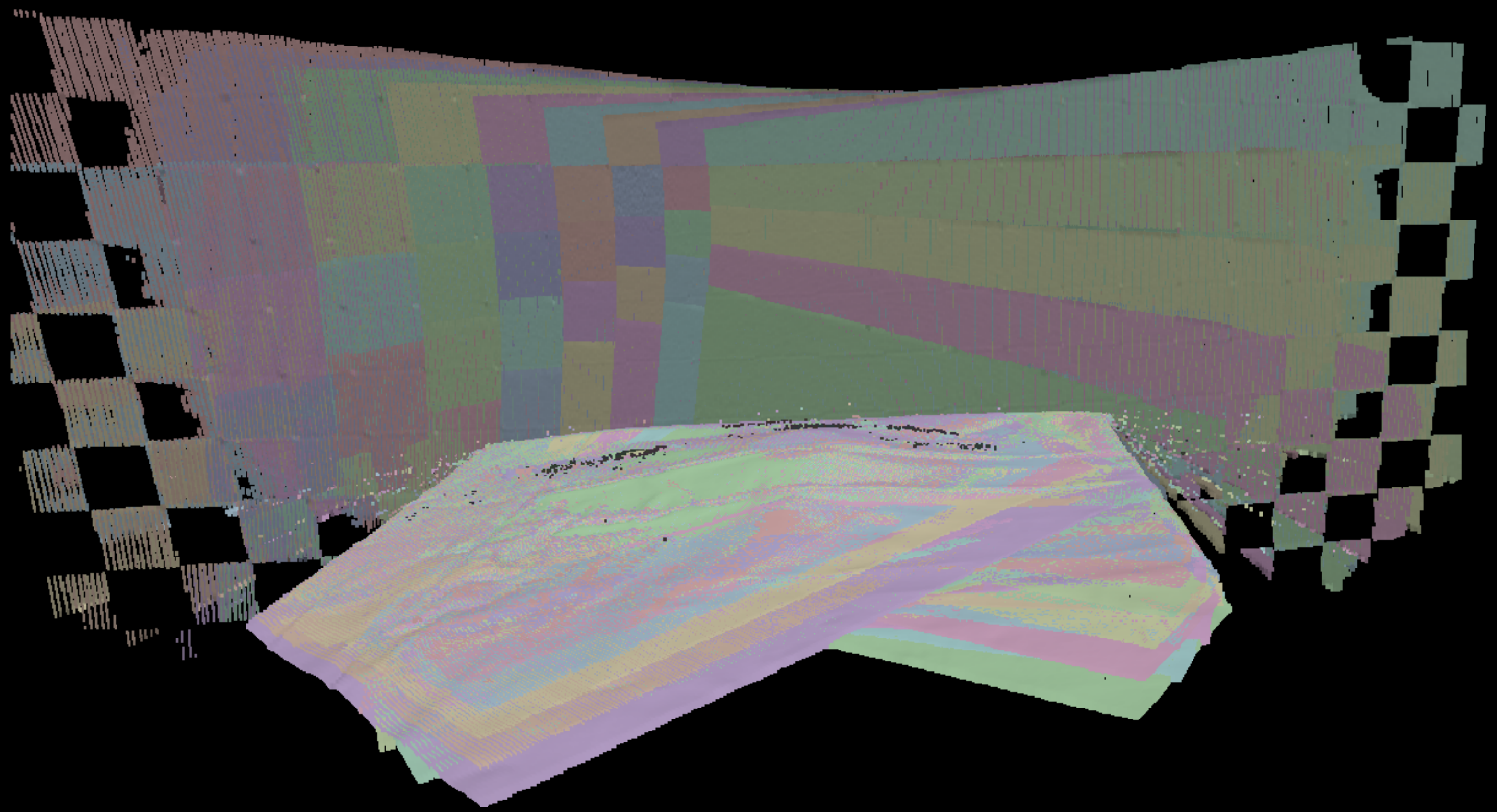


Geometry & Appearance Digitization

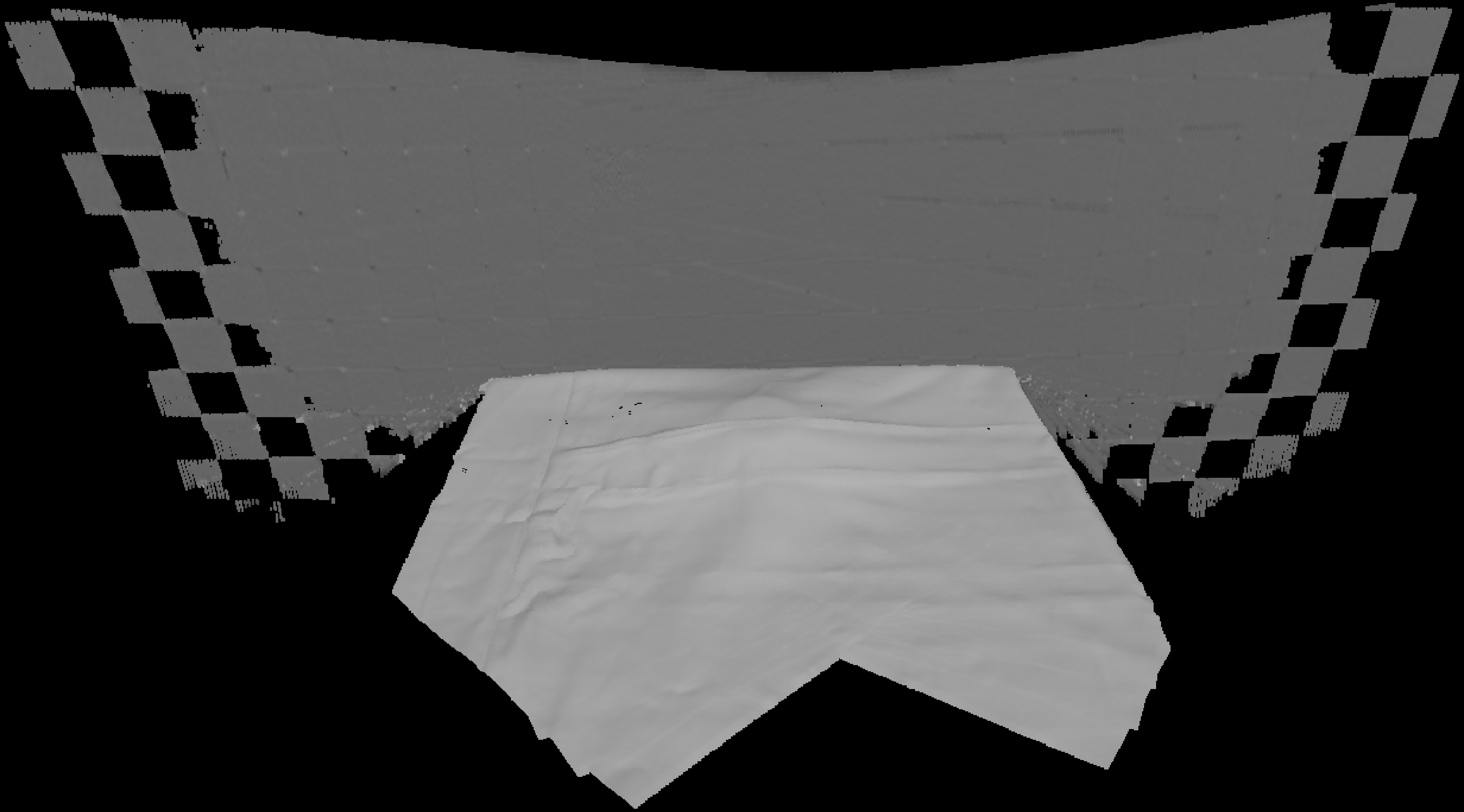


Standard Gray code
Posdamer et al [1982]
Geng [2011]

