E016712: Computer Graphics

Computer Vision + Computer Graphics
Simulating and estimating materials and shape

slides by Simon Donné

Me

PhD student at IPI - TELIN, Ghent University
Graduated Spring 2018

Postdoctoral researcher at the Max Planck Institute in Tuebingen, Germany
Working on scene and material reconstruction from RGB images (2 years)

Currently Applied Scientist at Amazon Scout
Working on autonomous vision and robotic perception (1 year+)

Goal

Model a given scene as accurately as possible, both structure and material!

Restrictions:

- Only observations, no “disruptive” measurement
- “Reasonably limited” number of observations
- “Reasonably limited” time for processing

Setting

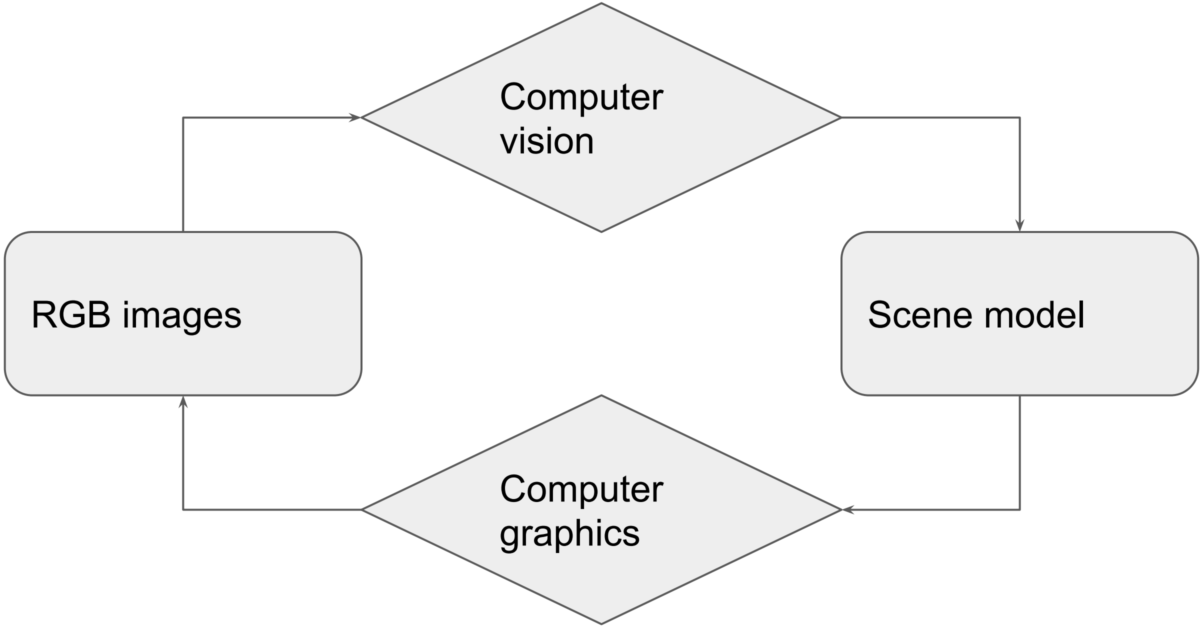
Image processing/computer vision: improve and analyze image content

→ a.o. recognize and reconstruct a 3D model of a scene from 2D images

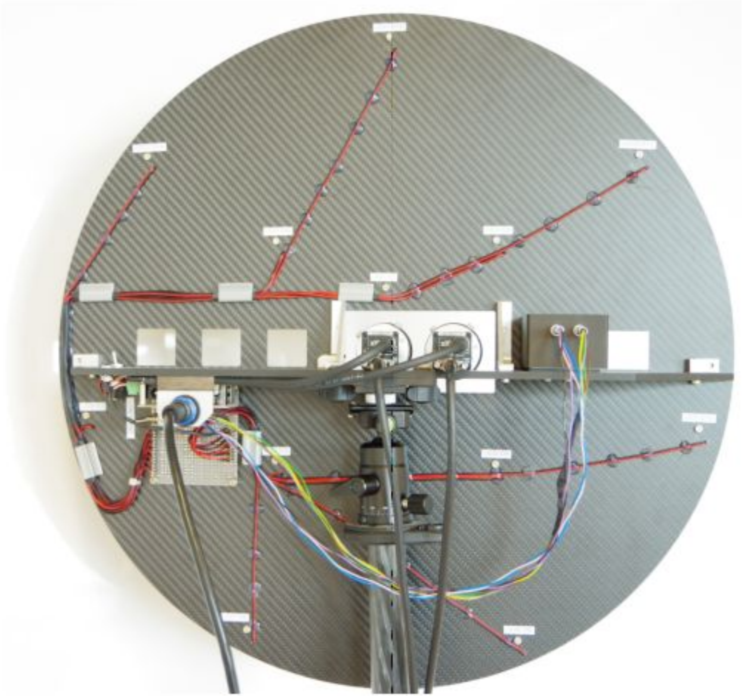
Computer graphics: synthesize images from computer models

