Announcements

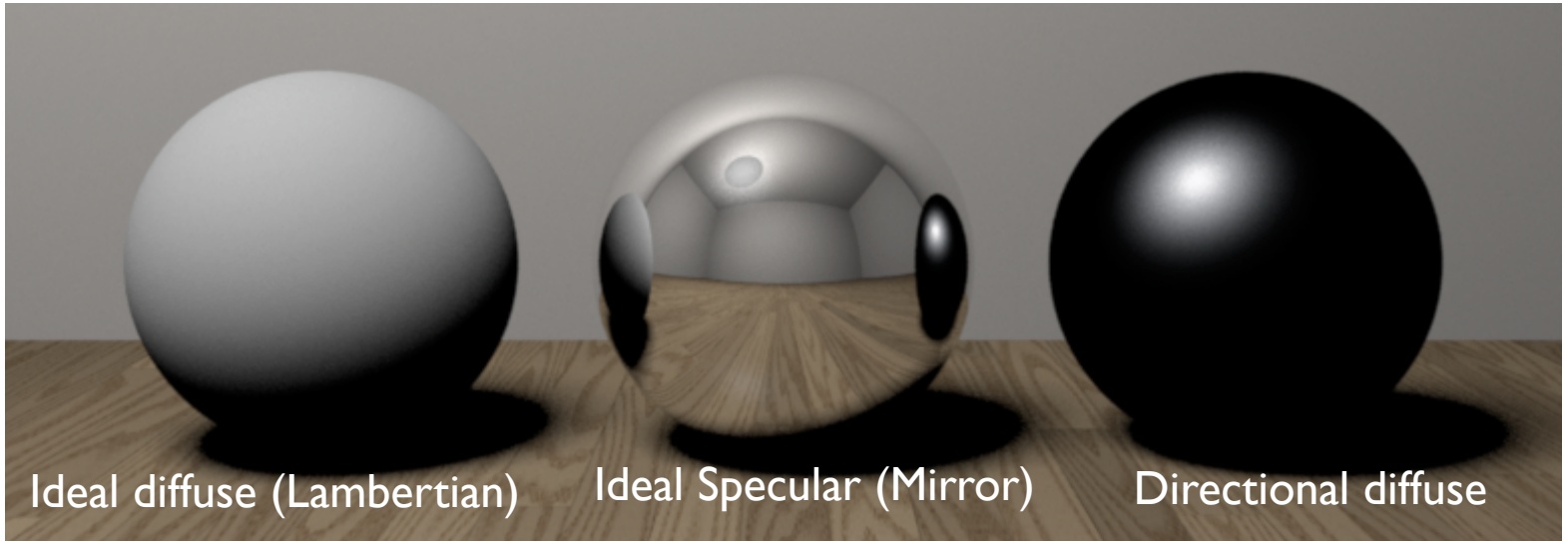
- Assignment 3 deadline extended to Friday March 27, 11:59pm EDT
- Artifact due at the same time

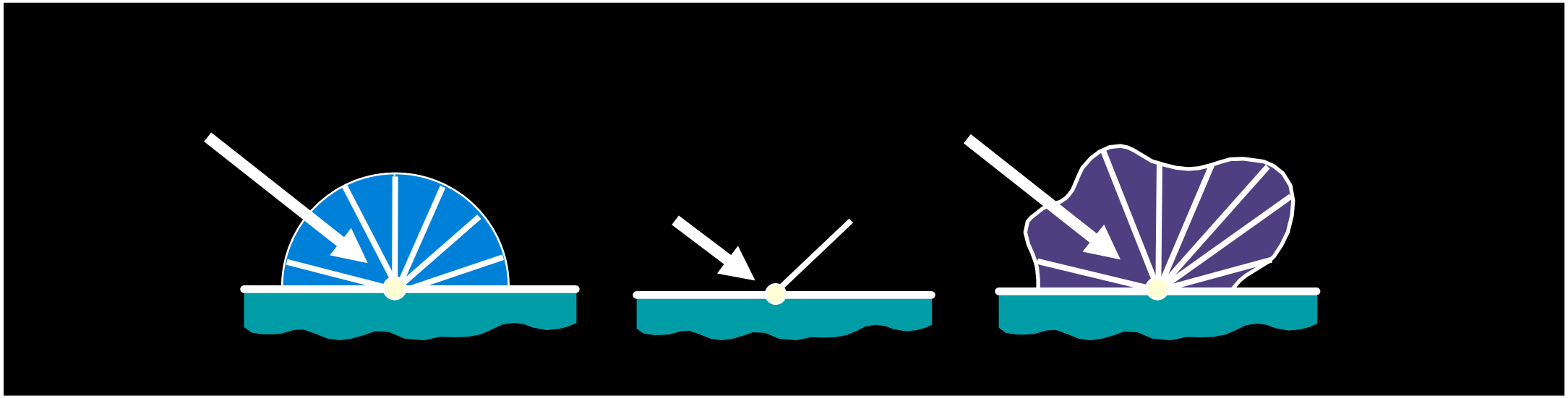
A Note on COVID-19 Chaos

- Everyone is having to do a bit of improvisation…
  - We know it's a stressful time
  - We will try to be patient, flexible, and understanding
  - Please be patient, flexible, and understanding

- Exams will be graded by the end of the week
- Lecture videos and slides will go up today or tomorrow
- Let us know if you have problems or concerns
  - Some details may take a while to work out, but we will do our best to make sure you get the most out of the class

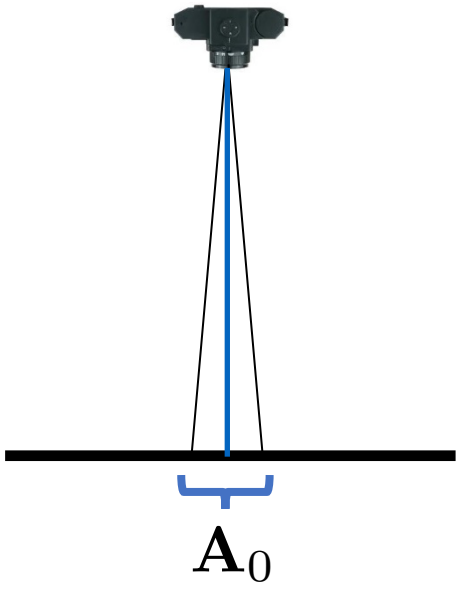
Diffuse and Specular Reflectance





Lambertian Surfaces: Appearance vs Reflected Photons

- Appearance is the same from every angle
- Radiant Intensity? (how many photons reflected per angle)

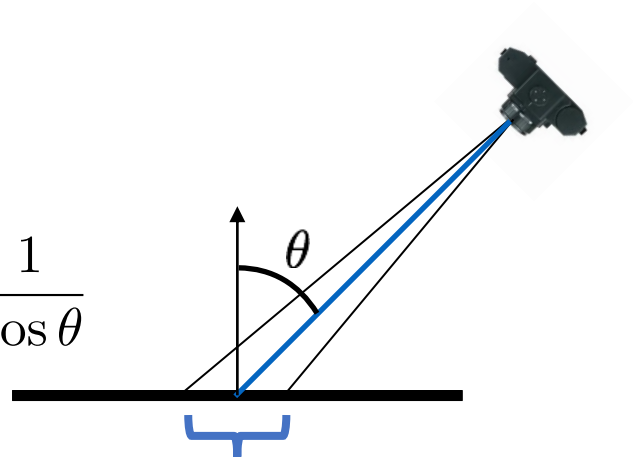


$$Pixel = \mathbf{B}_0\mathbf{A}_0 = \mathbf{B}_\theta\mathbf{A}_\theta$$

Lambertian Surfaces: Appearance vs Reflected Photons

- Appearance is the same from every angle
- Radiant Intensity? (how many photons reflected per angle)