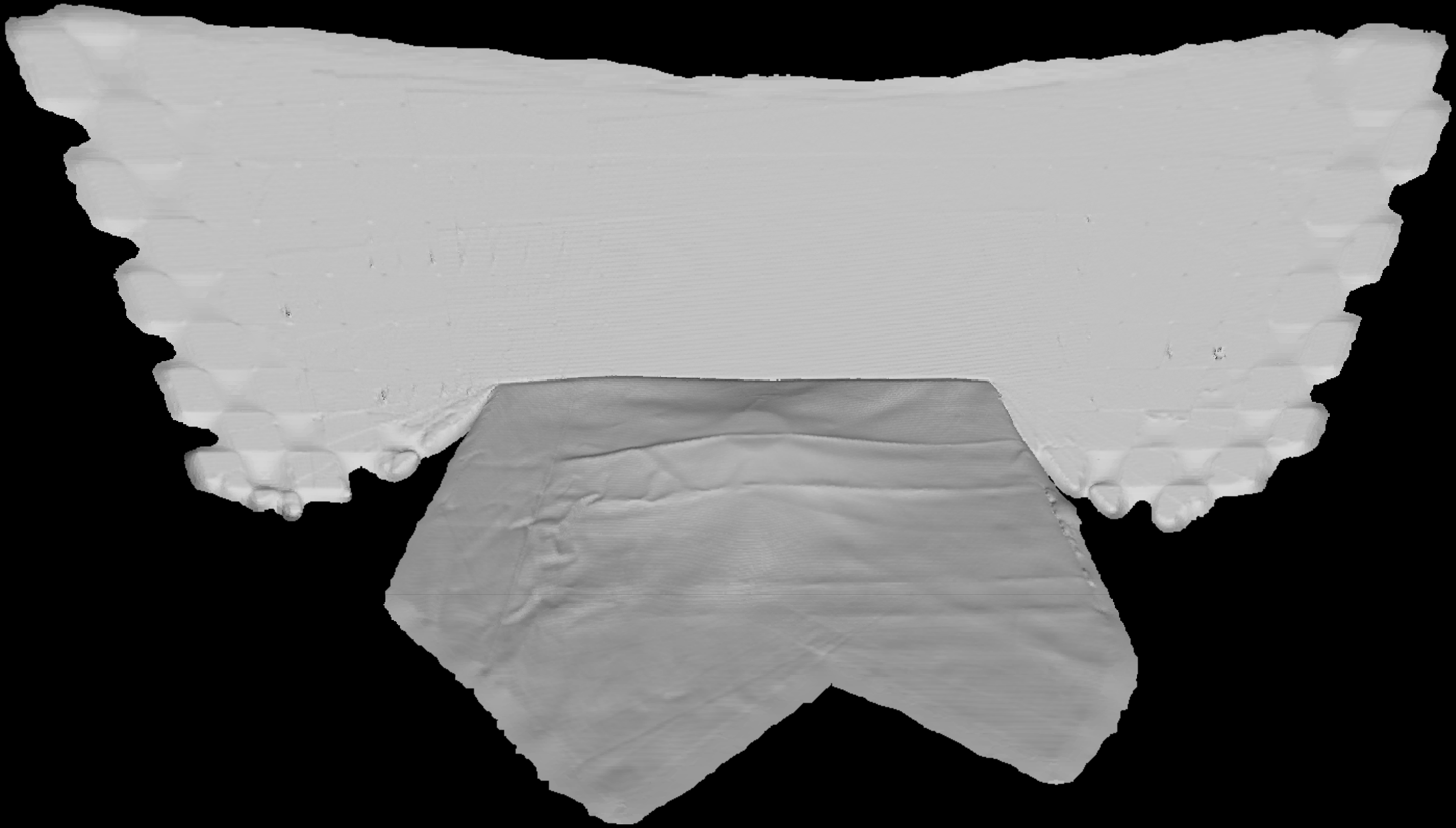
Geometry & Appearance Digitization



Geometry & Appearance Digitization

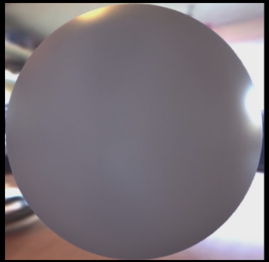


Geometry & Appearance Digitization



Micropolygon labeling for assigning BRDFs

White checker

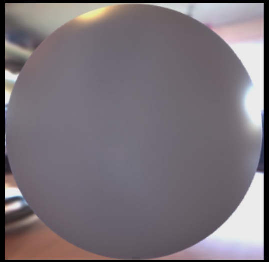


Grey checker

White cloth

Micropolygon labeling for assigning BRDFs

White checker

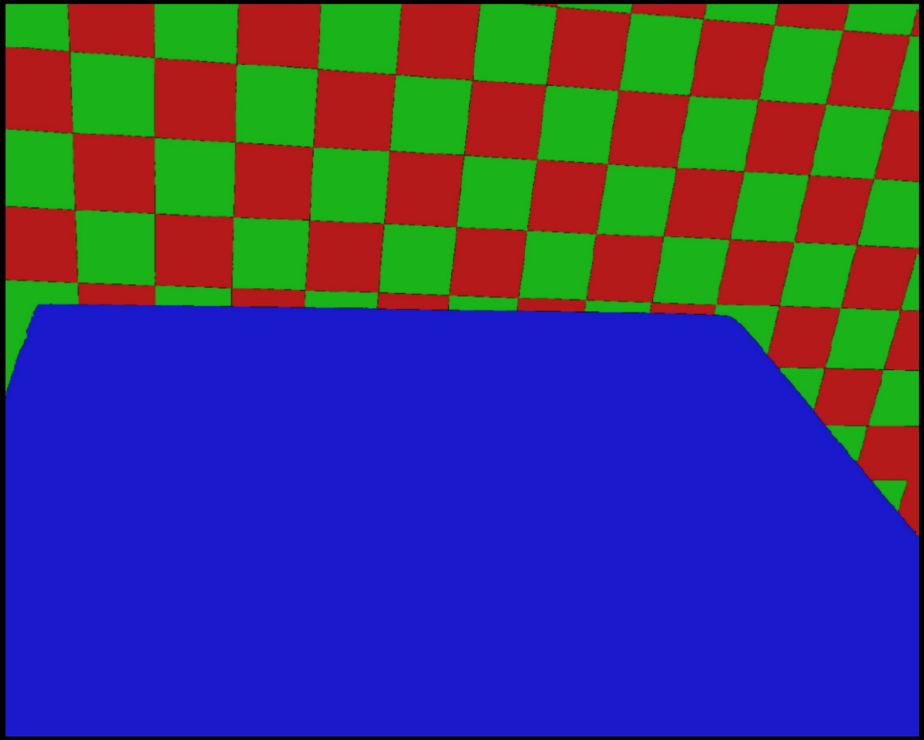


Grey checker

White cloth

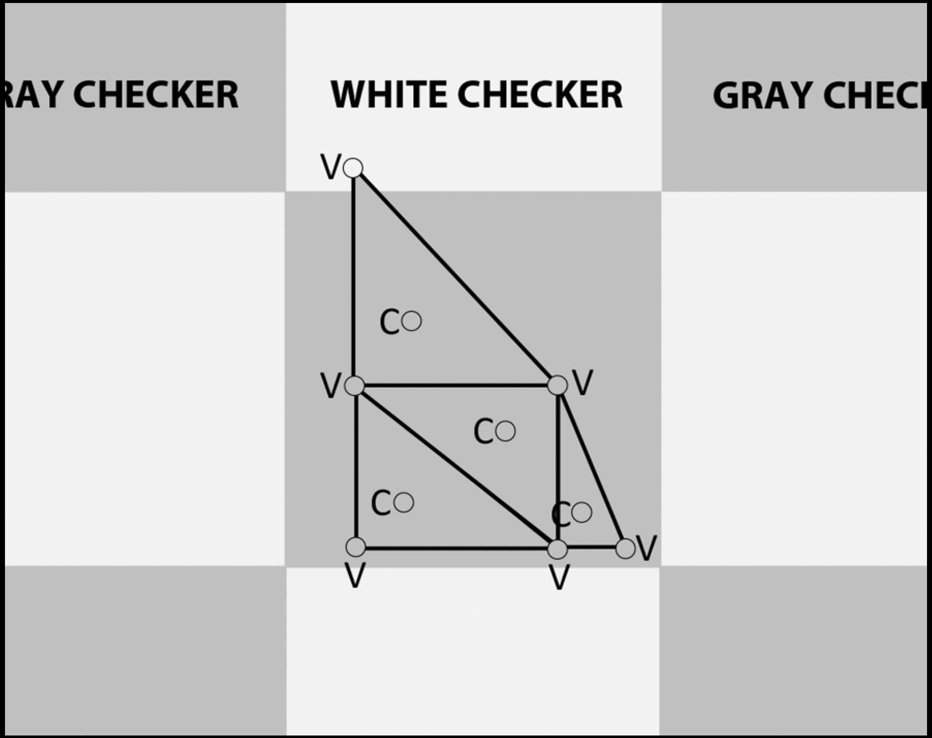
Micropolygon labeling for assigning BRDFs

Micropolygon labeling for assigning BRDFs

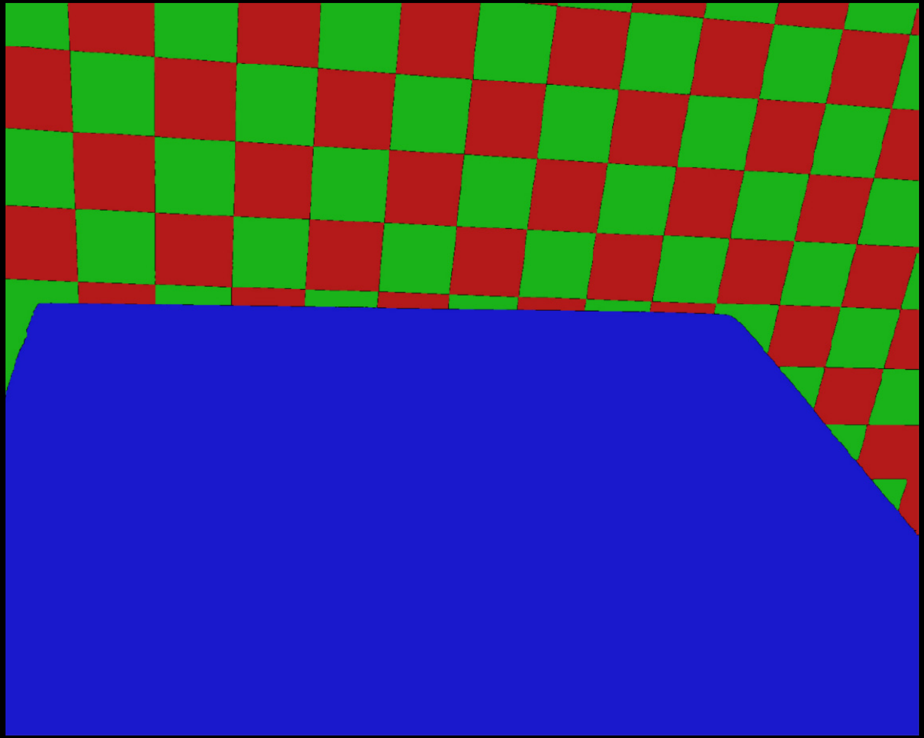


Edge Detection + Watershed

Micropolygon labeling for assigning BRDFs

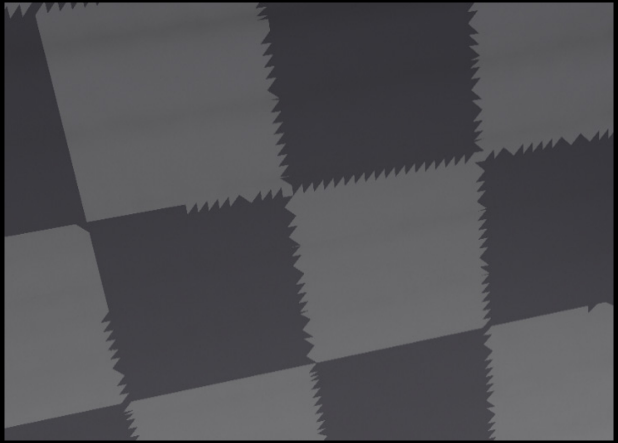


Backprojection using NN Search

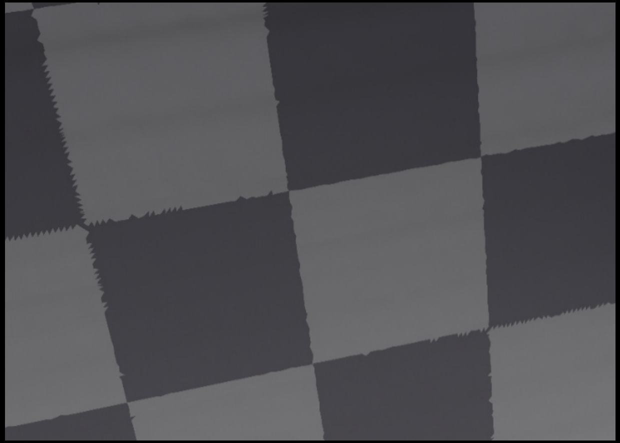


Edge Detection + Watershed

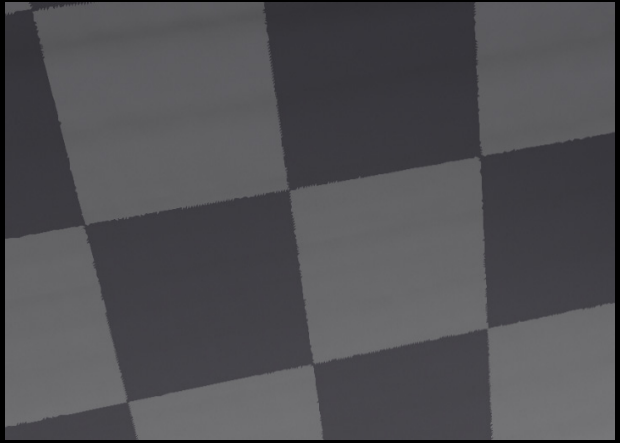
Micropolygon labeling for assigning BRDFs