→ a.o. rendering a 2D images from a 3D model

Setting

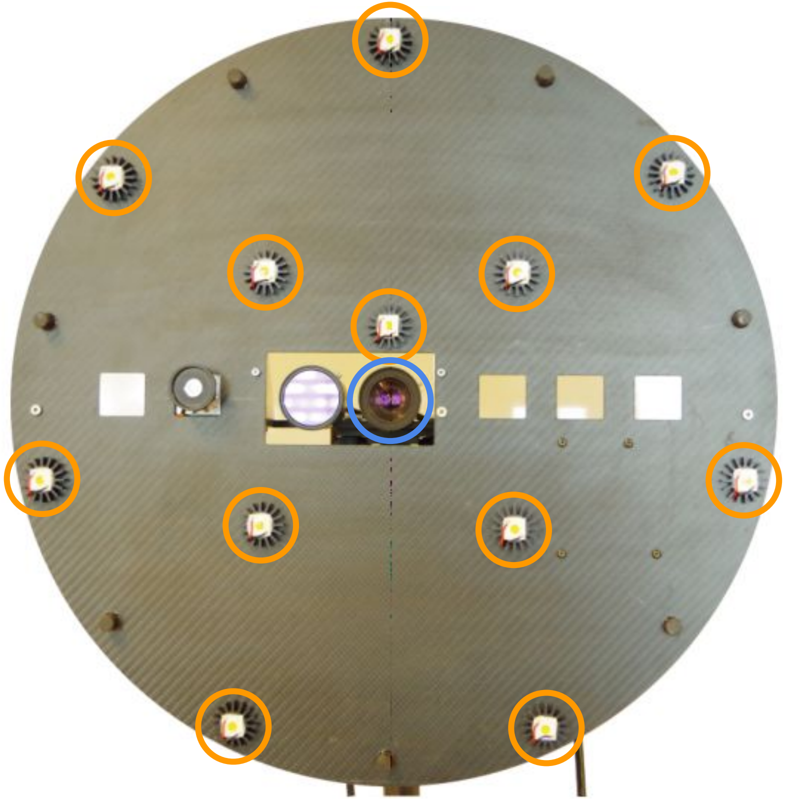


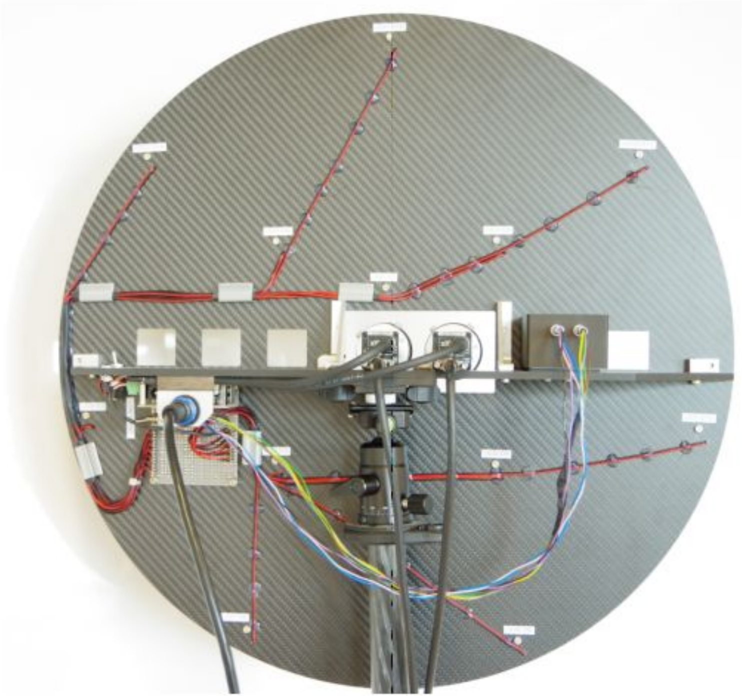
Sensor



Sensor

STRONG point light sources
RGB camera

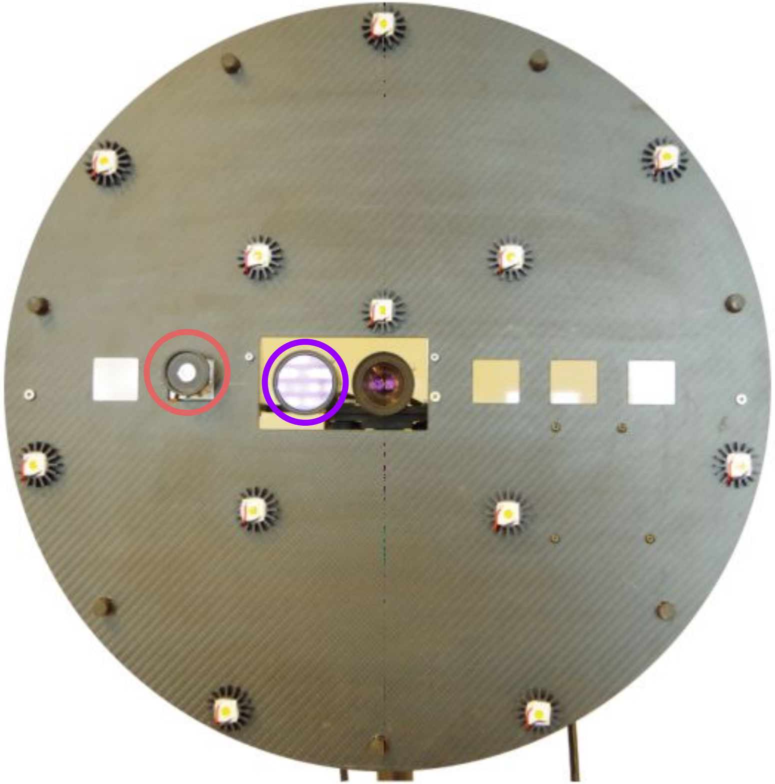


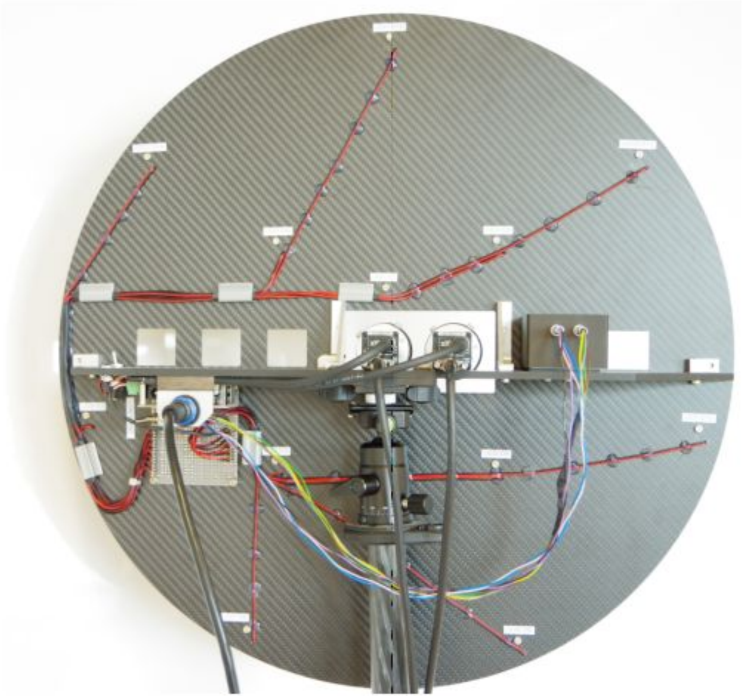


Sensor

Infra-red laser projector

IR camera





Setting

Questions?

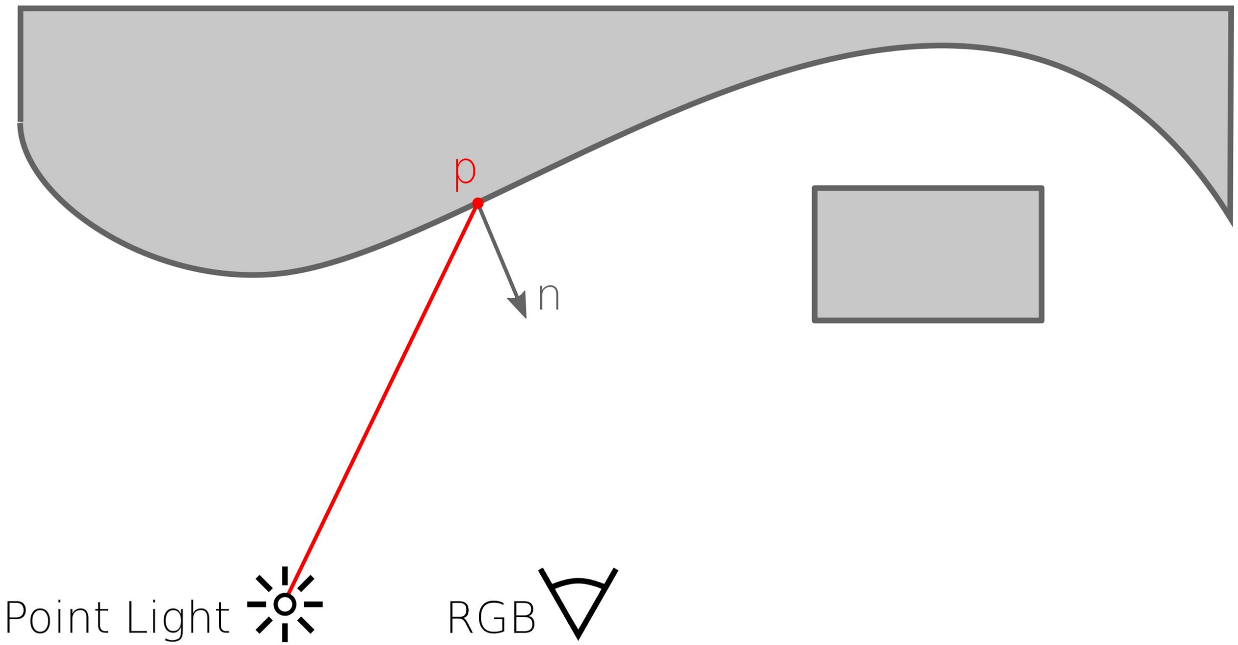
Overview

- **Rendering theory (from scene model to simulated observation)**

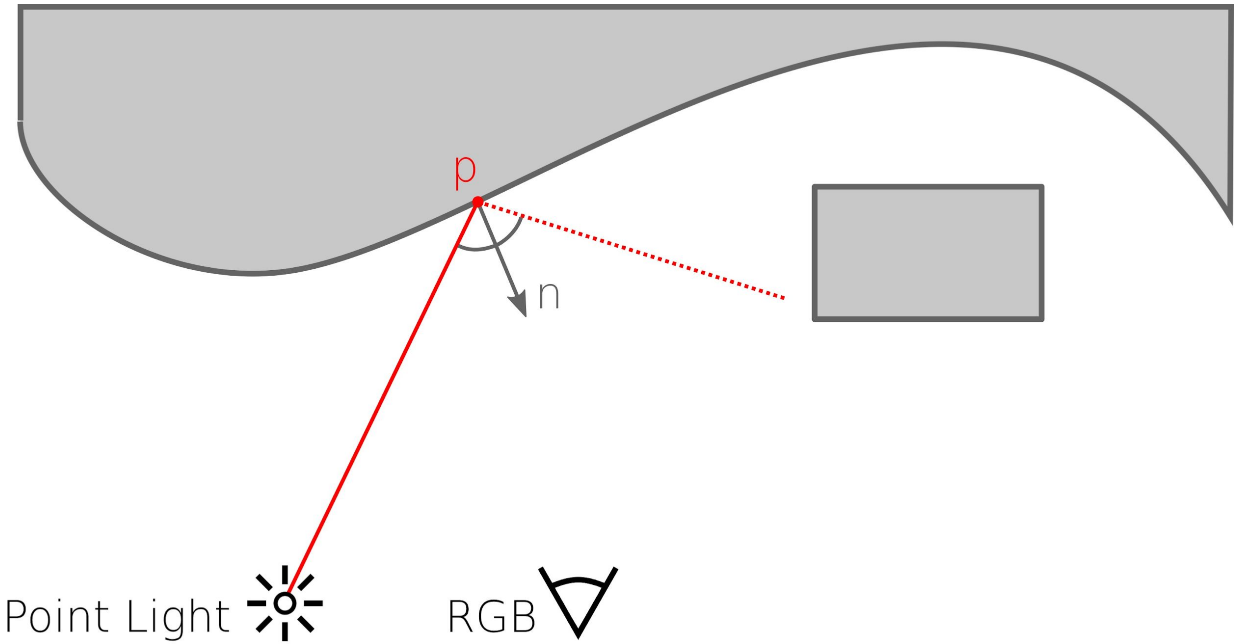
- 3D Reconstruction + material estimation
    - Representation (how we store the scene)
    - Initialization (finding a starting point)
    - Optimization (making simulations match the actual observations)

- Extra: Calibration (know what we are seeing)