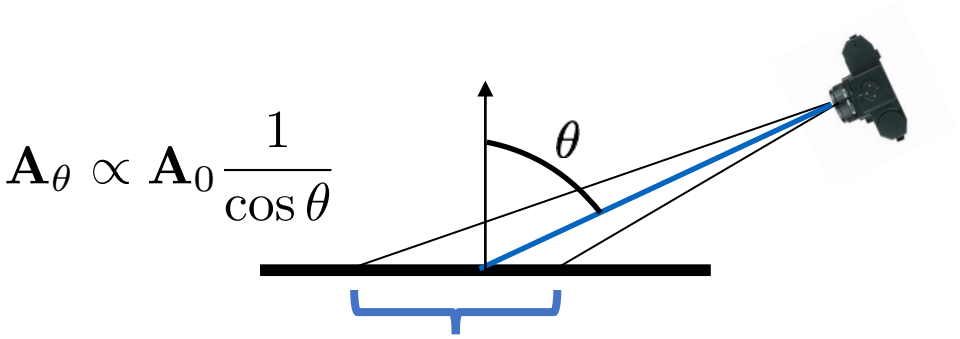
$$\mathbf{A}_\theta \propto \mathbf{A}_0 \frac{1}{\cos\theta}$$

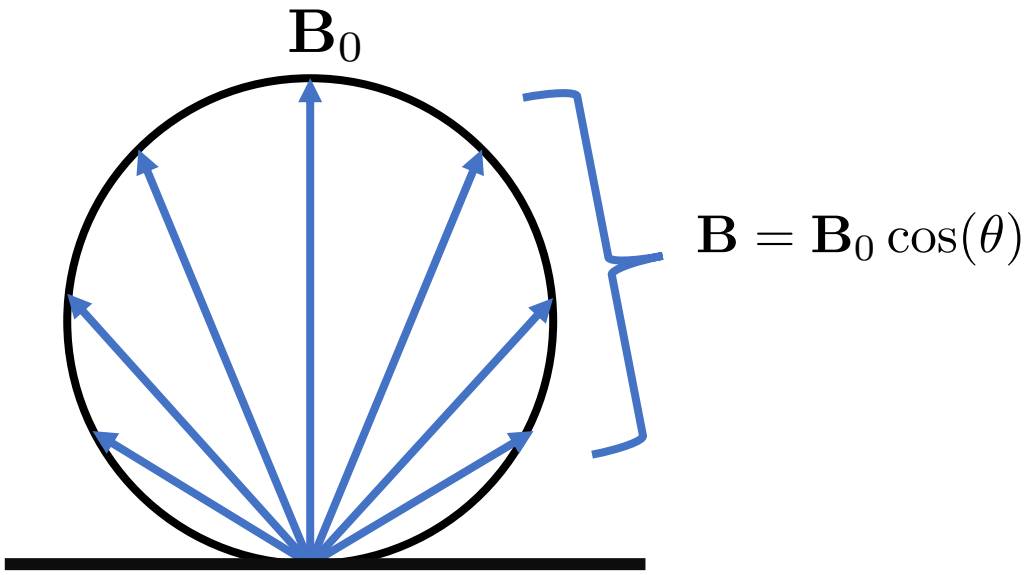


$$Pixel = \mathbf{B}_0\mathbf{A}_0 = \mathbf{B}_\theta\mathbf{A}_\theta$$

Lambertian Surfaces: Appearance vs Reflected Photons

- Appearance is the same from every angle
- Radiant Intensity? (how many photons reflected per angle)





Lambert's cosine law: $\mathbf{B} = \mathbf{B}_0 \cos(\theta)$

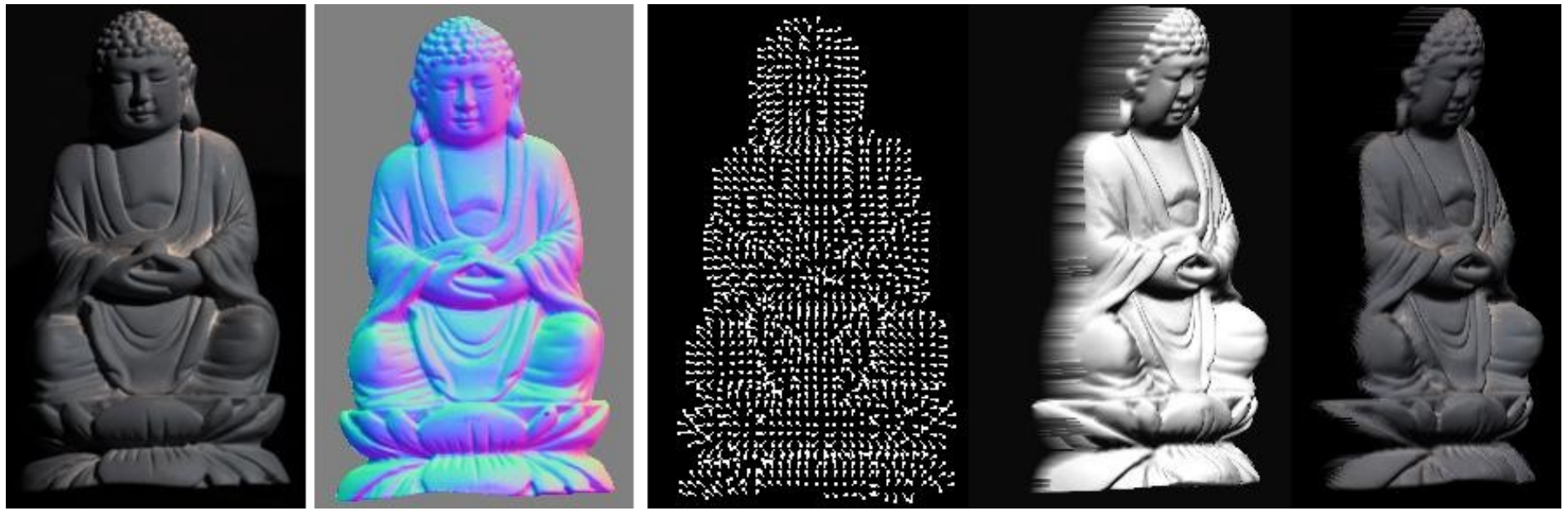
$$Pixel = \mathbf{B}_0\mathbf{A}_0 = \mathbf{B}_\theta\mathbf{A}_\theta$$