Original

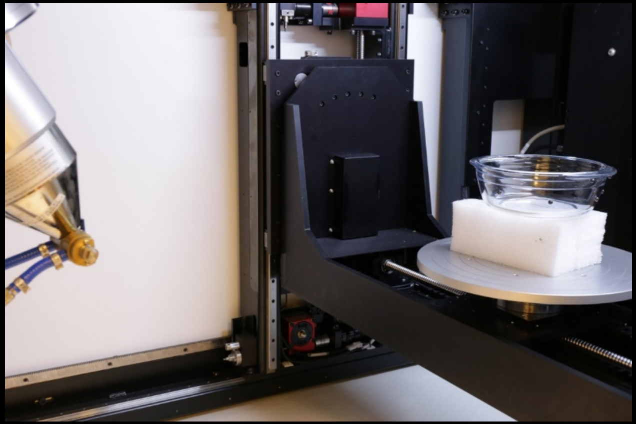


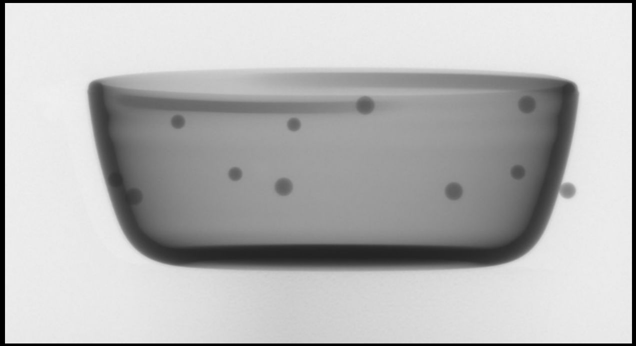
One subdivision



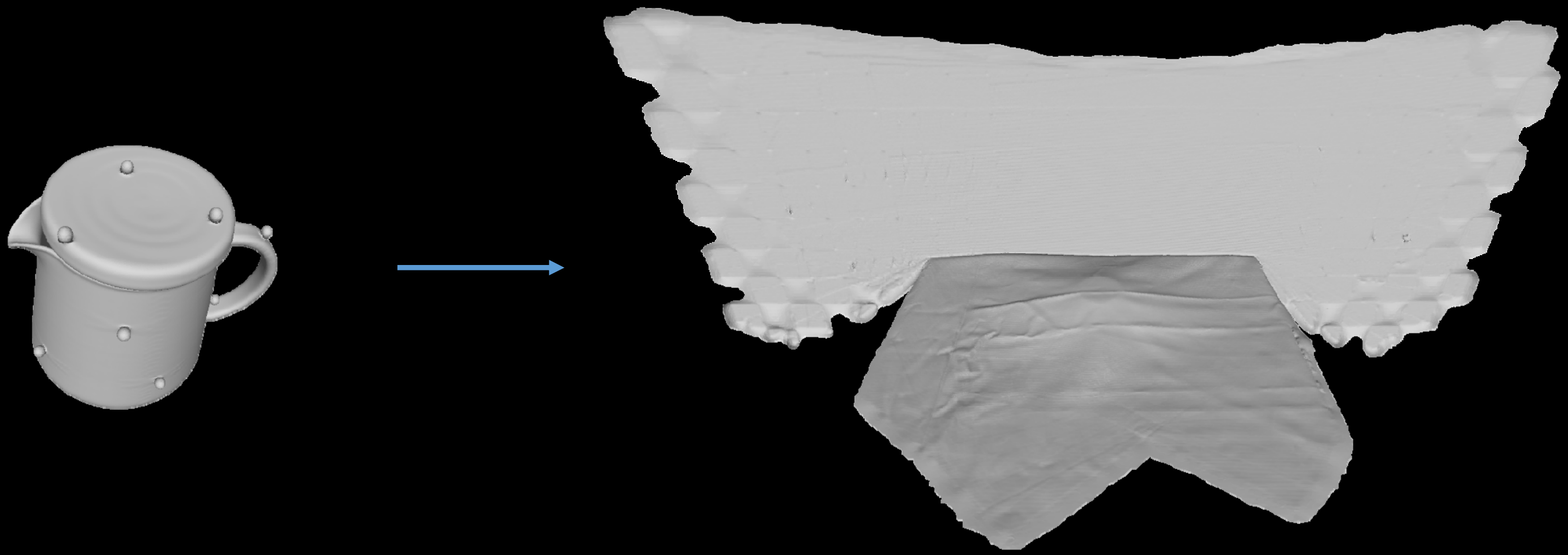
Two subdivisions

A cross modality marker-based placement approach

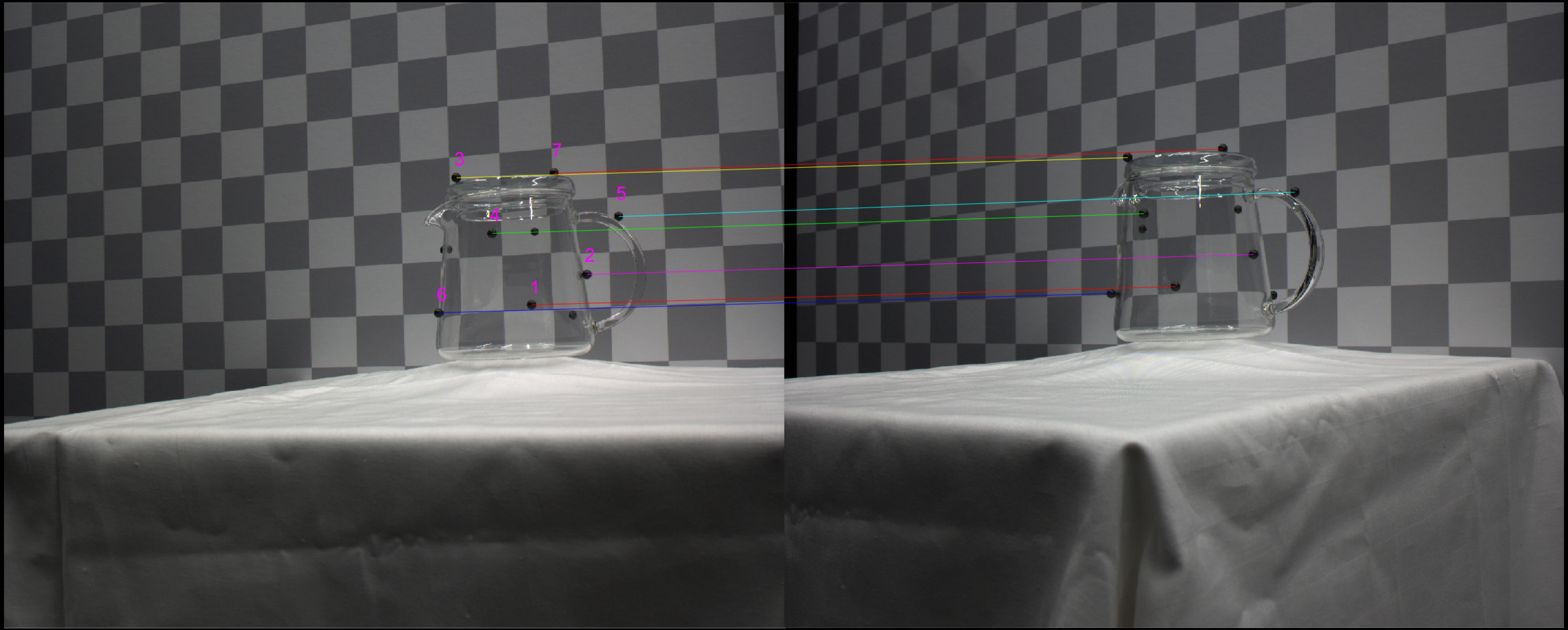




A cross modality marker-based placement approach



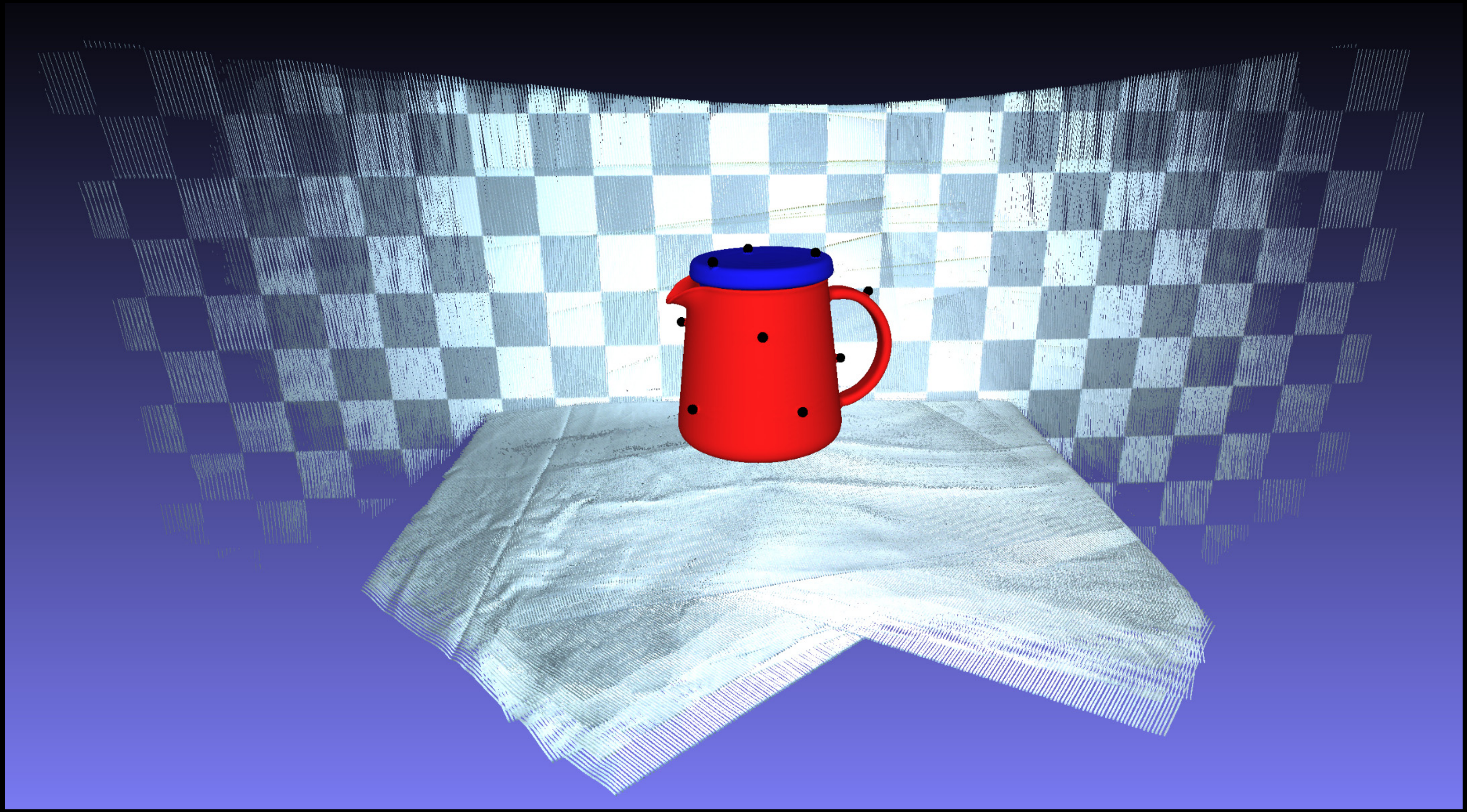
A cross modality marker-based placement approach



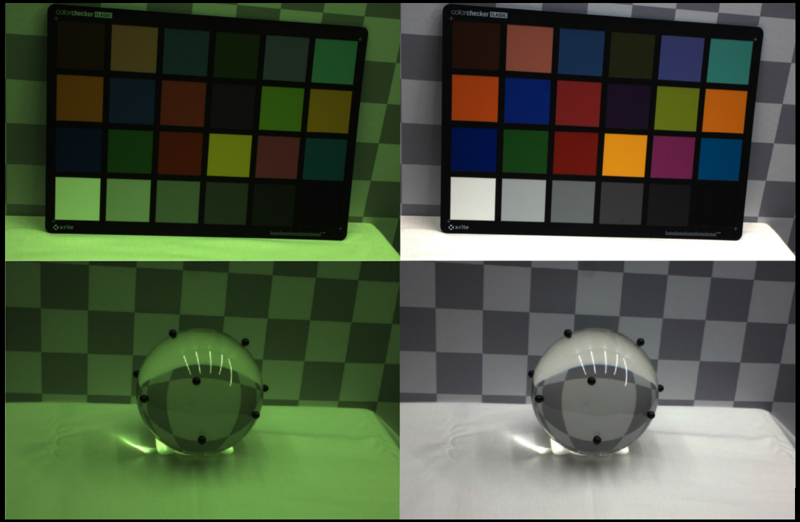