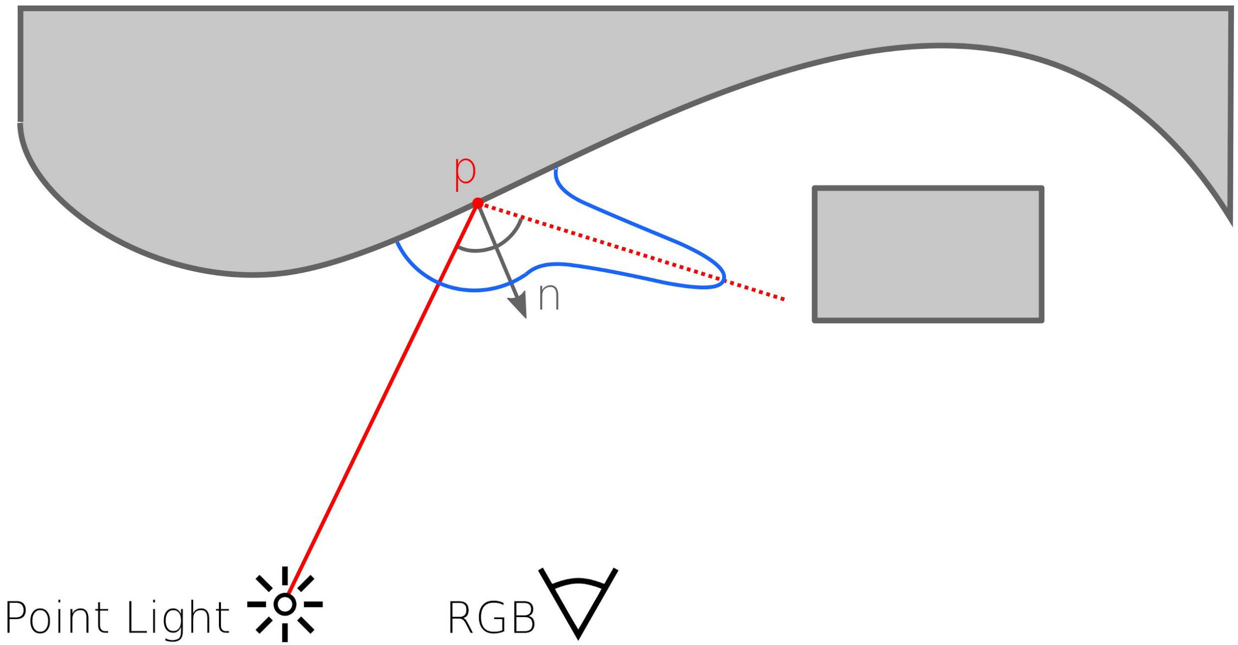
Rendering - light



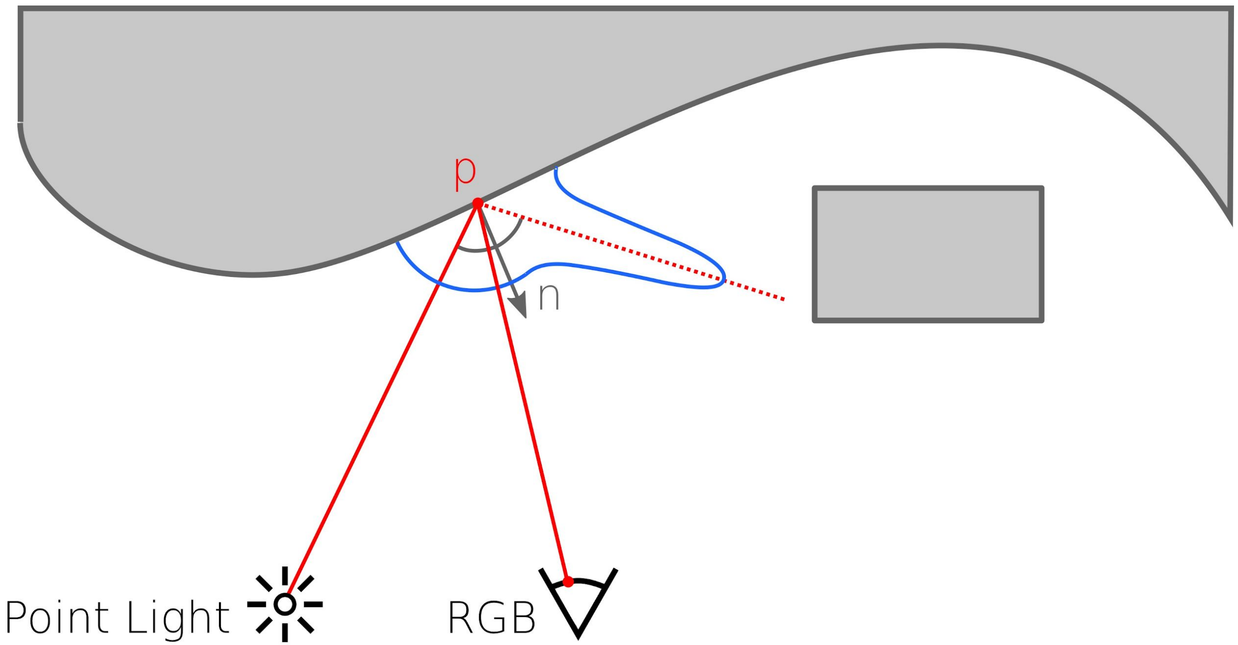
Rendering - surface normal



Rendering - BRDF



Rendering - BRDF

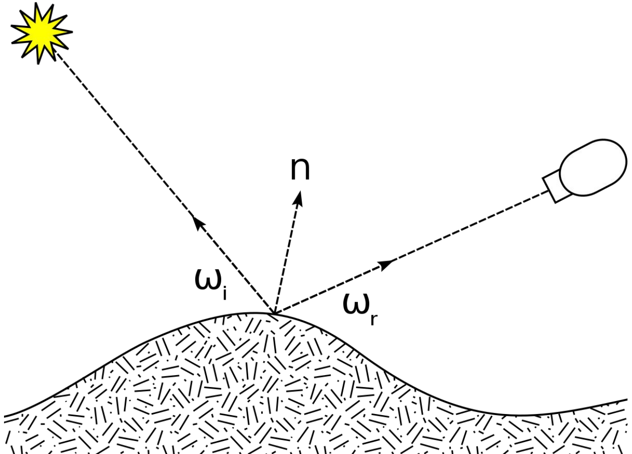


Rendering - BRDF

Bi-directional Reflectance Distribution Function

A material model that, given a an incoming light direction, states what fraction of that light is reflected in any given outgoing direction.

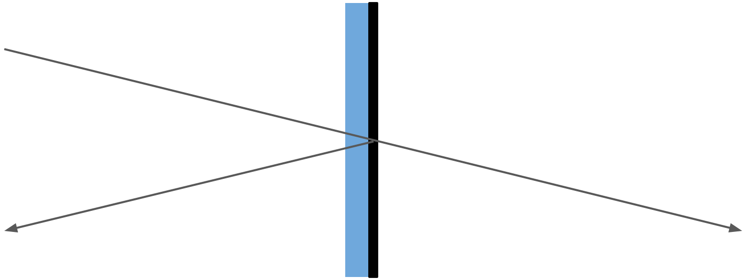
e.g. Lecture 7: Shading & Illumination



Rendering - BRDF examples

Mirror surfaces (highly specular surfaces)

→ light is always reflected along the mirrored ray

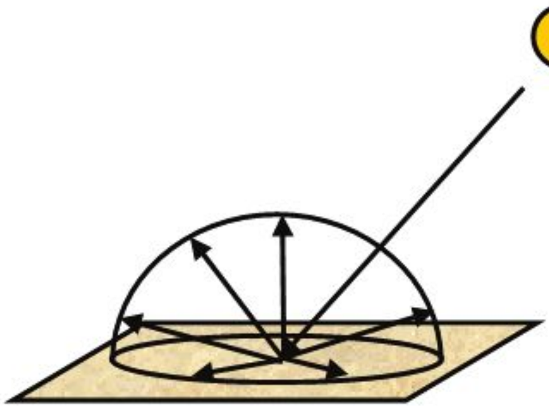


Rendering - BRDF examples

Lambertian surfaces: completely homogeneous

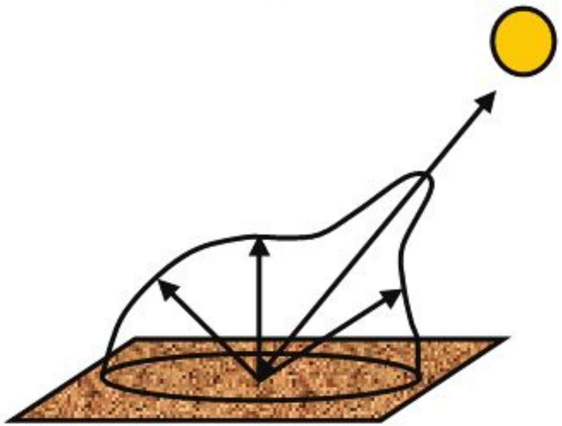
→ light from any incoming direction is equally scattered in all directions

Idealized matte surface



Rendering - BRDF examples

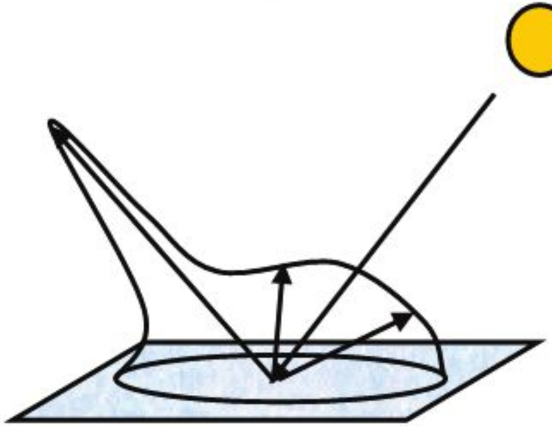
“Reflector” surfaces: reflect light back the way it came



Rendering - BRDF examples

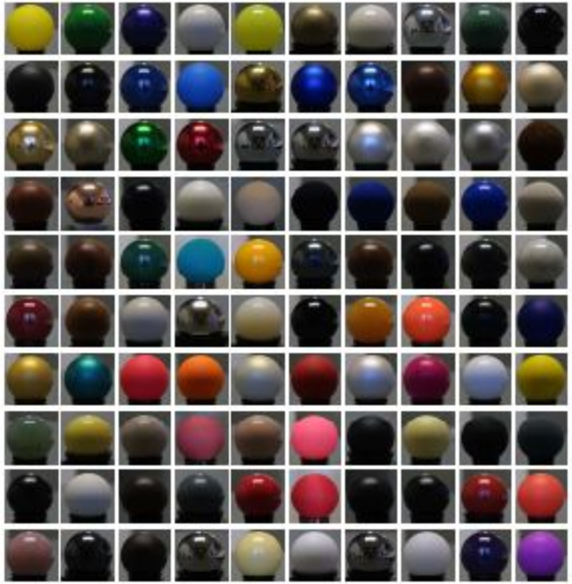
“Normal” surfaces: a mix of lambertian and specular (=mirror) reflection