$$Pixel = \mathbf{B}_0\mathbf{A}_0\frac{\cos\theta}{\cos\theta}$$

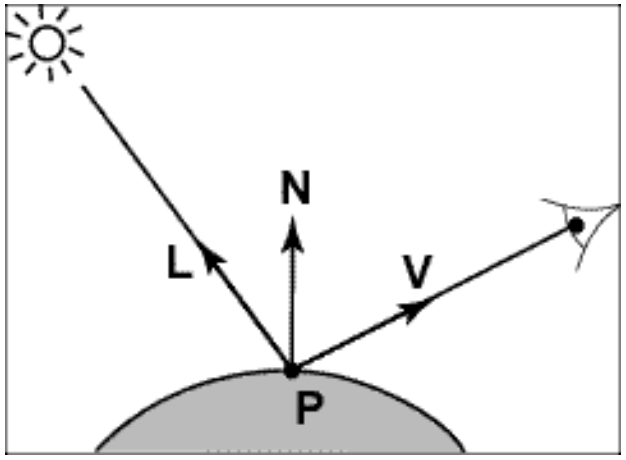
CS5670 : Computer Vision

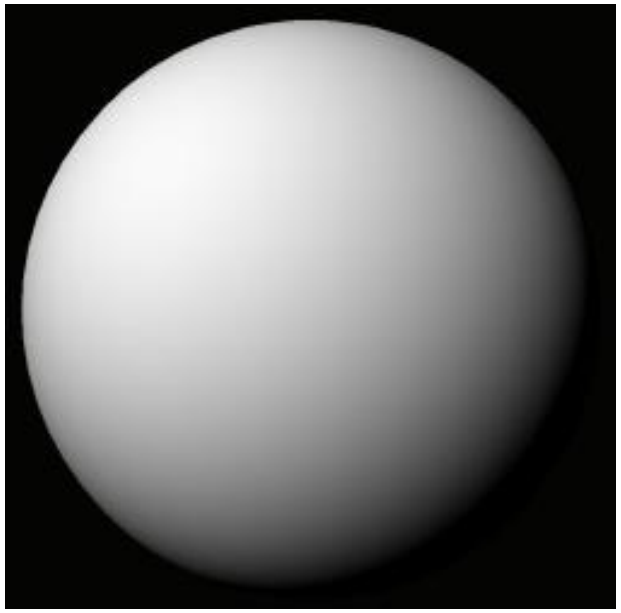
Abe Davis | (most slides from Noah Snavely)

Photometric stereo



Recap:
Lambertian (Diffuse) Reflectance





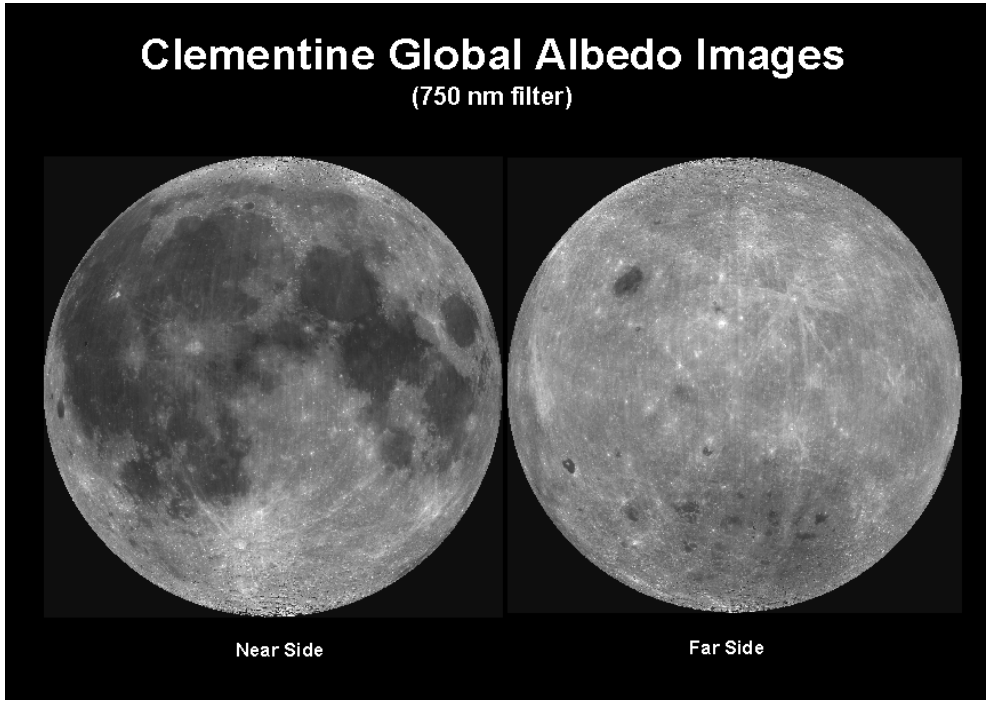
$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

- I : observed image intensity
- k_d : object albedo
- $\mathbf{N}$: surface normal
- $\mathbf{L}$: light source direction

Lambertian sphere with constant albedo lit by a directional light source

Sample albedos

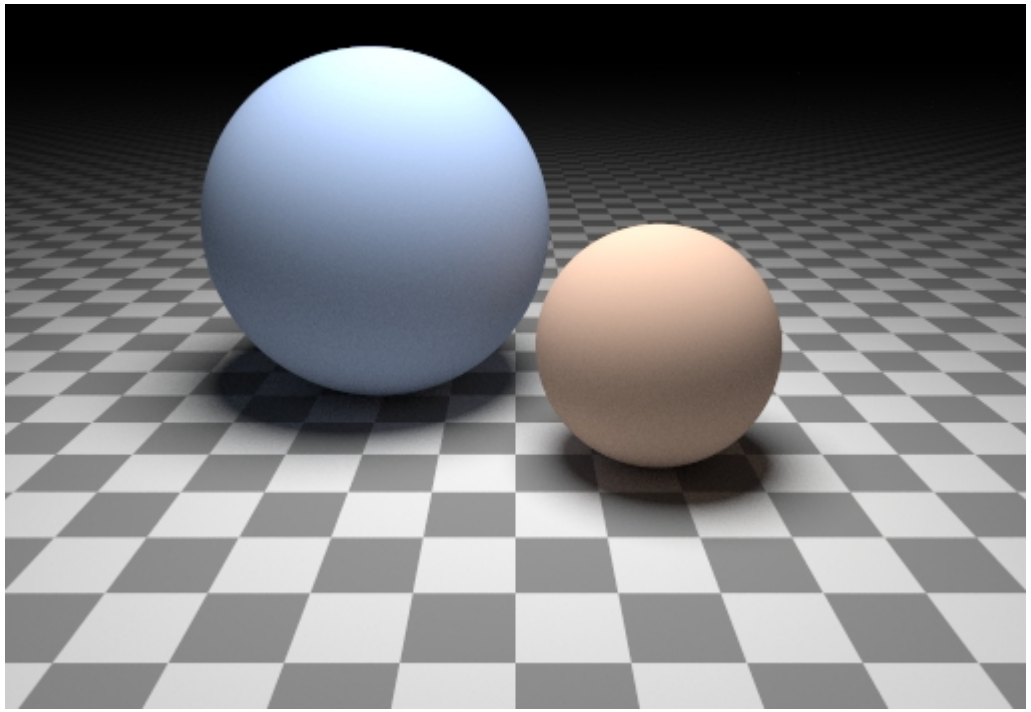
Surface	Typical albedo
Fresh asphalt	0.04[4]
Open ocean	0.06[5]
Worn asphalt	0.12[4]
Conifer forest (Summer)	0.08,[6] 0.09 to 0.15[7]
Deciduous trees	0.15 to 0.18[7]
Bare soil	0.17[8]
Green grass	0.25[8]
Desert sand	0.40[9]
New concrete	0.55[8]
Ocean ice	0.5–0.7[8]
Fresh snow	0.80–0.90[8]