Size Invariant Circular Hough Transform for marker detection
[Atherton et al 1999]

Search for matches on Epipolar Lines

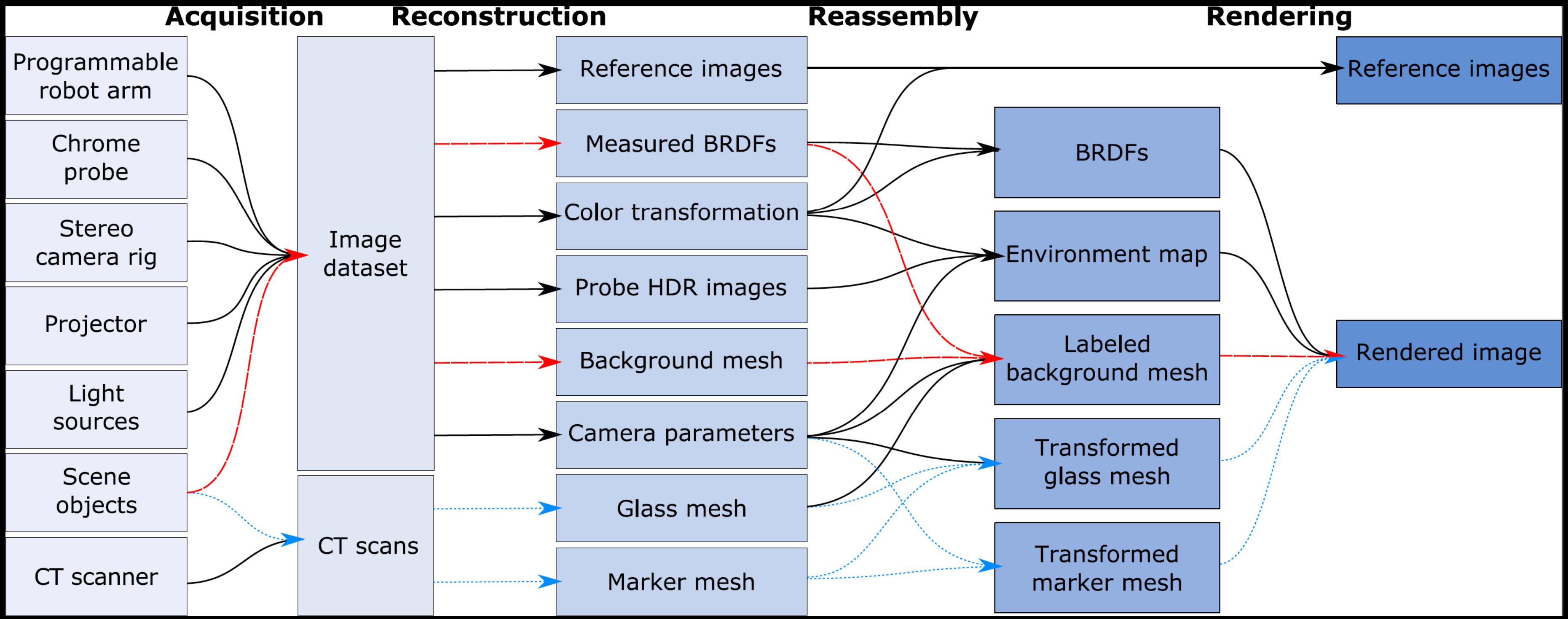
A cross modality marker-based placement approach



Geometry & Appearance Digitization

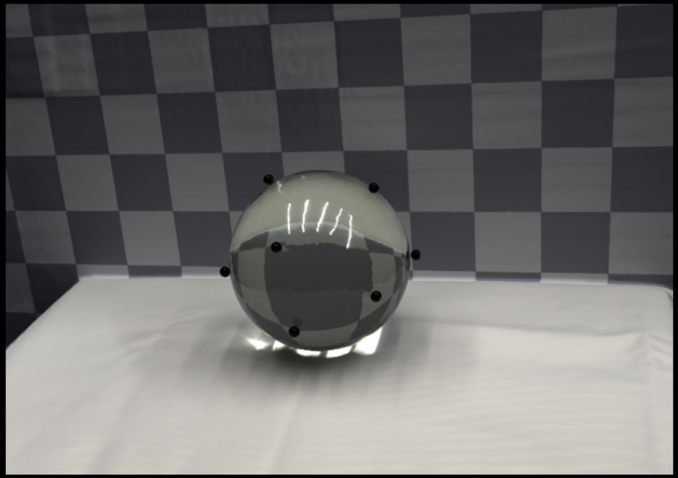


Geometry & Appearance Digitization

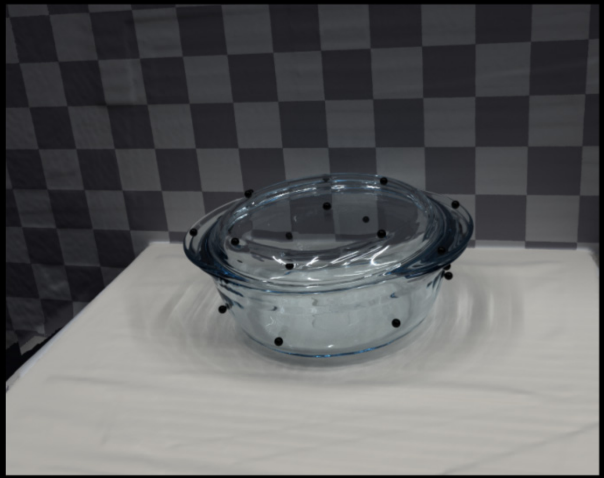


Geometry & Appearance Digitization

Rendering

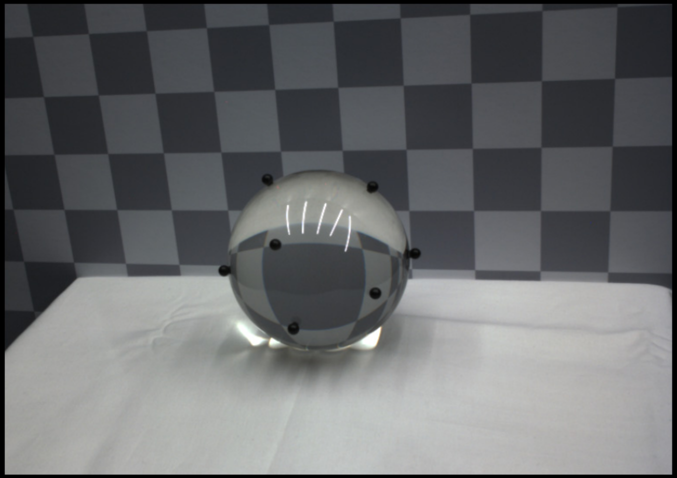
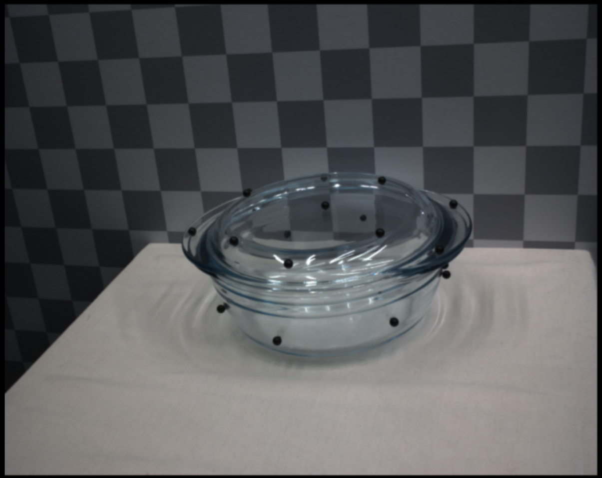