Nearly all real materials



Rendering - BRDF measurement

Measuring a real BRDF is expensive, slow and difficult



source: MERL material database

Rendering equation (full)

$$L_o(\mathbf{x},\omega_o,\lambda,t) = L_e(\mathbf{x},\omega_o,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega_i,\omega_o,\lambda,t)L_i(\mathbf{x},\omega_i,\lambda,t)(\omega_i \cdot \mathbf{n})\,\mathrm{d}\,\omega_i$$

Observed light
- at point **x**
- in direction ω_o
- with wavelength λ
- at time t

Emitted light

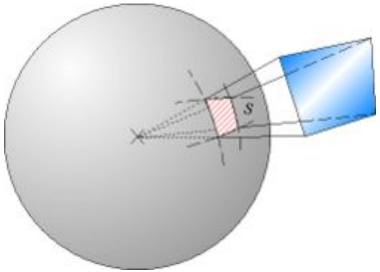
Reflected light
- from all possible $\omega_i \in \Omega$
- by BRDF f_r
- from light with strength L_i
- attenuated by $(\omega_i \cdot \mathbf{n})$

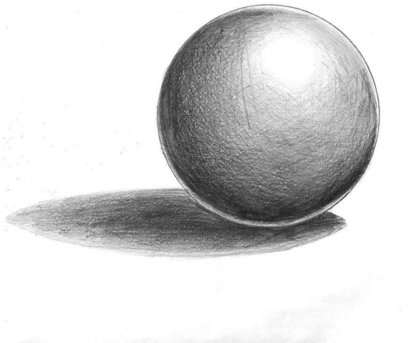
Rendering equation - factor ($\omega_i \cdot n$)

Light falling onto **x** ~ spatial angle of **x** for L_i

1) Light fades $O(1/n^2)$ with distance

2) Less light strikes turned-away surfaces





Rendering equation simplified

$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}},\lambda,t) = L_{\mathrm{e}}(\mathbf{x},\omega_{\mathrm{o}},\lambda,t) \ + \int_{\Omega} f_{\mathrm{r}}(\mathbf{x},\omega_{\mathrm{i}},\omega_{\mathrm{o}},\lambda,t)L_{\mathrm{i}}(\mathbf{x},\omega_{\mathrm{i}},\lambda,t)(\omega_{\mathrm{i}}\cdot\mathbf{n})\,\mathrm{d}\,\omega_{\mathrm{i}}$$

No emittance:

$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}},\lambda,t) = \int_{\Omega} f_{\mathrm{r}}(\mathbf{x},\omega_{\mathrm{i}},\omega_{\mathrm{o}},\lambda,t)L_{\mathrm{i}}(\mathbf{x},\omega_{\mathrm{i}},\lambda,t)(\omega_{\mathrm{i}}\cdot\mathbf{n})\,\mathrm{d}\,\omega_{\mathrm{i}}$$

Single light source:

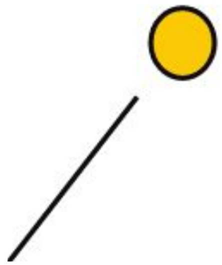
$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}},\lambda,t) = f_{\mathrm{r}}(\mathbf{x},\omega_{\mathrm{i}},\omega_{\mathrm{o}},\lambda,t)L_{\mathrm{i}}(\mathbf{x},\omega_{\mathrm{i}},\lambda,t)(\omega_{\mathrm{i}}\cdot\mathbf{n})$$

Lambertian material:

$$L_{\mathrm{o}}(\mathbf{x},\omega_{\mathrm{o}},\lambda,t) = f_{\mathrm{r}}L_{\mathrm{i}}(\mathbf{x},\omega_{\mathrm{i}},\lambda,t)(\omega_{\mathrm{i}}\cdot\mathbf{n})$$

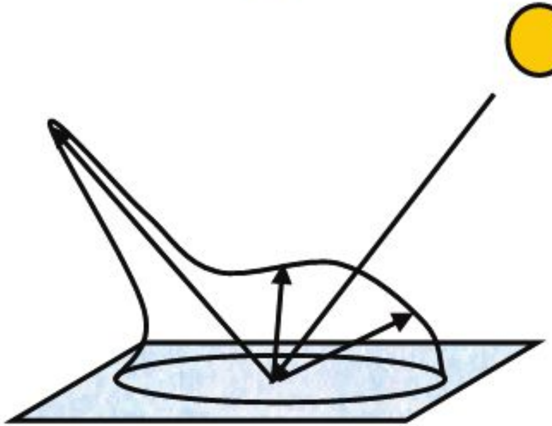
Rendering approximation: single light

We assume only a single light source ...



Rendering approximation: single light

… when in practice every point becomes a light source after reflection!



https://en.wikipedia.org/wiki/Cornell_box

Rendering approximation: single light

This quickly becomes impossible to simulate accurately

In practice: sampling + denoising

- everytime incoming light beam is split
- sample outgoing directions from the BRDF
- consider samples as new light sources

Rendering approximation: single light

This quickly becomes impossible to simulate accurately

For us:

- we capture in a dark tent with especially dark fabric (non-reflective)
- we capture only one object at a time

→ in practice, such multiple-bounce paths are very limited

→ we can still simulate them afterwards, this does not restrict application!

Rendering

Questions?

Overview

- Rendering theory (from scene model to simulated observation)

- **3D Reconstruction + material estimation**
  - Representation (how we store the scene)
  - Initialization (finding a starting point)
  - Optimization (making simulations match the actual observations)

- Extra: Calibration (know what we are seeing)

3D Reconstruction + material estimation

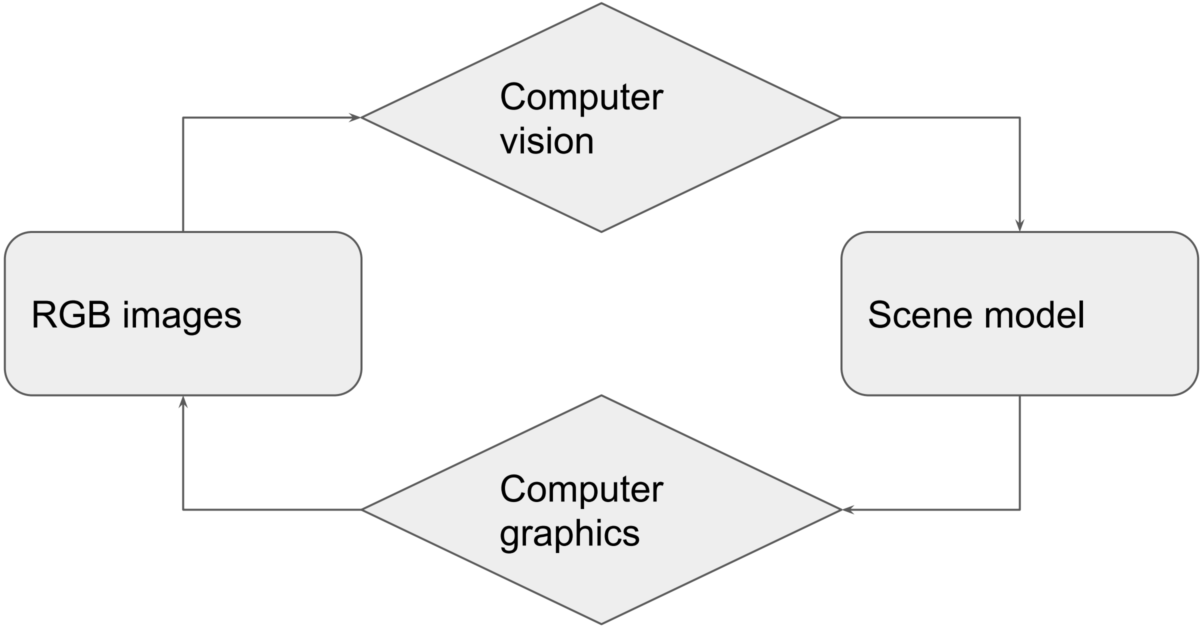
Material and structure are tightly linked

- Accurate structure is required for accurate material estimation
- Accurate material knowledge is required for accurate structure estimation

This means the optimization problem is complex and has a lot of local minima

As we have neither accurate material nor structure knowledge,
we get a rough initialization for both and then refine those estimates

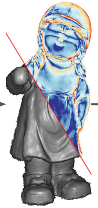
3D Reconstruction + material estimation



3D Reconstruction + material estimation

RGB image measurements

RGB image reconstructions



Model optimization

Scene model

Computer graphics

Overview

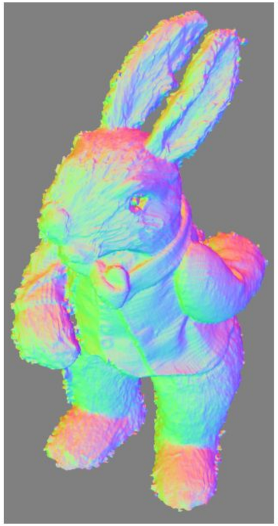
- Rendering theory (from scene model to simulated observation)
- 3D Reconstruction + material estimation
  - **Representation (how we store the scene)**
  - Initialization (finding a starting point)
  - Optimization (making simulations match the actual observations)
- Extra: Calibration (know what we are seeing)