Objects can have varying albedo and albedo varies with wavelength

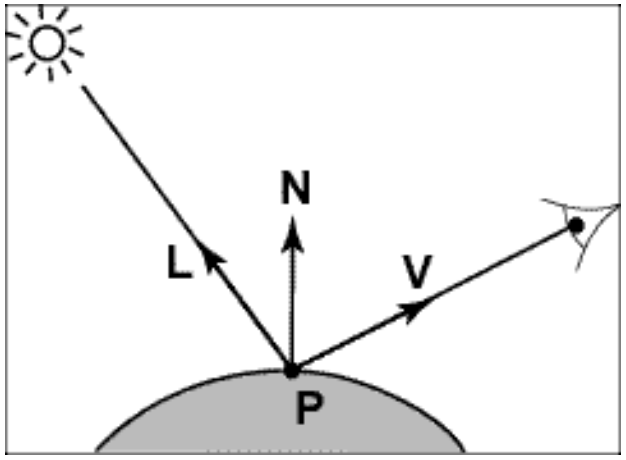
Source:
https://en.wikipedia.org/wiki/Albedo

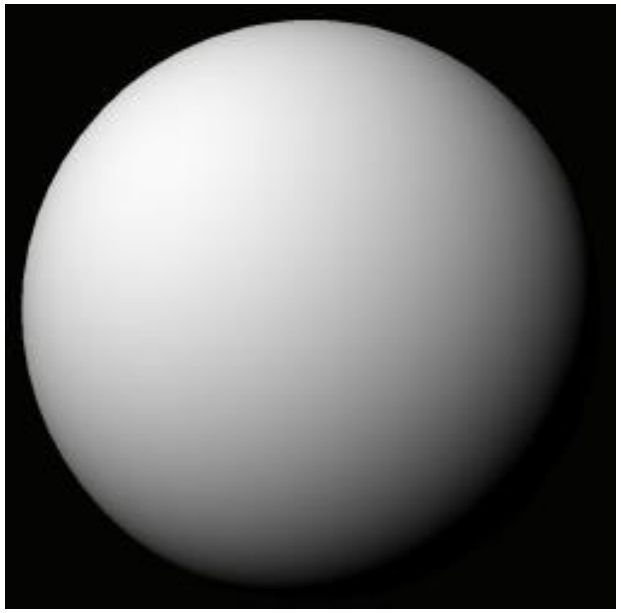
Can we determine shape from lighting?



- Are these spheres?
  - Or just flat discs painted with varying albedo?

Inferring Lambertian Surfaces: Shape from Shading



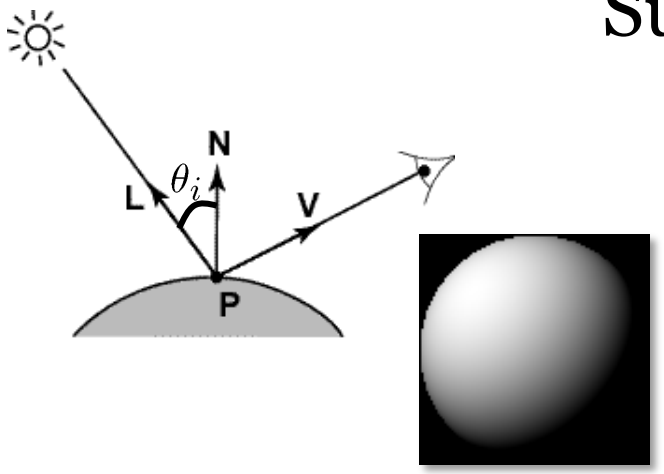


$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

- I : observed image intensity
- k_d : object albedo
- $\mathbf{N}$: surface normal
- $\mathbf{L}$: light source direction

Lambertian sphere with constant albedo lit by a directional light source

Shape from shading



Suppose $k_d = 1$

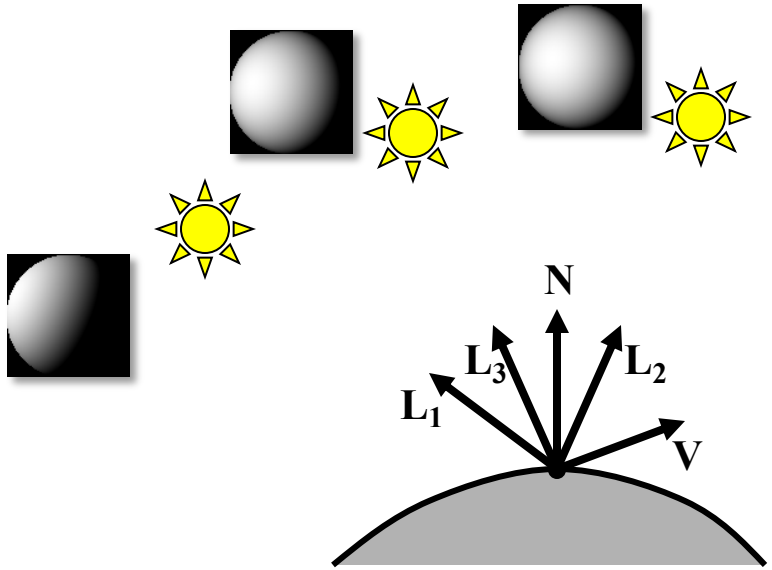
$$
\begin{aligned}
I &= k_d \mathbf{N} \cdot \mathbf{L} \\
&= \mathbf{N} \cdot \mathbf{L} \\
&= cos\ \theta_i
\end{aligned}
$$

You can directly measure angle between normal and light source

- Not quite enough information to compute surface shape
- But can be if you add some additional info, for example
  - assume a few of the normals are known (e.g., along silhouette)
  - constraints on neighboring normals—"integrability"
  - smoothness
- Hard to get it to work well in practice
  - plus, how many real objects have constant albedo?
  - But, deep learning can help

Let’s take more than one photo!

Photometric stereo



$$
\begin{aligned}
I_1 &= k_d \mathbf{N} \cdot \mathbf{L_1} \\
I_2 &= k_d \mathbf{N} \cdot \mathbf{L_2} \\
I_3 &= k_d \mathbf{N} \cdot \mathbf{L_3}
\end{aligned}
$$

Can write this as a matrix equation:

$$
\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = k_d \begin{bmatrix} \mathbf{L_1}^T \\ \mathbf{L_2}^T \\ \mathbf{L_3}^T \end{bmatrix} \mathbf{N}
$$

Solving the equations

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}}_{\mathbf{I}} = \underbrace{\begin{bmatrix} \mathbf{L_1}^T \\ \mathbf{L_2}^T \\ \mathbf{L_3}^T \end{bmatrix}}_{\mathbf{L}} \underbrace{k_d \mathbf{N}}_{\mathbf{G}}$$

3×1 3×3 3×1

$$\mathbf{G} = \mathbf{L}^{-1}\mathbf{I}$$

$$k_d = \|\mathbf{G}\|$$

$$\mathbf{N} = \frac{1}{k_d}\mathbf{G}$$

Solve one such linear system **per pixel** to solve for that pixel's surface normal

More than three lights

Get better results by using more lights

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \mathbf{L_1} \\ \vdots \\ \mathbf{L_n} \end{bmatrix} k_d \mathbf{N}$$

Least squares solution:

$$\begin{aligned} \mathbf{I} &= \mathbf{LG} \\ \mathbf{L^T I} &= \mathbf{L^T L G} \\ \mathbf{G} &= (\mathbf{L^T L})^{-1}(\mathbf{L^T I}) \end{aligned}$$

Solve for N, k_d as before

What's the size of $\mathbf{L^T L}$?