Rendering

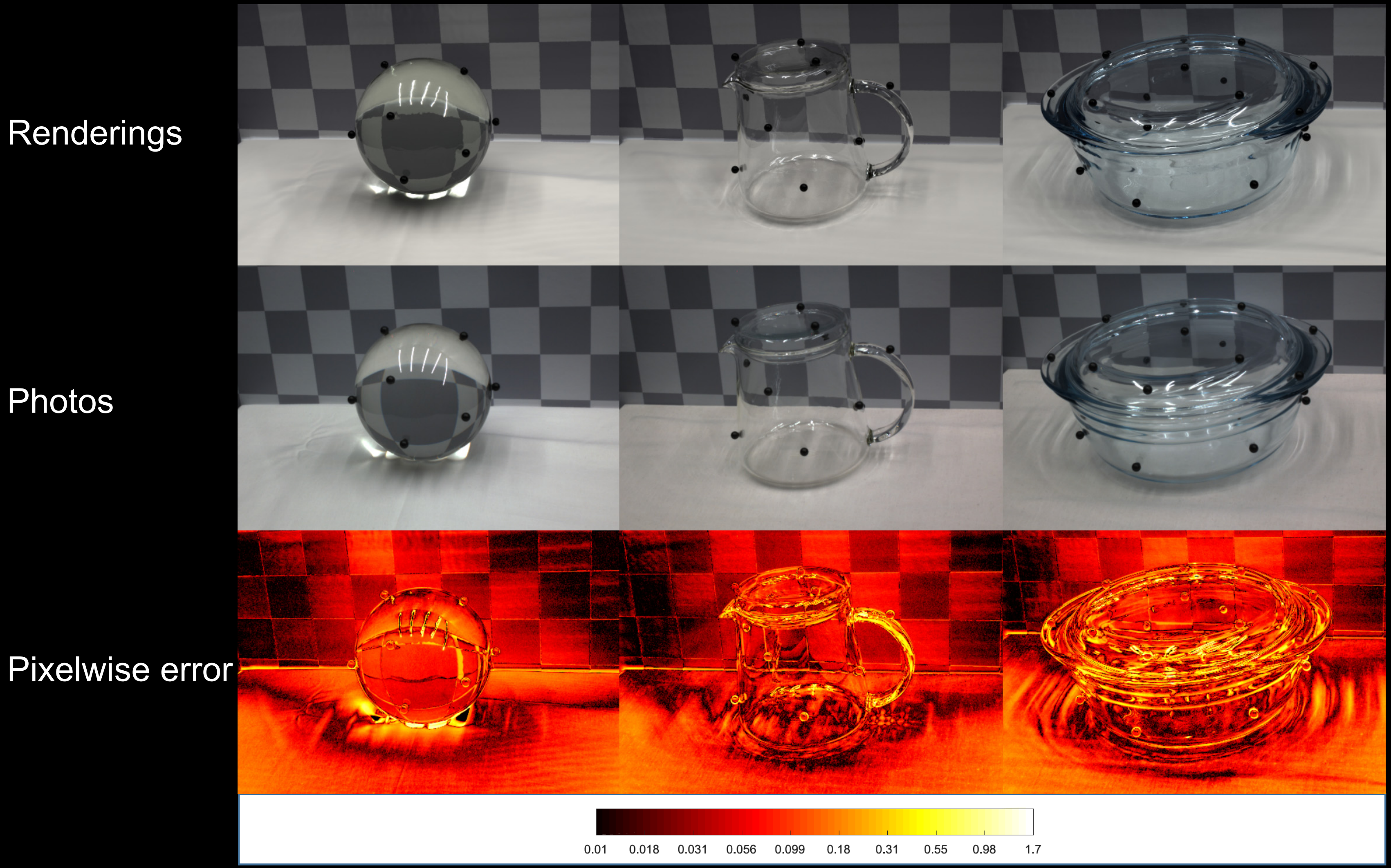


Rendering

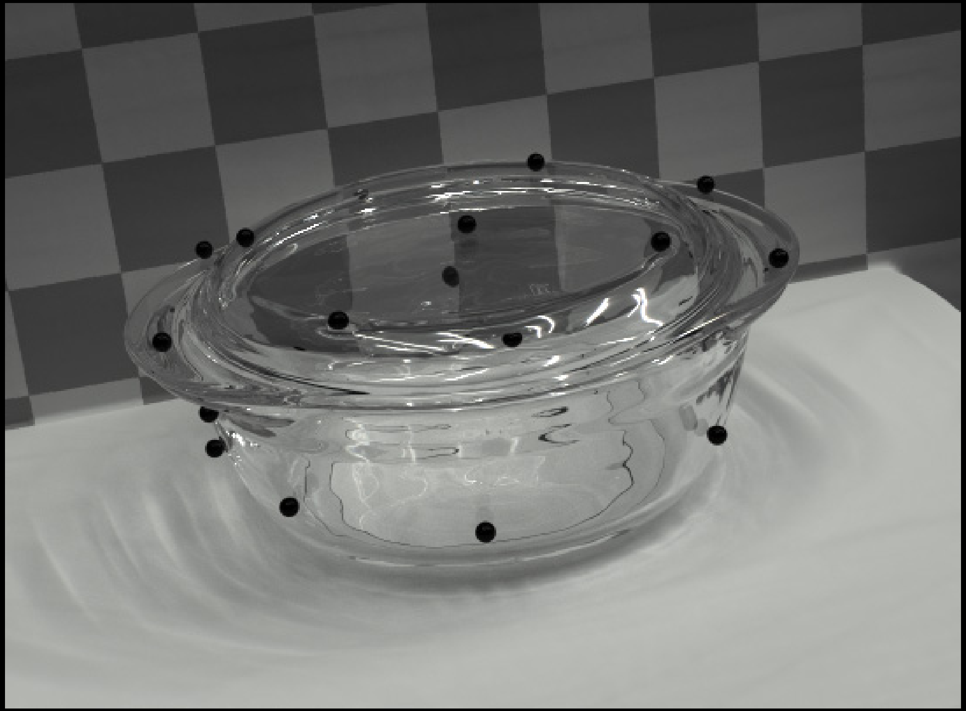
Geometry & Appearance Digitization

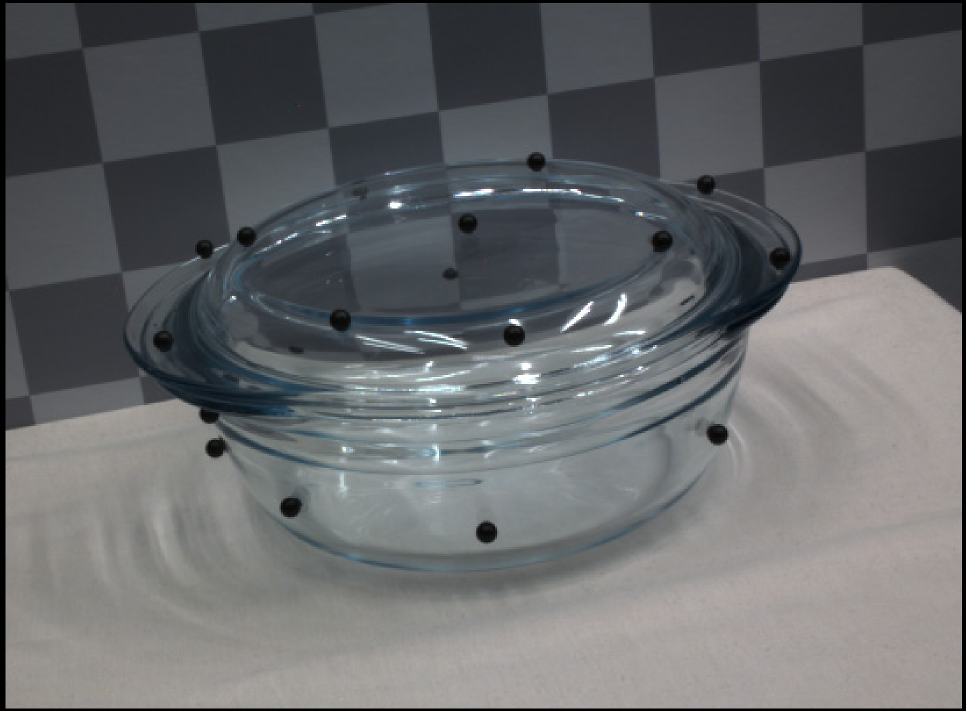
Geometry & Appearance Digitization



Analysis by Synthesis



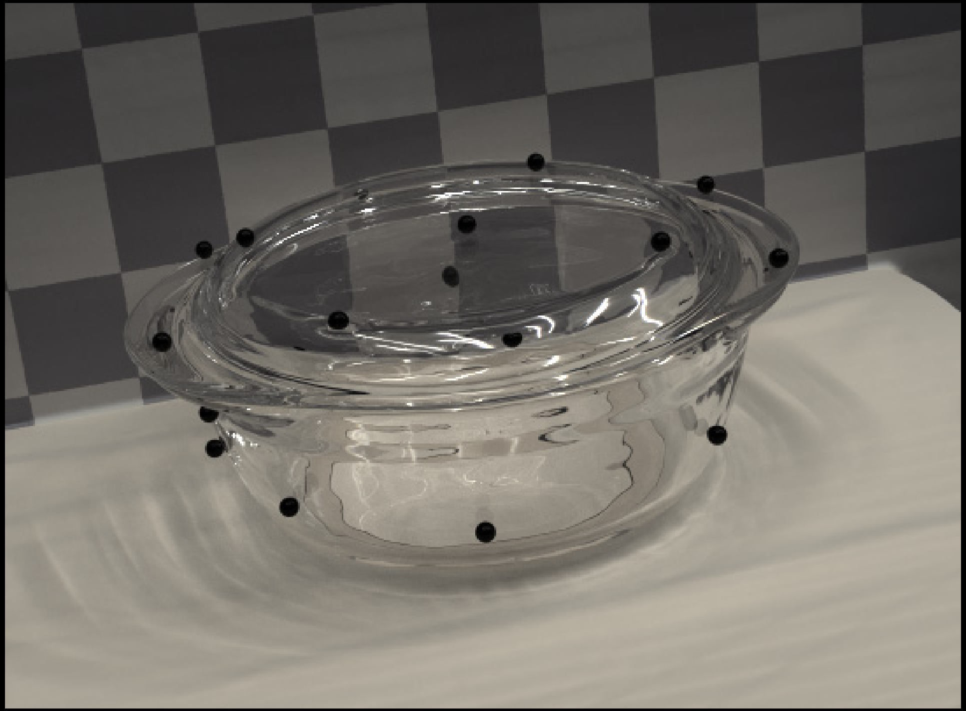
Rendered Image



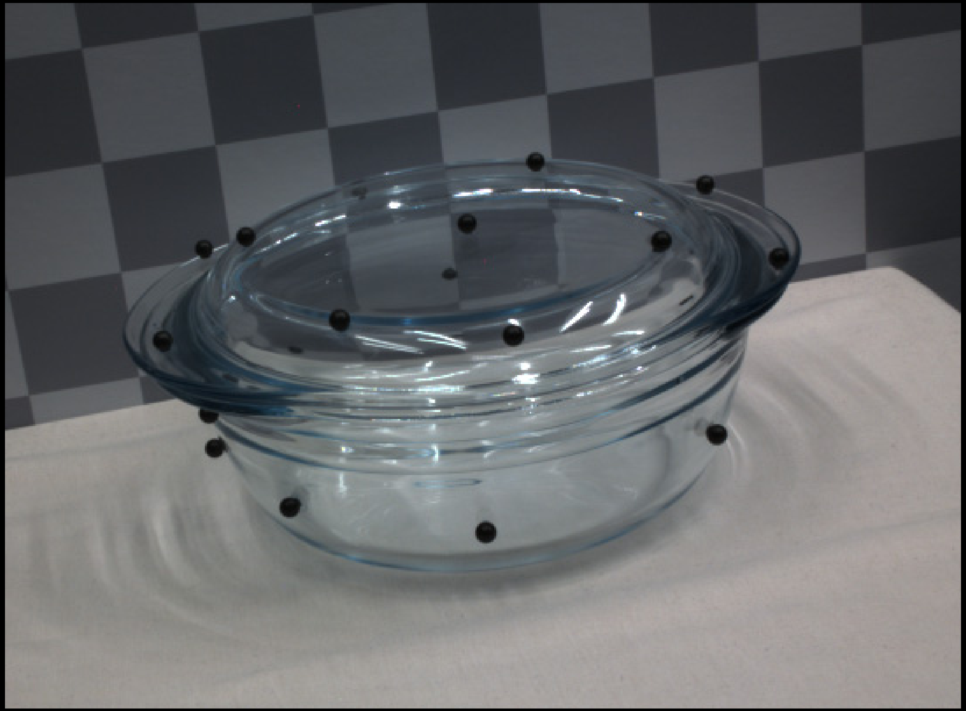
Reference Image

Color Corrected

Analysis by Synthesis



Rendered Image

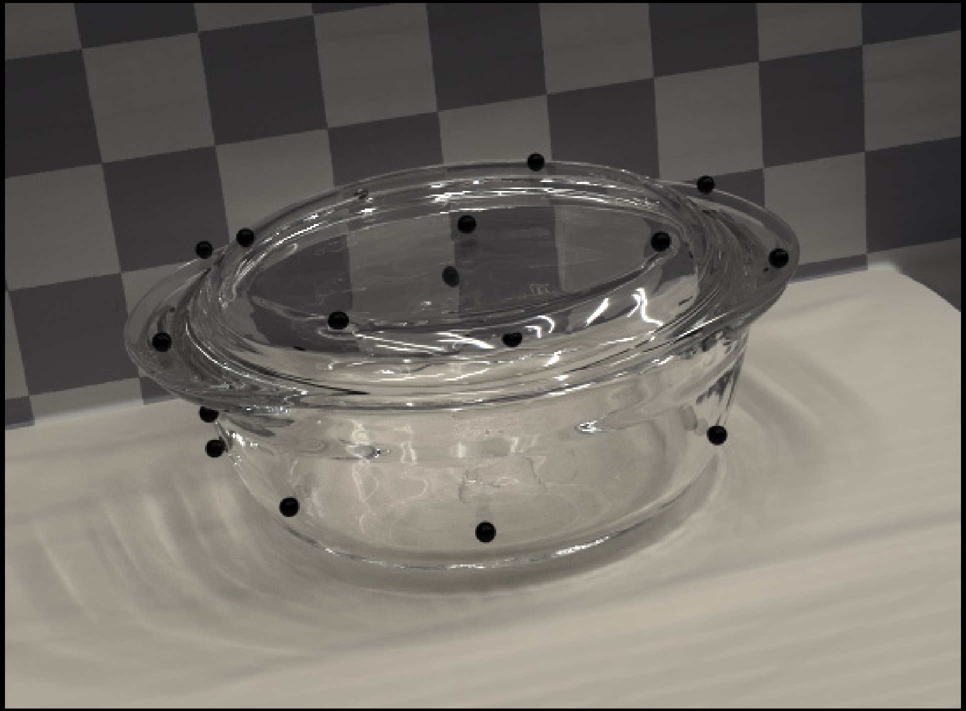


Reference Image