Representation

We say the scene is a set of points in space

Exactly corresponding in size and position to the pixels of a depth image
Oriented according to the normals in a normal image

Geometry



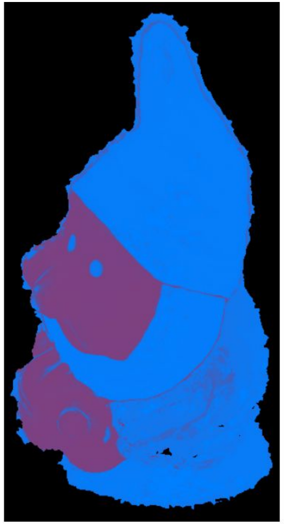
Normals

Representation

We say the scene is a set of points in space

Each of this points has its own lambertian color (albedo)
Its specular behavior is one of a small number of base behaviors

Albedo

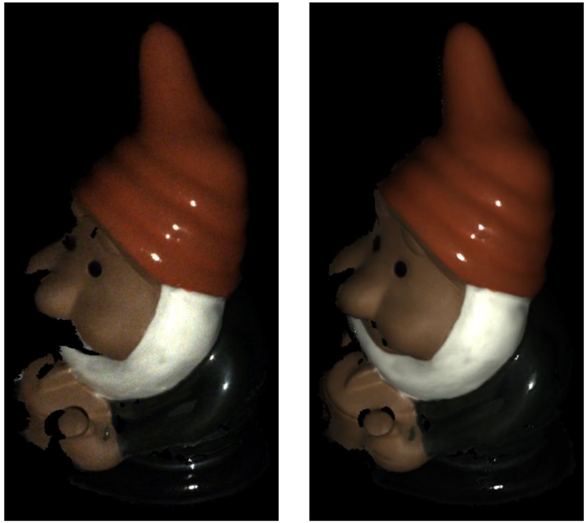


Segmentation

Representation

We say the scene is a set of points in space

Each of this points has its own lambertian color (albedo)
Its specular behavior is one of a small number of base behaviors



Observation Reconstruction

Representation

Questions?

Overview

- Rendering theory (from scene model to simulated observation)

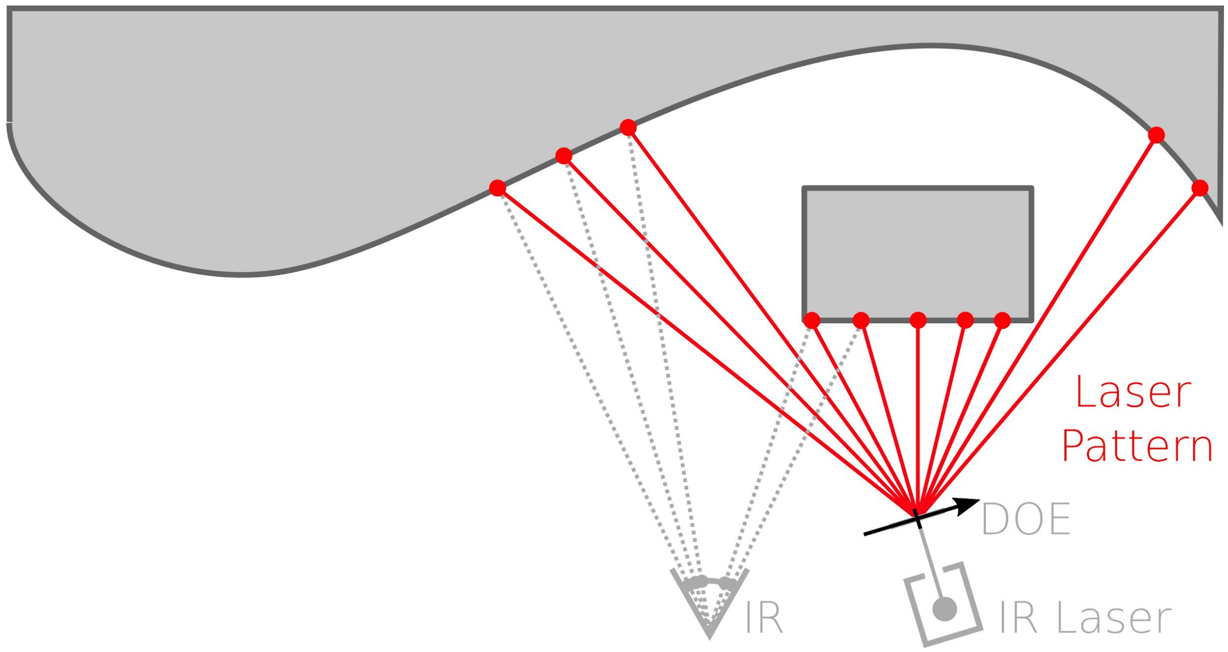
- 3D Reconstruction + material estimation
    - Representation (how we store the scene)
    - **Initialization (finding a starting point)**
    - Optimization (making simulations match the actual observations)

- Extra: Calibration (know what we are seeing)

Initialization - shape

A laser projects a known pattern onto the scene, observed by an IR camera

Initialization - shape

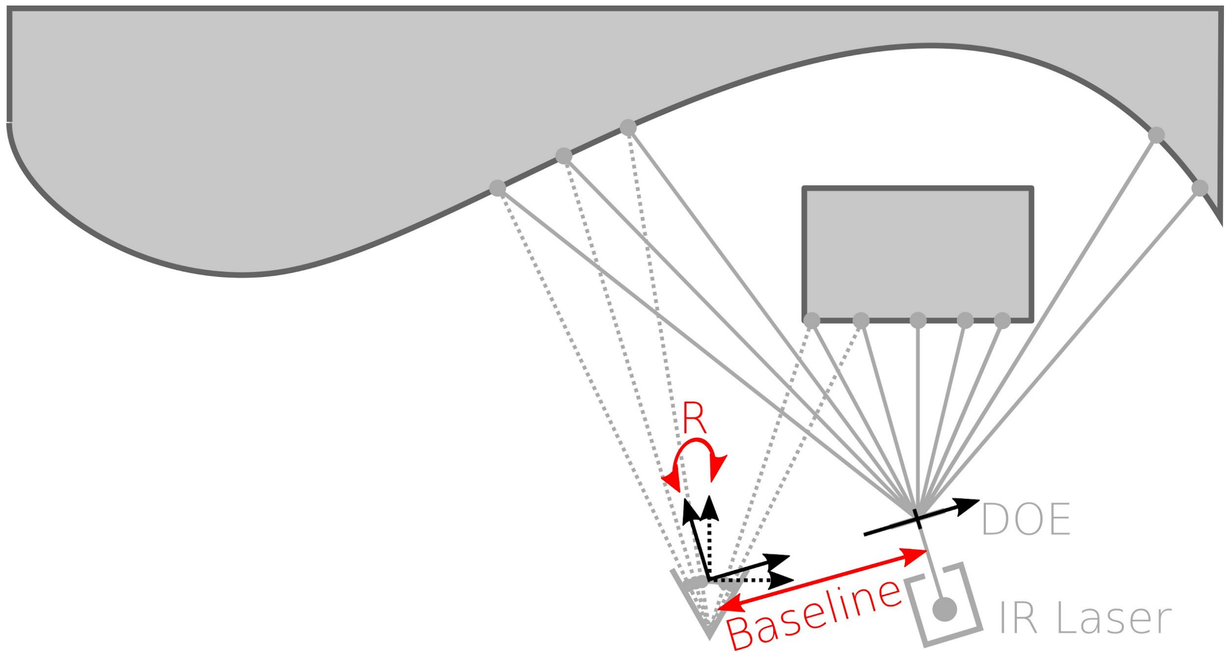


Initialization - shape

A laser projects a known pattern onto the scene, observed by an IR camera

From the known relative position of laser and camera, we can triangulate points in the scene (**stereo matching**)