Least Squares Refresher:

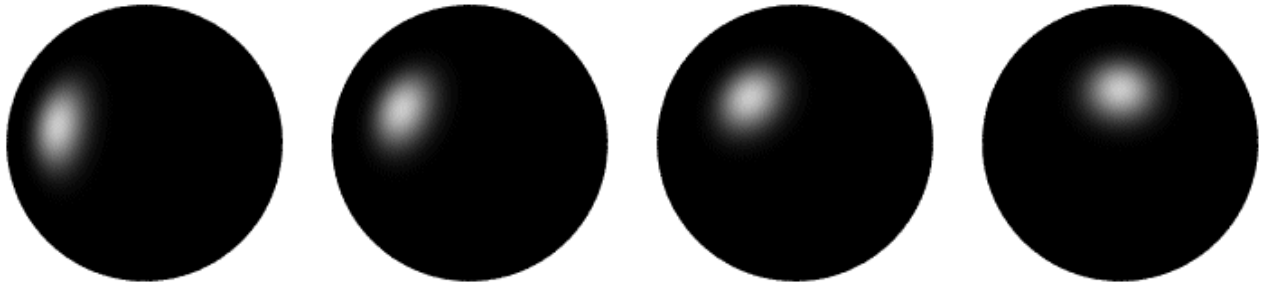
Problem: $\mathbf{Ax} = \mathbf{b}$

Psuedo Inverse: $(\mathbf{A^T A})^{-1}\mathbf{A^T}$

Least Squares Solution: $\mathbf{x} \approx (\mathbf{A^T A})^{-1}\mathbf{A^T b}$

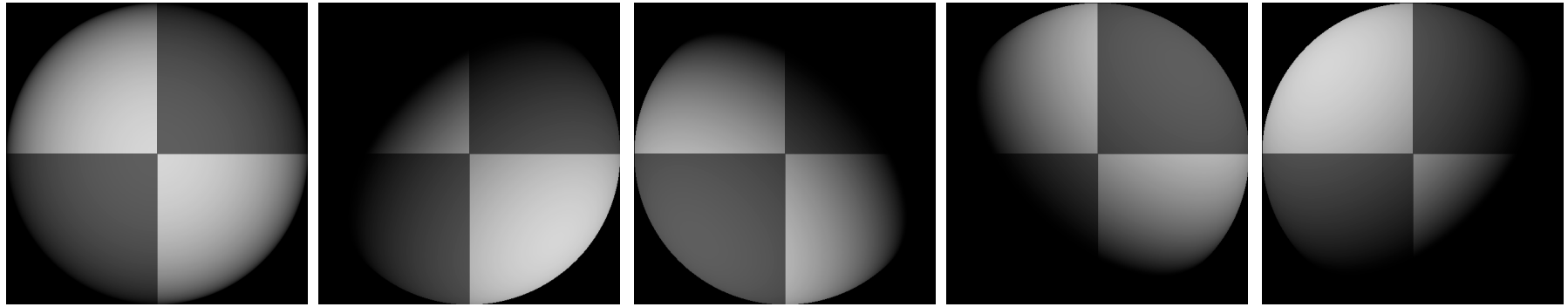
Computing light source directions

Trick: place a chrome sphere in the scene

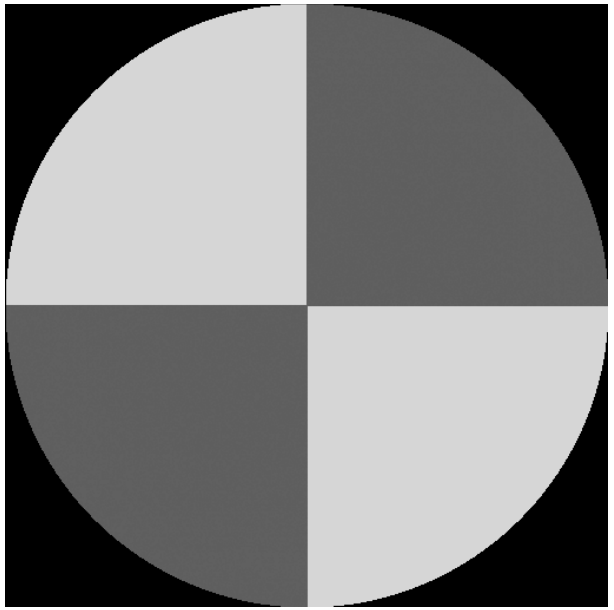


- the location of the highlight tells you where the light source is

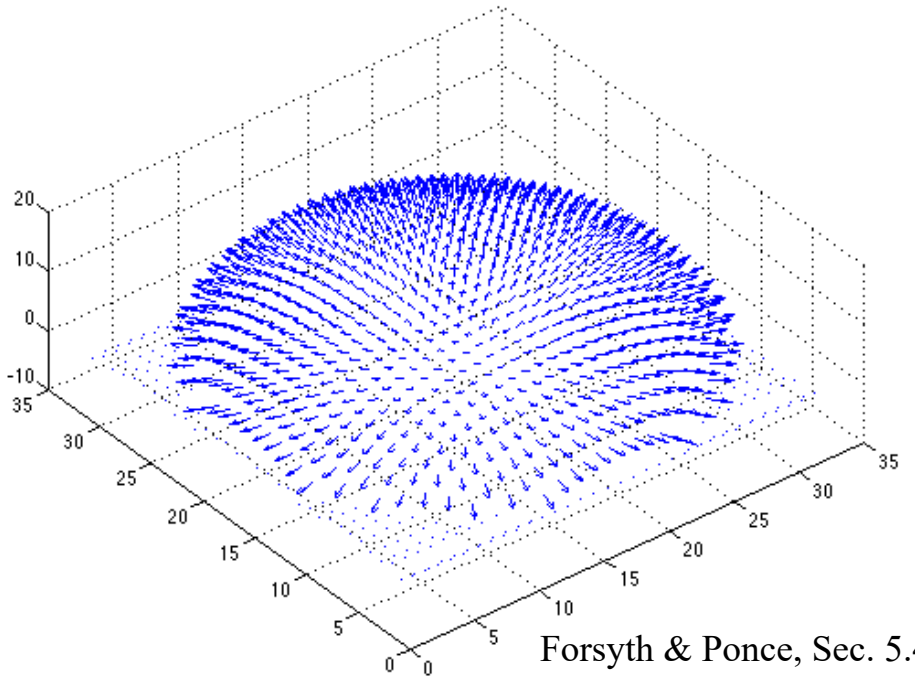
Example



Recovered albedo



Recovered normal field



Forsyth & Ponce, Sec. 5.4

Depth from normals

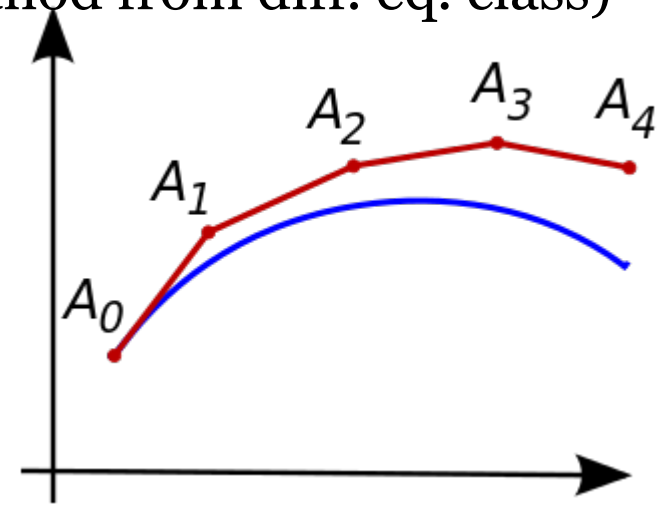
- Solving the linear system per-pixel gives us an estimated surface normal for each pixel

Input photo | Estimated normals | Estimated normals (needle diagram)

- How can we compute depth from normals?
  - Normals are like the “derivative” of the true depth