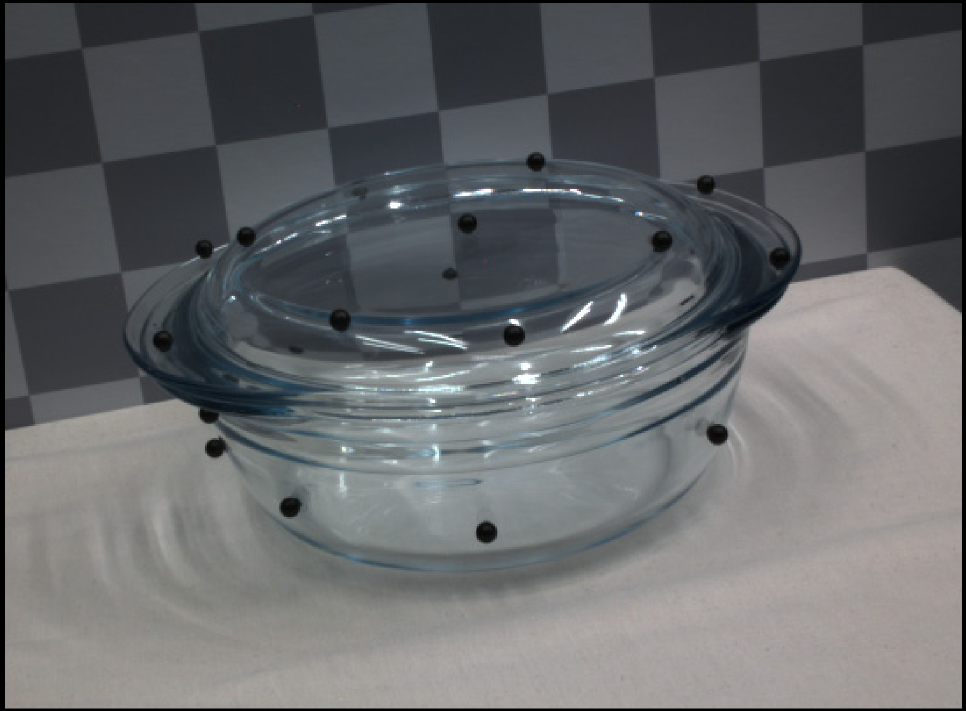
Color Corrected + BRDF

Geometry & Appearance Digitization

Analysis by Synthesis



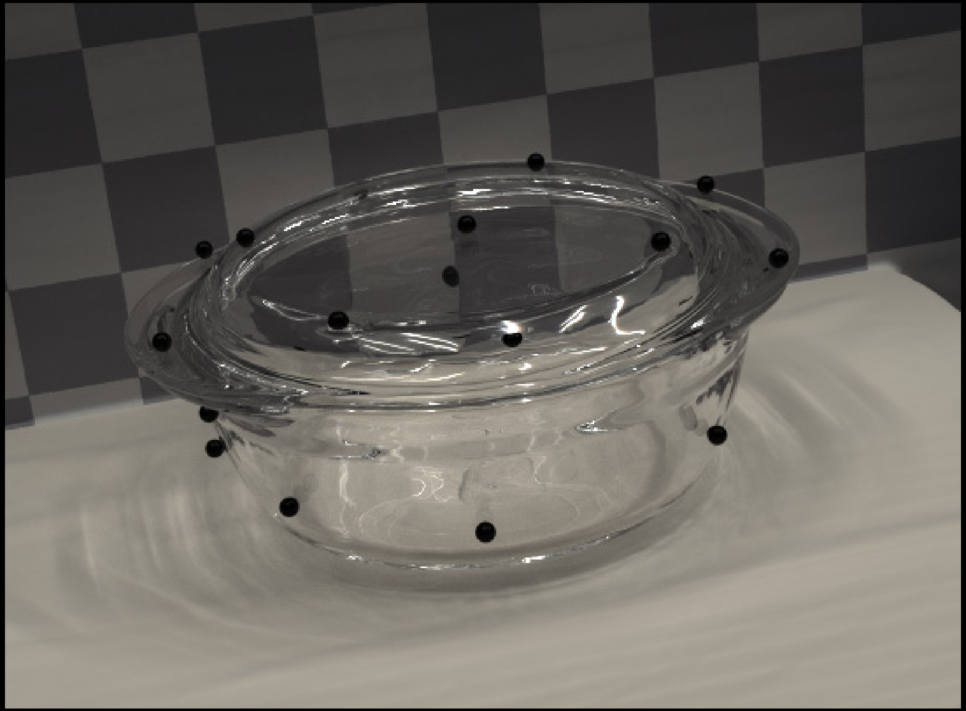
Rendered Image



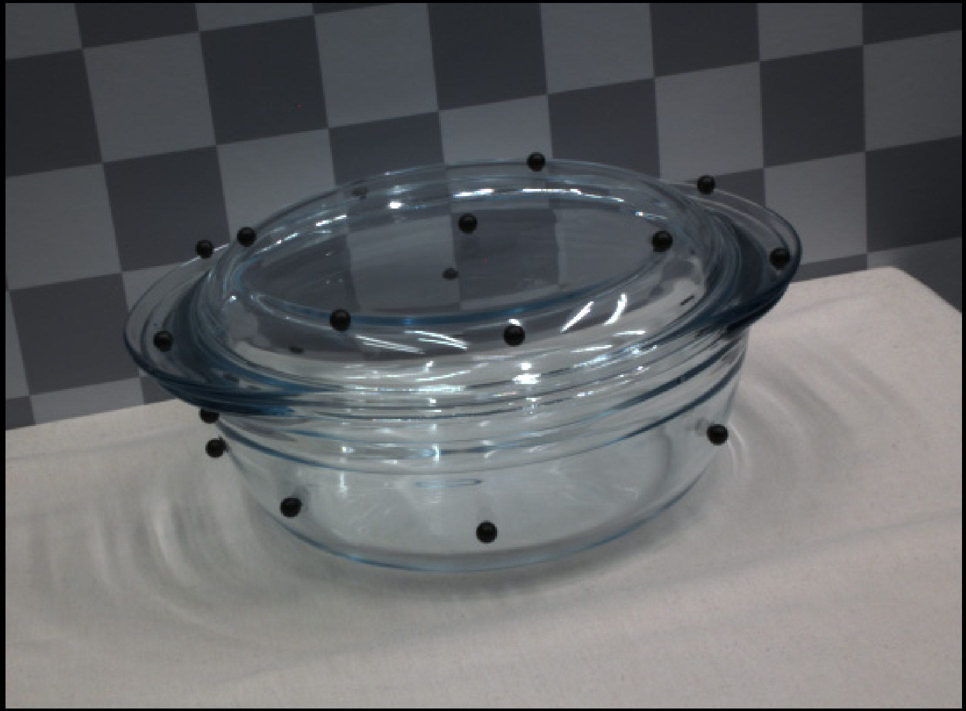
Reference Image

Color Corrected + BRDF + Deformation

Analysis by Synthesis



Rendered Image

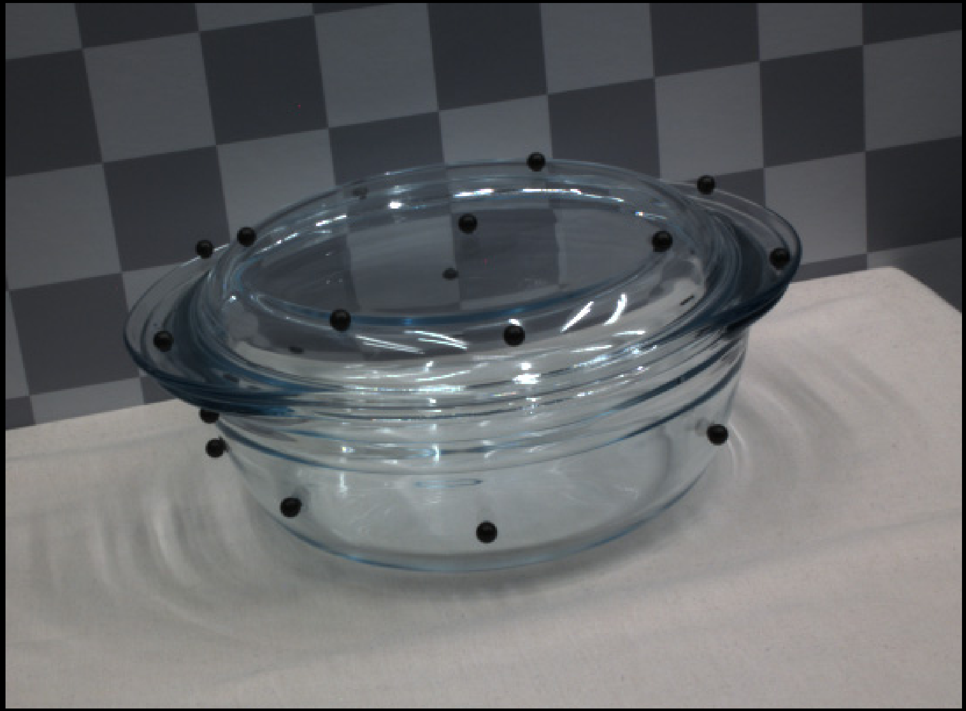


Reference Image

Color Corrected + BRDF + Deformation+ Perspective Envmap

Analysis by Synthesis

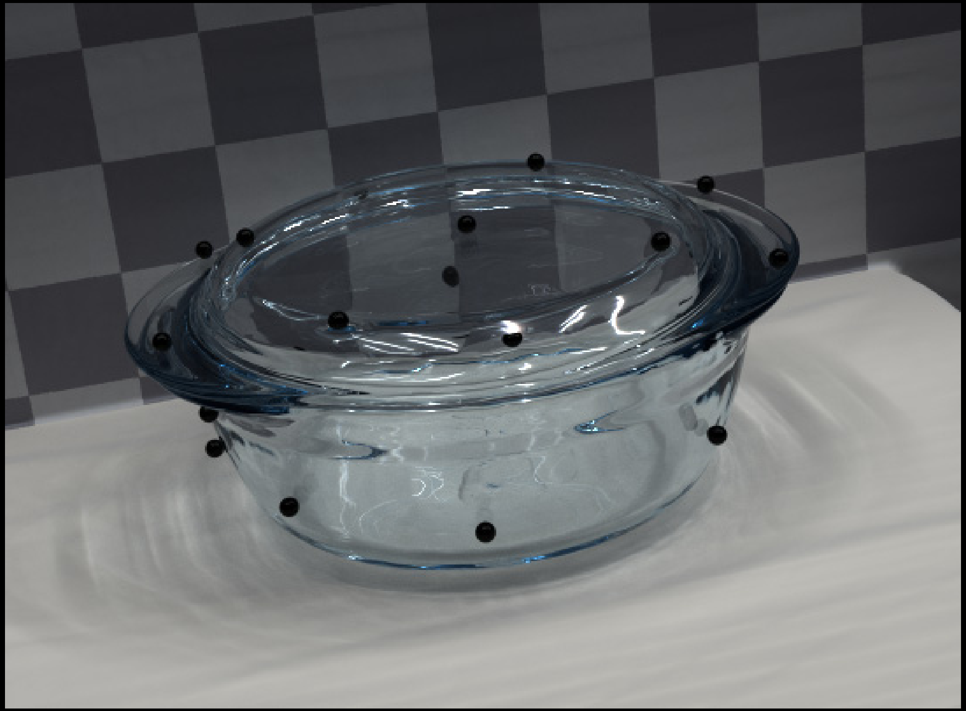
Rendered Image