Initialization - shape

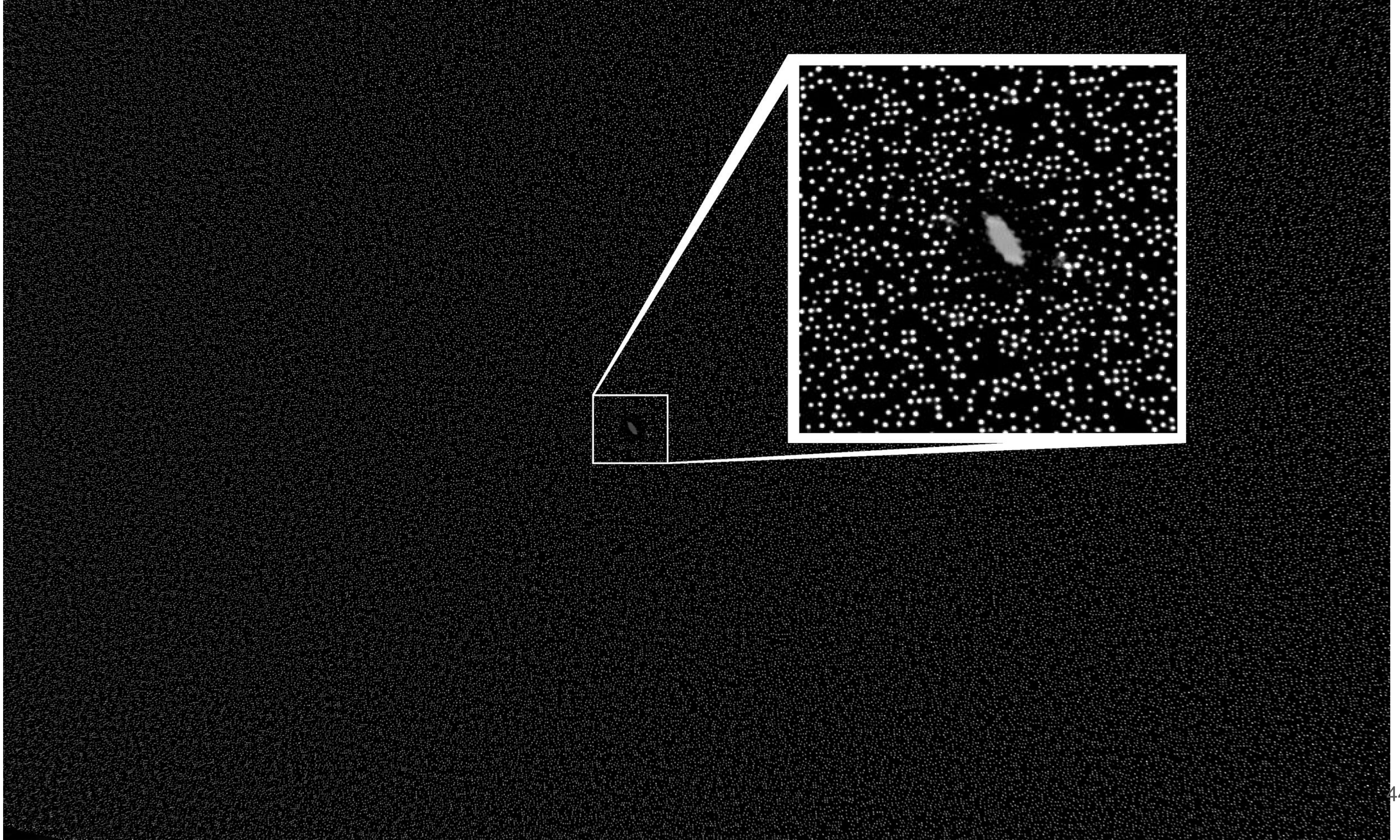


Initialization - shape

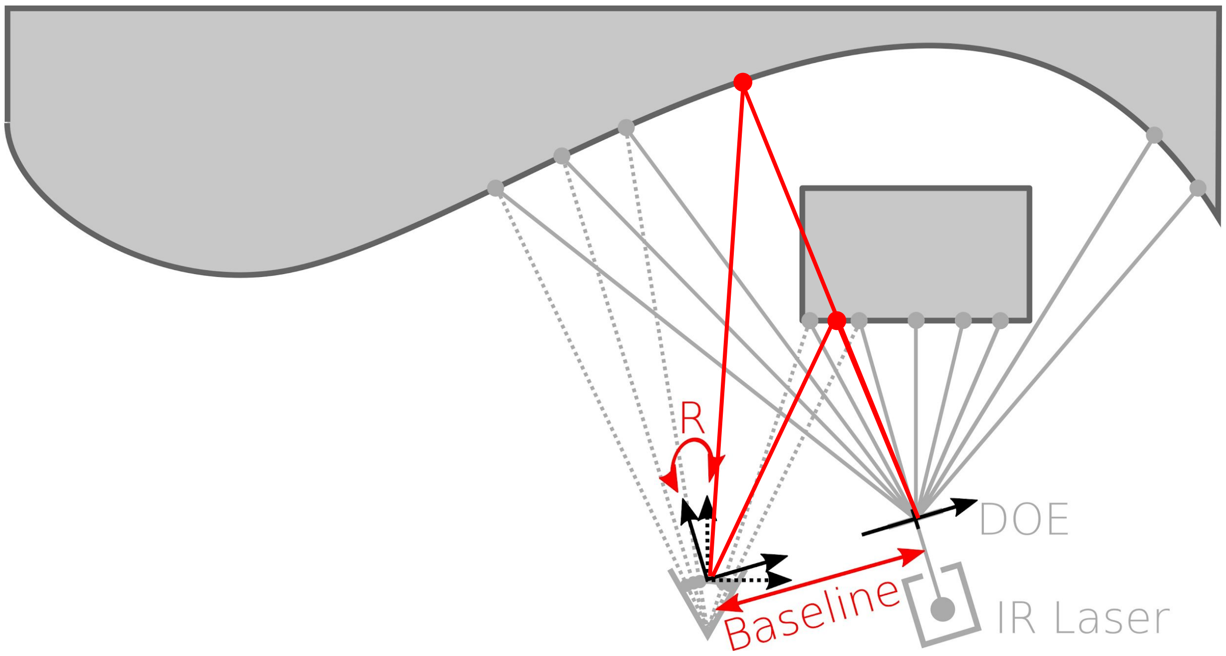
A laser projects a known pattern onto the scene, observed by an IR camera

From the known relative position of laser and camera, we can triangulate points in the scene (stereo matching)

… but only if we can identify any given area of the pattern!



Initialization - shape



Initialization - shape

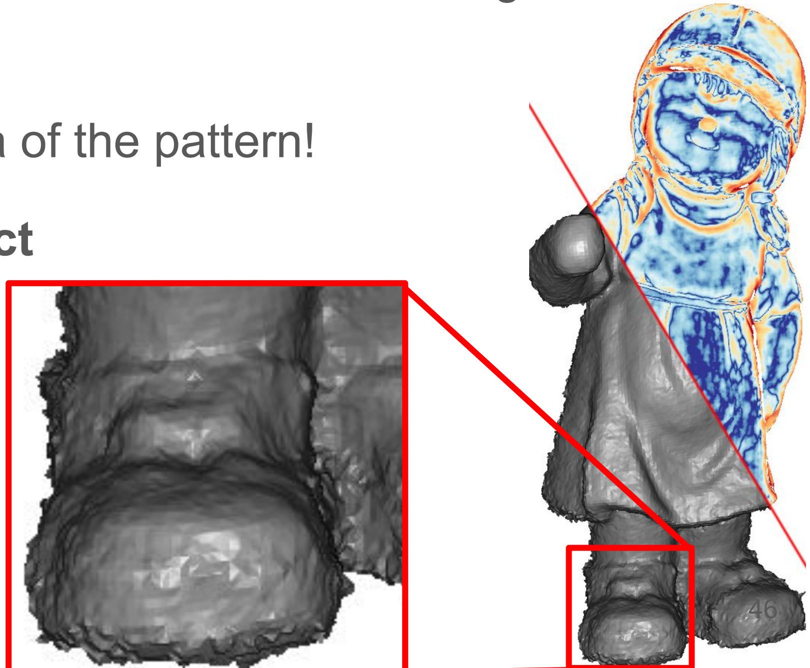
A laser projects a known pattern onto the scene, observed by an IR camera

From the known relative position of laser and camera, we can triangulate points in the scene (stereo matching)

… but only if we can identify any given area of the pattern!

the pattern is **physically limited/not perfect**

- resolution
- randomness
- doesn’t know about materials



Initialization - material

Recall the simplified rendering equation $L_o(\mathbf{x}, \omega_o, \lambda, t) = f_r L_i(\mathbf{x}, \omega_i, \lambda, t)(\omega_i \cdot \mathbf{n})$

This is simple enough we can get a lambertian estimate f_r easily

E.g. by minimizing the mean squared error:

$$\min \sum \| L_o - f_r L_i (\omega_i \cdot n) \|^2$$

$$\Rightarrow f_r = \sum L_o L_i(\omega_i \cdot n) / \sum L_i^2(\omega_i \cdot n)^2$$

Initialization

Questions?

Overview

- Rendering theory (from scene model to simulated observation)

- 3D Reconstruction + material estimation
  - Representation (how we store the scene)
  - Initialization (finding a starting point)
  - **Optimization (making simulations match the actual observations)**

- Extra: Calibration (know what we are seeing)

Optimization - photometric loss function

$$\frac{1}{N}\sum_i \sum_p \| \varphi_p^i \left[\mathcal{I}_i(\pi_i(\mathbf{x}_p)) - \mathcal{R}_i(\mathbf{x}_p, \mathbf{n}_p, f_p)\right] \|_1$$

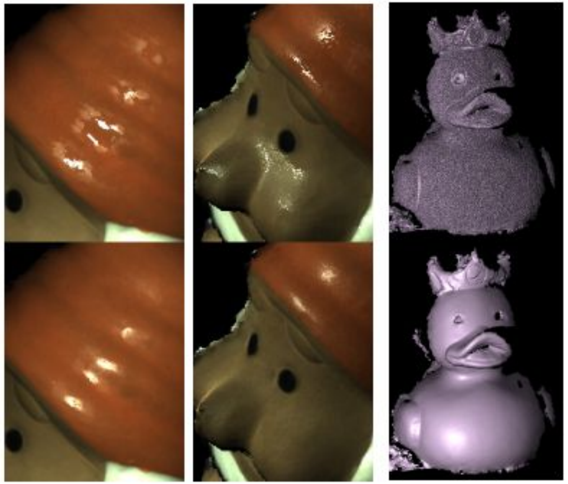
Our actual observation should match the simulated observation

Optimization - important auxiliary loss functions

The object should be smooth: its normal is slowly changing

The object should be consistent: its BRDF is consistent

Otherwise, we end up modeling noise in the reconstruction!