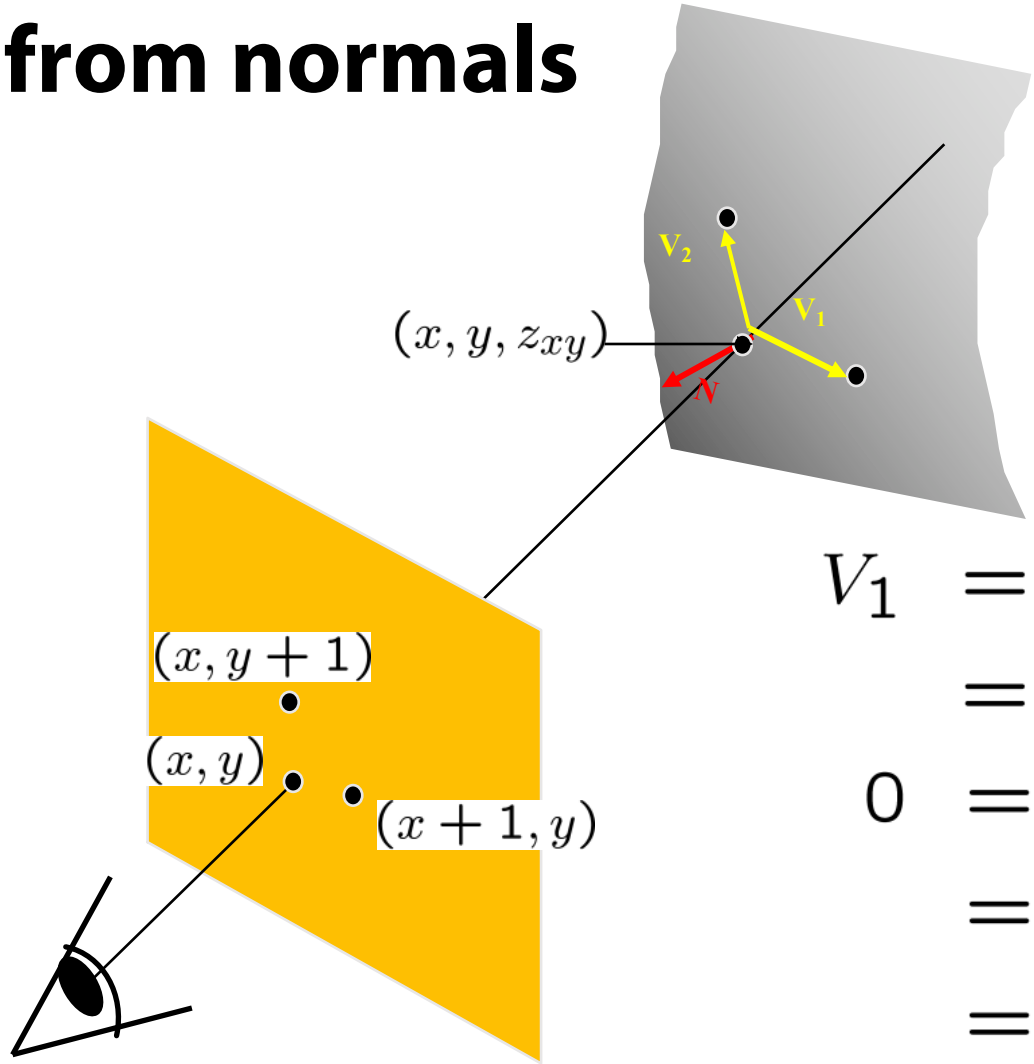
Normal Integration

- Integrating a set of derivatives is easy in 1D
  - (similar to Euler's method from diff. eq. class)



- Could just integrate normals in each column / row separately
- Instead, we formulate as a linear system and solve for depths that *best agree with the surface normals*

Depth from normals

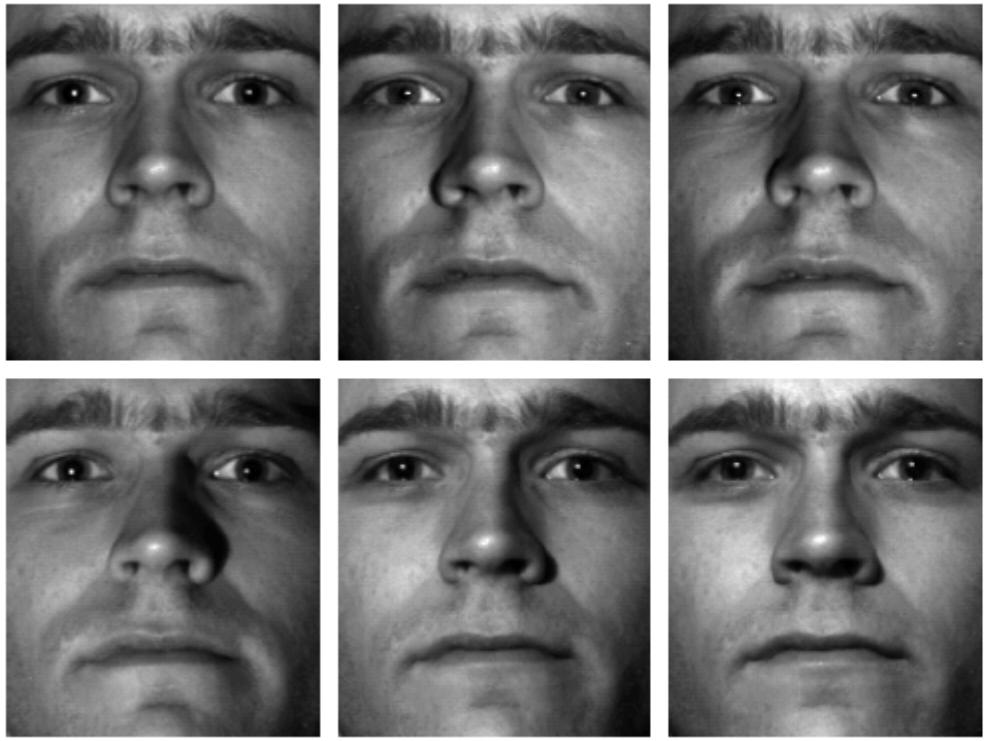


$$
\begin{aligned}
V_1 &= (x+1, y, z_{x+1,y}) - (x, y, z_{xy}) \\
&= (1, 0, z_{x+1,y} - z_{xy}) \\
0 &= N \cdot V_1 \\
&= (n_x, n_y, n_z) \cdot (1, 0, z_{x+1,y} - z_{xy}) \\
&= n_x + n_z(z_{x+1,y} - z_{xy})
\end{aligned}
$$

Get a similar equation for $\mathbf{V_2}$

- Each normal gives us two linear constraints on z
- compute z values by solving a matrix equation