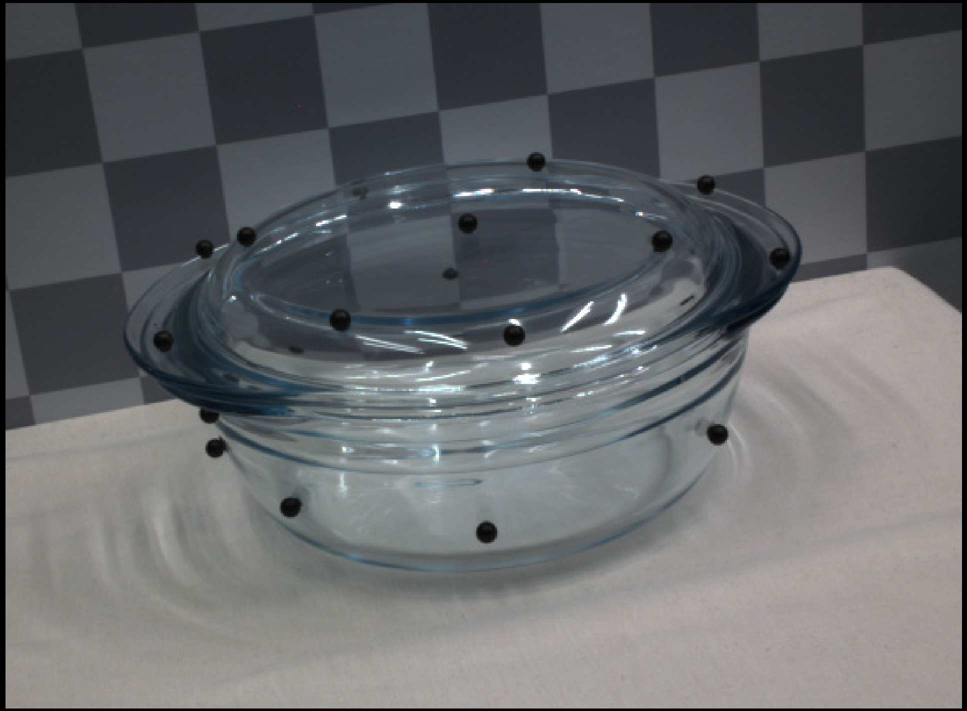
Reference Image

Color Corrected + BRDF + Deformation+ Perspective Envmap+ Chrome Correction

Analysis by Synthesis



Rendered Image



Reference Image

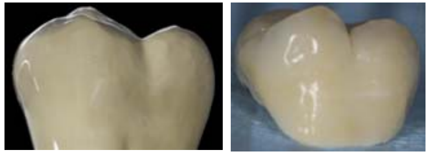
Color Corrected + BRDF + Deformation + Perspective Envmap + Chrome Correction + Absorption (analysis by synthesis)

Geometry & Appearance Digitization

Thank you for your attention

Thanks to Otto Abildgaard, Alessandro Dal Corso, and Jonathan Dyssel Stets.