Optimization - “backwards pass” of the rendering equation

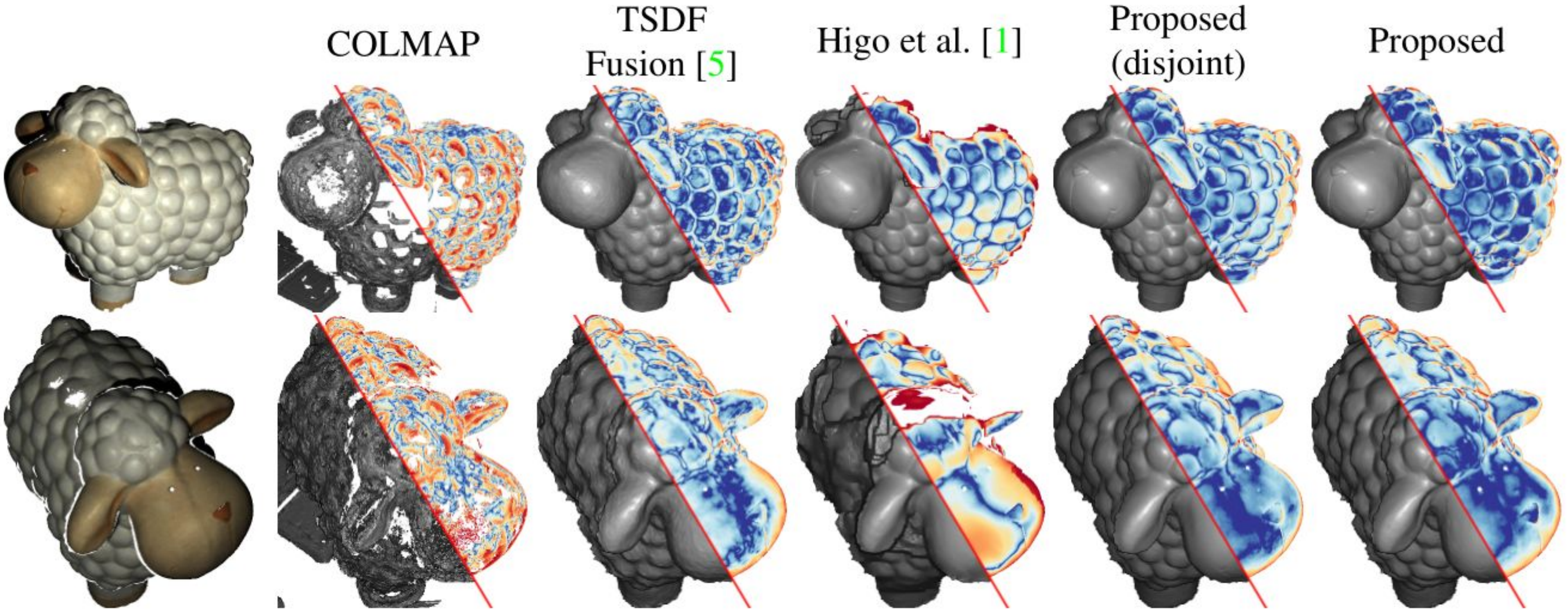
The loss function is complex (and cannot be solved closed form) …

… but fully differentiable

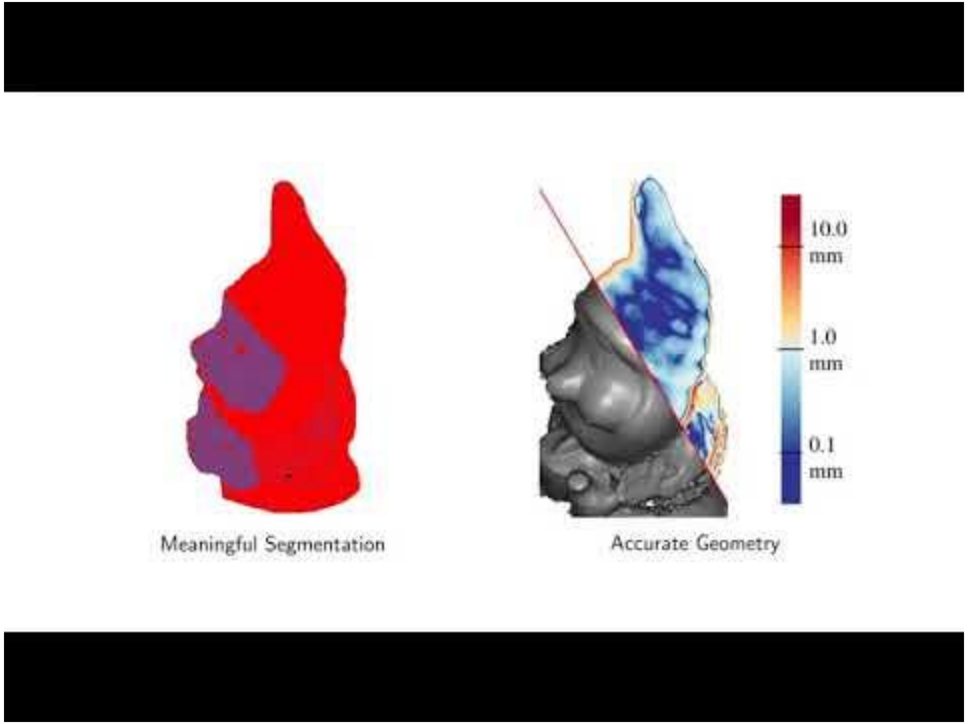
So we optimize by calculating the derivative of the loss function w.r.t. the 3D structure and the material to improve our reconstruction.

(and we cheat by implementing it in PyTorch, which does differentiation for us)

Optimization - “backwards pass” of the rendering equation



Optimization - result (relighting/new viewpoints)



https://youtu.be/_xxSQPD9qU0?t=47

Optimization

Questions?

Overview

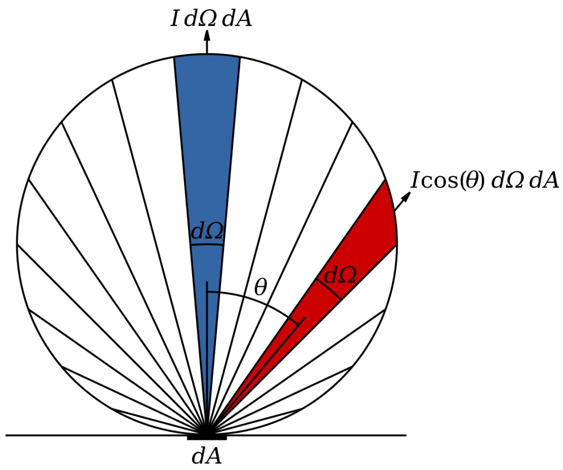
- Rendering theory (from scene model to simulated observation)

- 3D Reconstruction + material estimation
    - Representation (how we store the scene)
    - Initialization (finding a starting point)
    - Optimization (making simulations match the actual observations)

- **Extra: Calibration (know what we are seeing)**

Calibration - the light sources & vignetting

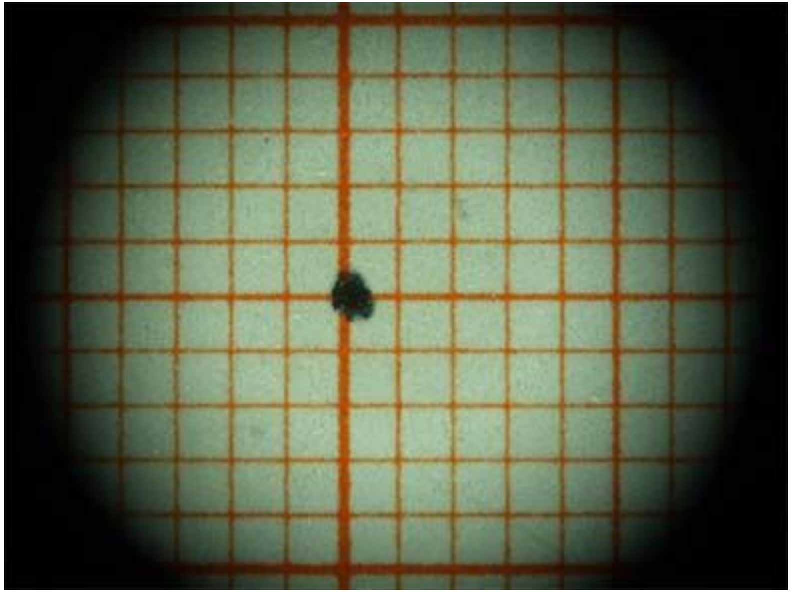
In practice, lights are not perfect point lights