[Larsen et al. 2012]



render photo

[Frisvad et al. 2005]

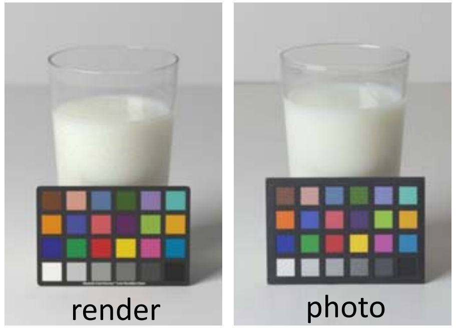
algae in sea ice

[Frisvad 2008]

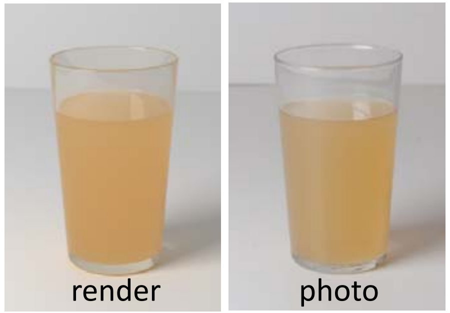
[Andersen et al. 2016]

[Dal Corso et al. 2016]

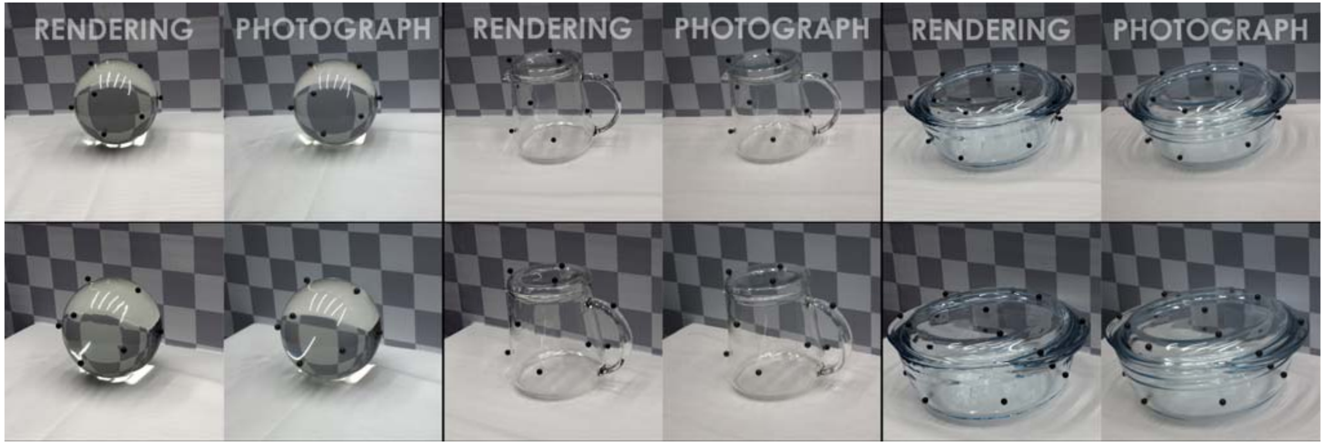
organic low fat milk



unfiltered apple juice



[Stets et al. 2017]



Related work

An Experimental Evaluation of Computer Graphics Imagery, Meyer et al (1986)

Accurate simulation of light and human perceptual comparison.

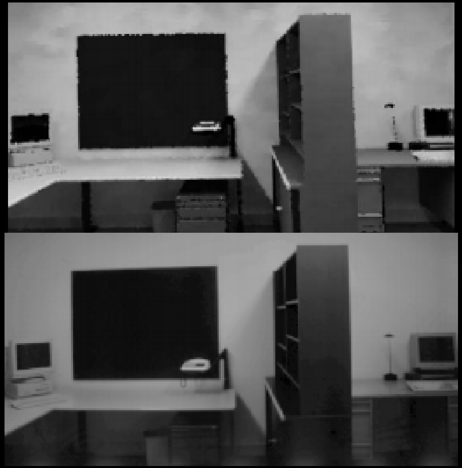
Comparing Real and Synthetic Images: Some Ideas About Metrics, Rushmeier et al (1995)

Techniques for comparing real and synthetic luminance images.

A Concept for Evaluating the Accuracy of Computer Generated Images, Karner et al (1996)

Mathematical comparison of computer simulated images of a real scene.

Meyer et al (1986)



Karner et al (1996)

Applications

- Cultural heritage preservation
- Industrial inspection
- Virtual product placement
- Additive manufacturing
- Agile production
- Estimation of radiometric properties

Geometry & Appearance Digitization

14 | IMAGING | BILLEDANALYSE | DYNAMO | 53 | 06 | 18

FORSKERE SKABER BILLEDER MED MATEMATIK

De tre glaselefanter er alle computerskabte renderinger, og der findes intet foto af en ægte glaselefant. De går fra en simpel fremstilling af glas til stadig mere avancerede renderinger af glaselefanten. Læg bl.a. mærke til, hvordan refleksionen af personen i baggrunden forandres i de sidste to renderinger, hvor forskerne har tilføjet beregninger af lysets brydning i materialet.

Det er lykkedes forskere inden for billedanalyse at skabe billeder af glasobjekter med fotografisk præcision – uden at fotografere dem.

Lotte Krull

DTU Compute og Applied Optics*, Jeppe Revall Frisvad

Grænsen mellem virkelighed og en virtuel verden er blevet yderligere udvisket, efter at et hold forskere har udviklet en ny metode, der kan gengive glasobjekter med fotografisk præcision – vel at mærke uden at fotografere dem. Billederne er digitalt frembragte ved hjælp af computere, som er fodret med matematiske modeller, algoritmer, samt data fra skanninger af objekterne. De digitale gengivelser af de virkelige glasobjekter er så vellykkede, at forskerne i efteråret 2017 fik publiceret deres resultater i det videnskabelige tidsskrift Applied Optics og samtidig opnåede en del medieomtale, især i USA.

"Øjet er nemt at narre, hvis vi kigger på komplekse optiske materialer som glas, der nok er det mest udfordrende materiale at digitalisere. Vi har derfor udviklet en metode, som gør det muligt at afgøre, i hvilken grad digitalt glas svarer til virkeligt glas," fortæller professor emeritus Knut Conradsen, som var med på holdet af de i alt 13 forskere fra DTU Compute samt en fra DTU Fysik.

"Klart glas er nærmest usynligt. Det er lyset og dets brydning gennem materialet, der sætter mennesker i stand til at se det. Samtidig bliver lyset også reflekteret på mange måder i glas. Det anses derfor for lidt af en bedrift at opnå så vellignende renderinger af glasobjekter," siger Knut Conradsen.

Virtuelle glasobjekter

Forskerne havde sat sig for at lave en digital kopi af en virkelig opstilling af tre glasobjekter – en kande, en kugle og et fad med låg – på en ternet baggrund belyst med LED-lys. Først blev hvert objekt fotograferet af to kameraer. Dernæst anvendte forskerne en robotarm til at registrere den nøjagtige placering af de to kameraer for at få data på den rumlige dimension af sceneriet, hvilket er afgørende for at kunne genskabe opstillingerne digitalt.

Den øverste række billeder er computerskabte renderinger af tre glasobjekter. De tre billeder nedenunder er de faktiske fotos af samme objekter.

Efterfølgende har forskerne placeret de digitale glasobjekter i et billede af et sofabord. Det er umuligt at se, at glaskanden, fadet og kuglen aldrig har stået på sofabordet.

DTU | 15

Renderingen illustrerer, hvordan vi kan narres til at tro på, at noget er glas, selvom det bare forvrænger lyset en smule.

1

Her indeholder renderingen nogle af beregningerne af lysets brydning i glasset. Men hvis man kigger på detaljerne, så er det ret mangelfuldt, fordi der mangler absorption samt mange brydninger og interne lysrefleksioner.

2

I denne sidste rendering bruges metoden, som forskerne har publiceret i Applied Optics, hvor man tager en fuld beregning af lysets brydninger, og hvor man tager højde for, at glasset også absorberer en lille del af lyset.

3

Beregninger af lys

De tre glasobjekter blev så CT-skannet, så forskerne kunne få oplysningerne om objekternes udformning, hvilket de oversatte til matematiske modeller. Det kombinerede forskerne med modeller for, hvordan lysstråler bevæger sig gennem forskellige objekter. Sluttelig blev de mange data samlet til de vellignende billeder. Med de oprindelige fotos af glasobjekterne kunne forskerne pixel for pixel sammenligne virkeligheden med den digitale version.

"Der er jo en fantastisk god lighed. Hvis man ser nøje efter, vil man opdage lidt afvigelser i lysspredningen i den digitale version. Men vi har vist, at vi kan genskabe glasobjekter digitalt. Ikke alene fordi vi har en kraftig computer, men også fordi vi har en forståelse af, hvad der sker. Det sætter os i stand til at lave matematiske modeller af, hvordan virkeligheden agerer," siger Knut Conradsen.

Billeder gengiver virkeligheden

Metoden til at skabe virtuelle glasobjekter kan være nyttig til filmproduktion og virtual reality-oplevelser, men her er kravene til absolut naturtro gengivelse ikke så store. Mere relevant er anvendelser i prototyping og designforbedringer af produkter samt i kvalitetssikring.

Men DTU-forskerne har demonstreret en større pointe: Med så avancerede billedanalysekompetencer er det rimeligt at antage, at de renderinger, der skabes ved hjælp af data fra f.eks. røntgensynkrotroner, og som visualiserer enzymer, nervebaner eller fibre i en vindmøllevinge, faktisk viser os en verden, som den ville se ud, hvis man kunne fotografere den. Ω

Knut Conradsen, professor emeritus, DTU Compute, knco@dtu.dk

Geometry & Appearance Digitization

Illustration from Dynamo by Jeppe Revall Frisvad

Geometry & Appearance Digitization

Illustration from Dynamo by Jeppe Revall Frisvad

Geometry & Appearance Digitization

Illustration from Dynamo by Jeppe Revall Frisvad