https://en.wikipedia.org/wiki/Lambert%27s_cosine_law

Calibration - the light sources & vignetting

In practice, cameras **do** care where light comes from

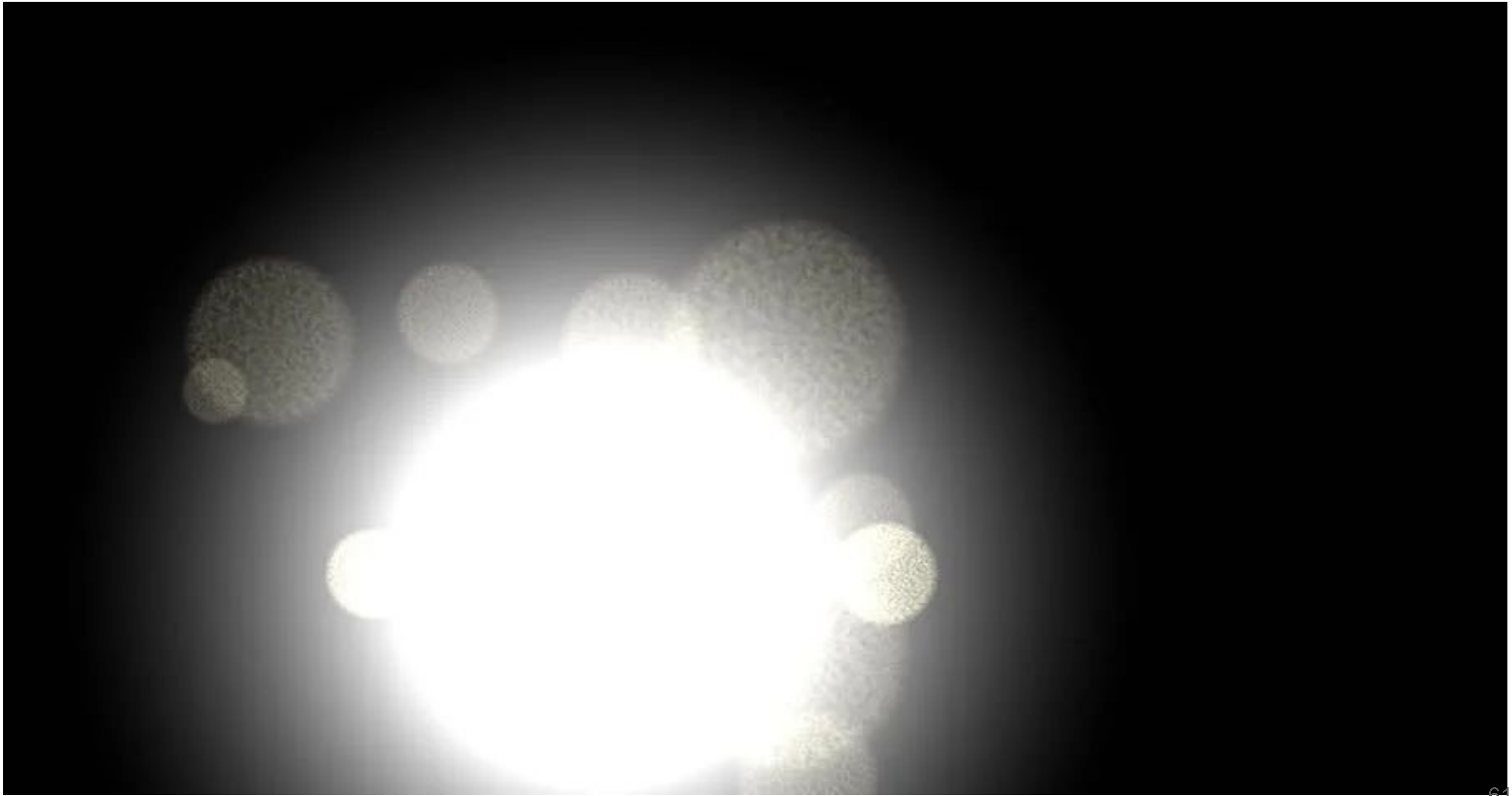


https://en.wikipedia.org/wiki/Vignetting

Calibration - the light sources & vignetting

Brightness, color and profile need to be estimated: L_i from the rendering eq.

$$L_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}, \lambda, t) = f_{\mathrm{r}} L_{\mathrm{i}}(\mathbf{x}, \omega_{\mathrm{i}}, \lambda, t)(\omega_{\mathrm{i}} \cdot \mathbf{n})$$



Calibration - the light sources & vignetting

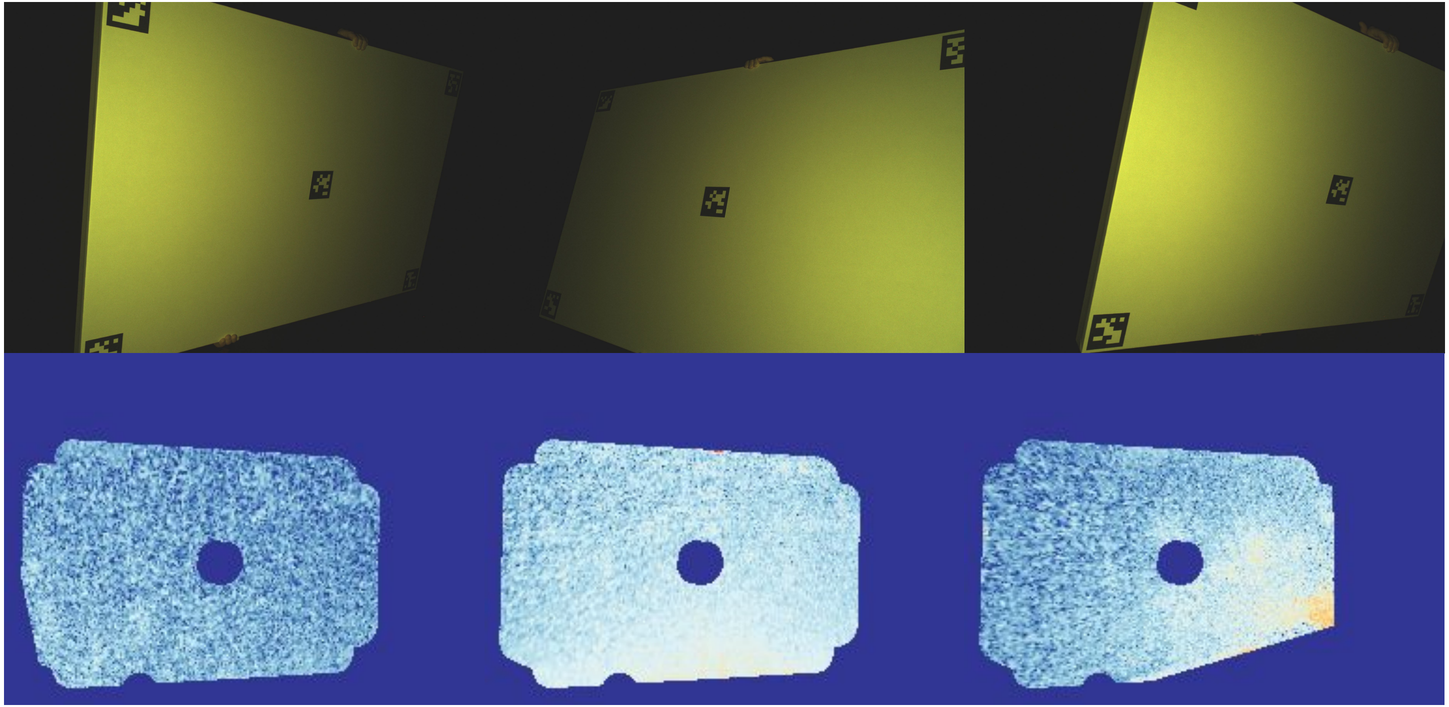
Brightness, color and profile need to be estimated: L_i from the rendering eq.

$$L_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o}, \lambda, t) = f_\mathrm{r} L_\mathrm{i}(\mathbf{x}, \omega_\mathrm{i}, \lambda, t)(\omega_\mathrm{i} \cdot \mathbf{n})$$

But lights are so bright this is hard to measure directly.
Perfect application for exactly this framework!

This time we use something with a known shape - a flat white wall
and instead optimize over the BRDF f_r **and** the light behaviour L_i

Calibration - the light sources & vignetting



Calibration - the light sources & vignetting

By capturing enough known poses, we can decouple vignetting and light attenuation.

This lets us model light sources and vignetting
way more accurately than previously

Full attenuation model rather than a single-parameter function!

Calibration - the light sources & vignetting

Questions?

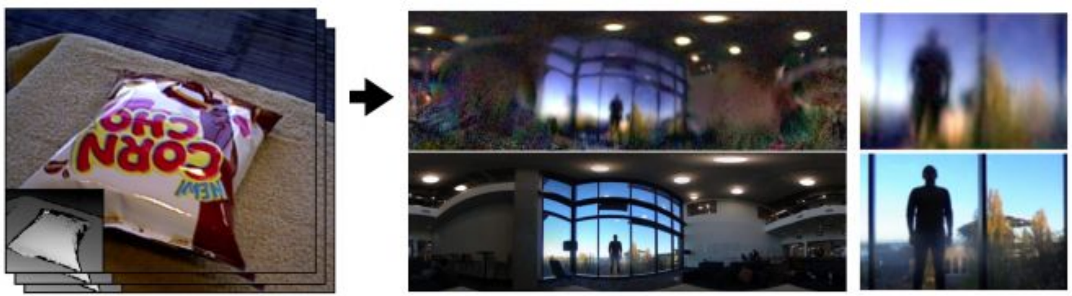
Further reading / related work

A hand-held photometric stereo camera for 3-d modeling
Tomoaki Higo, Yasuyuki Matsushita, Neel Joshi, and Katsushi Ikeuchi (ICCV 2009)
(good intro paper, lambertian-only)

On Joint Estimation of Pose, Geometry and svBRDF from a Handheld Scanner
Carolin Schmitt*, Simon Donné*, Gernot Riegler, Vladlen Koltun, Andreas Geiger (CVPR 2020)
(presented here)

Further reading / related work

Seeing the World in a Bag of Chips
Jeong Joon Park, Aleksander Holynski, Steve Seitz (CVPR 2020)
(really nice application!)



(a) Input images (b) Estimated environment (top), ground truth (bottom) (c) Zoom-in

Further reading / related work

Practical SVBRDF Acquisition of 3D Objects with Unstructured Flash Photography
Giljoo Nam, Joo Ho Lee, Diego Gutierrez, Min H. Kim (SIGGRAPH Asia 2018)
(Less accurate, but based on cell phone captures!)