Results

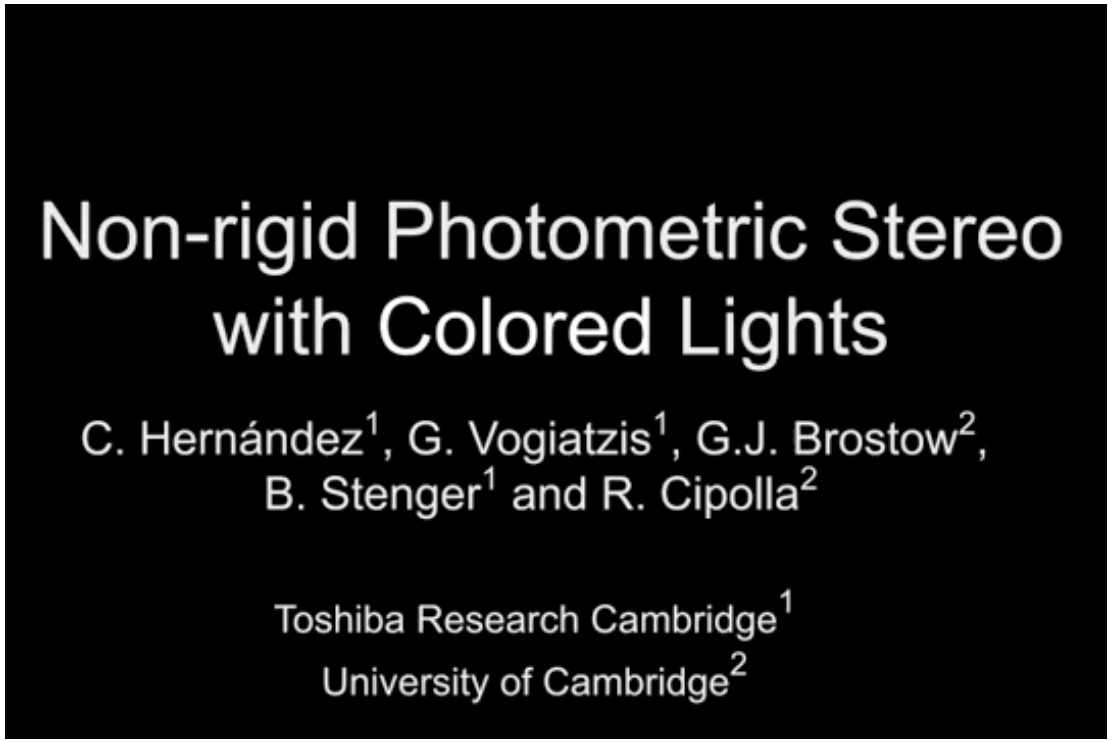


from Athos Georghiades

Results

Extension

- Photometric Stereo from Colored Lighting



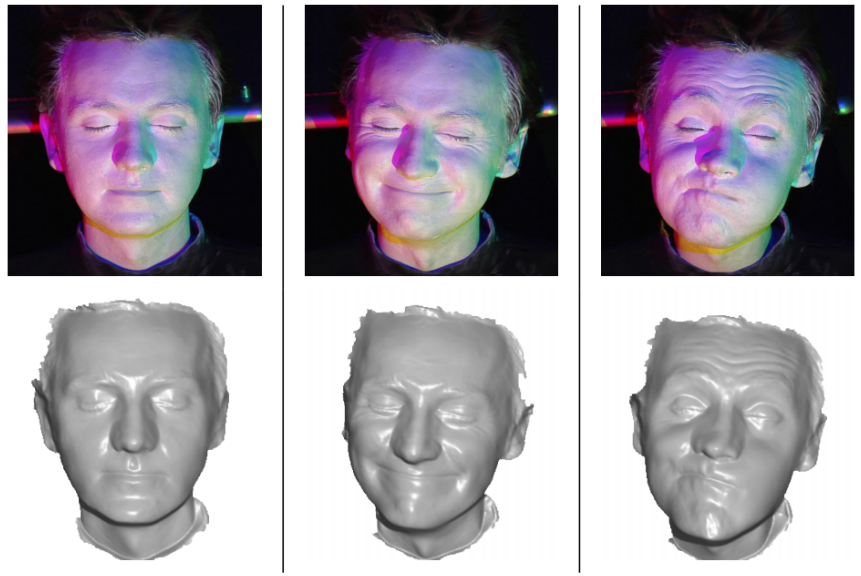


Fig. 2. **Applying the original algorithm to a face with white makeup.** Top: example input frames from video of an actor smiling and grimacing. Bottom: the resulting integrated surfaces.

Video Normals from Colored Lights
Gabriel J. Brostow, Carlos Hernández, George Vogiatzis, Björn Stenger, Roberto Cipolla
IEEE TPAMI, Vol. 33, No. 10, pages 2104-2114, October 2011.

Questions?

For now, ignore specular reflection

Slides from Photometric Methods for 3D Modeling, Matsushita, Wilburn, Ben-Ezra

And Refraction…

Slides from Photometric Methods for 3D Modeling, Matsushita, Wilburn, Ben-Ezra

And Interreflections…

Slides from Photometric Methods for 3D Modeling, Matsushita, Wilburn, Ben-Ezra

And Subsurface Scattering…

Slides from Photometric Methods for 3D Modeling, Matsushita, Wilburn, Ben-Ezra

Limitations

Bigger problems

- doesn't work for shiny things, semi-translucent things
- shadows, inter-reflections

Smaller problems

- camera and lights have to be distant
- calibration requirements
  - measure light source directions, intensities
  - camera response function

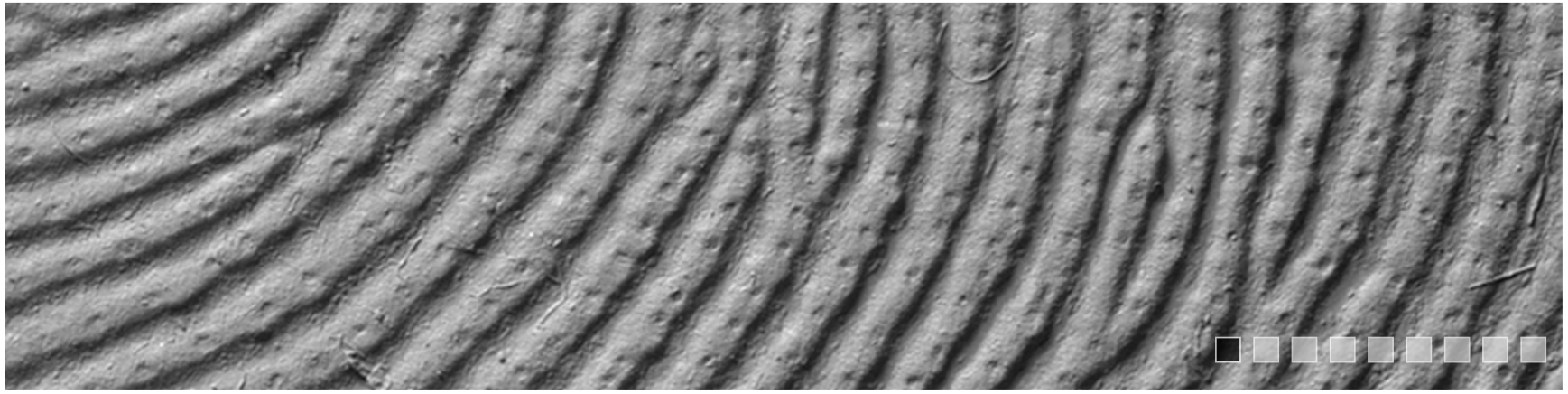
Newer work addresses some of these issues

Some pointers for further reading:

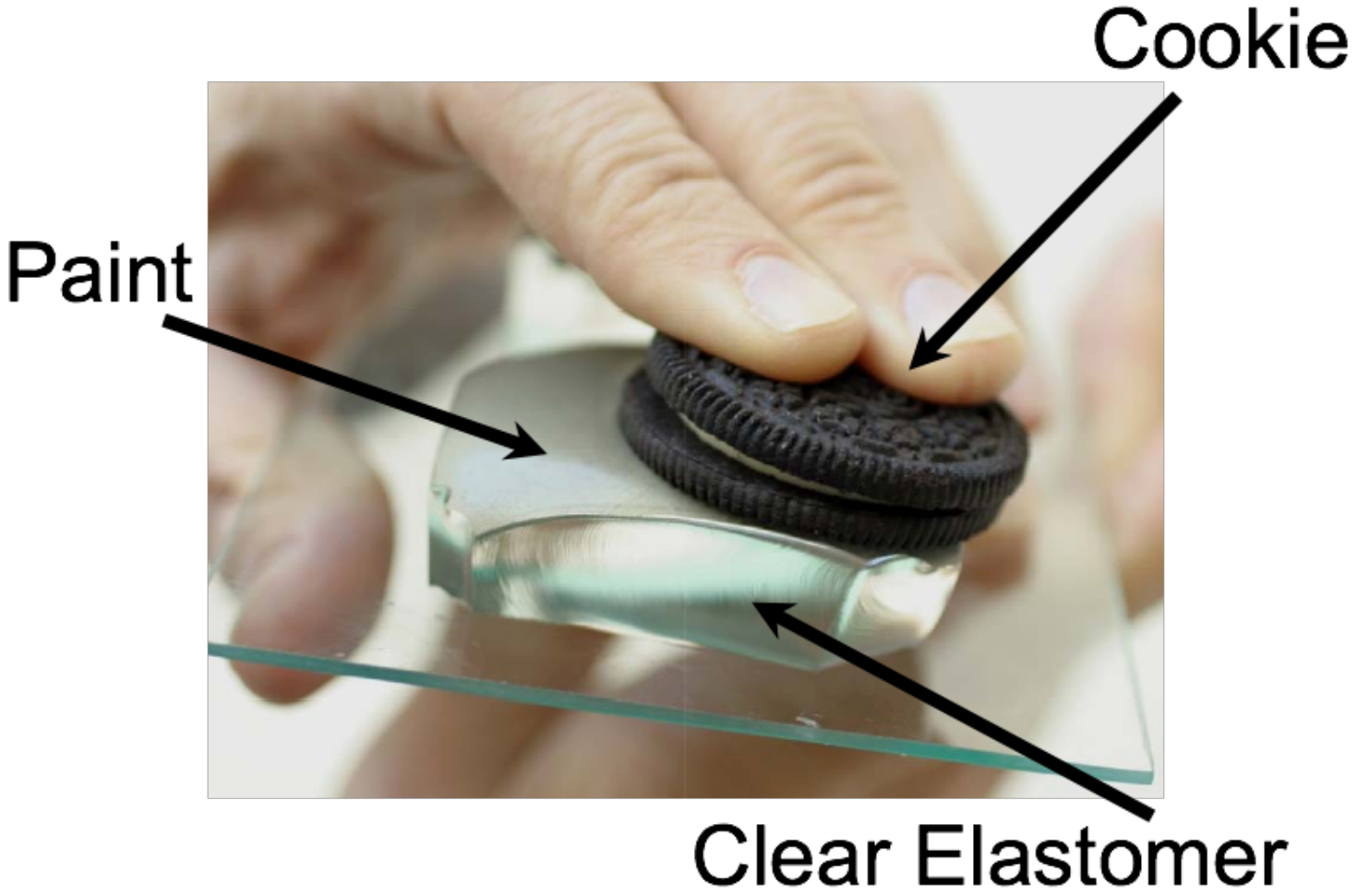
- Zickler, Belhumeur, and Kriegman, "*Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction*." IJCV, Vol. 49 No. 2/3, pp 215-227.
- Hertzmann & Seitz, "*Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs*." IEEE Trans. PAMI 2005

GELSIGHT

HOME PRODUCTS VIDEOS IMAGES PAPERS NEWS ABOUT US CONTACT



Johnson and Adelson, 2009

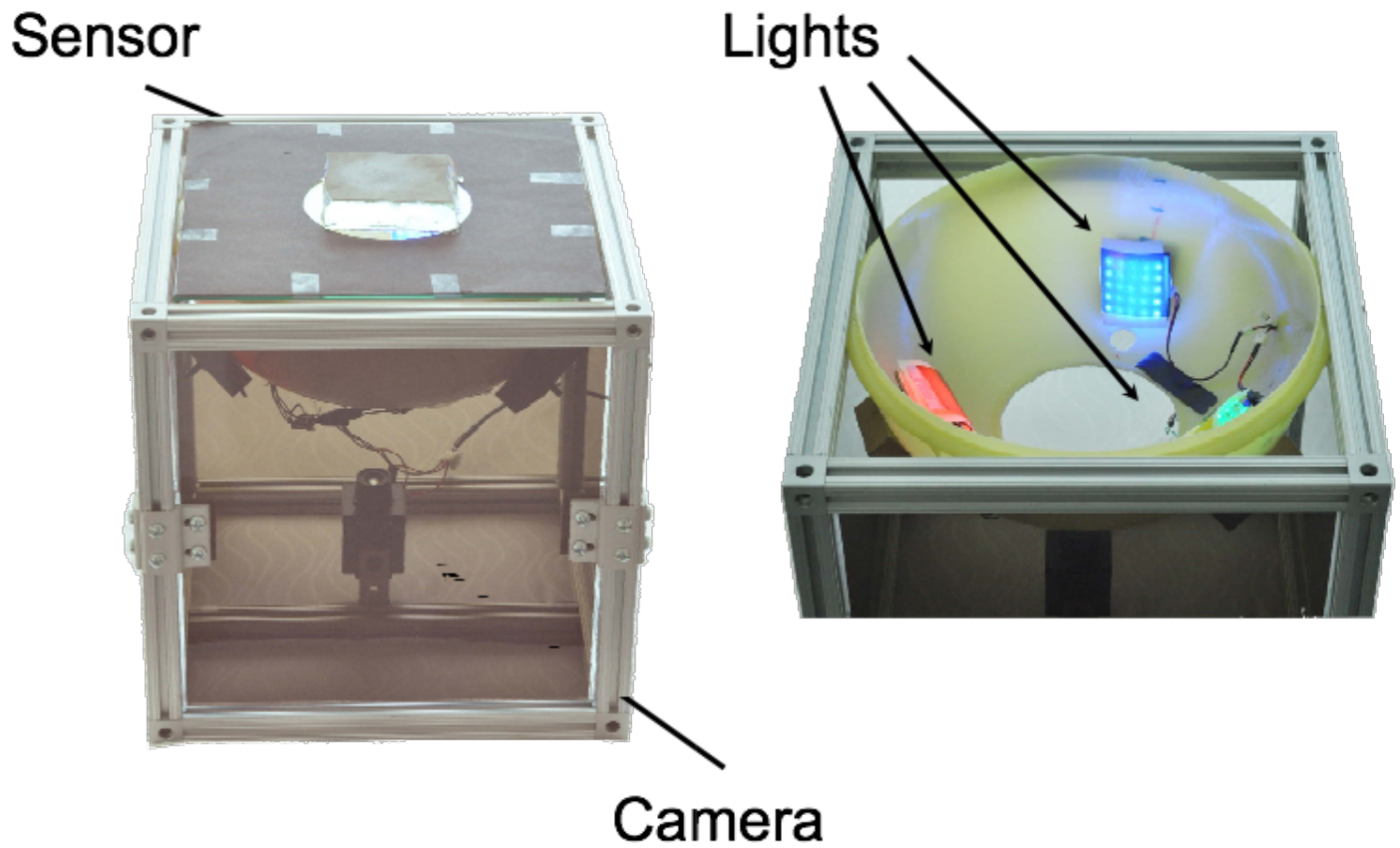


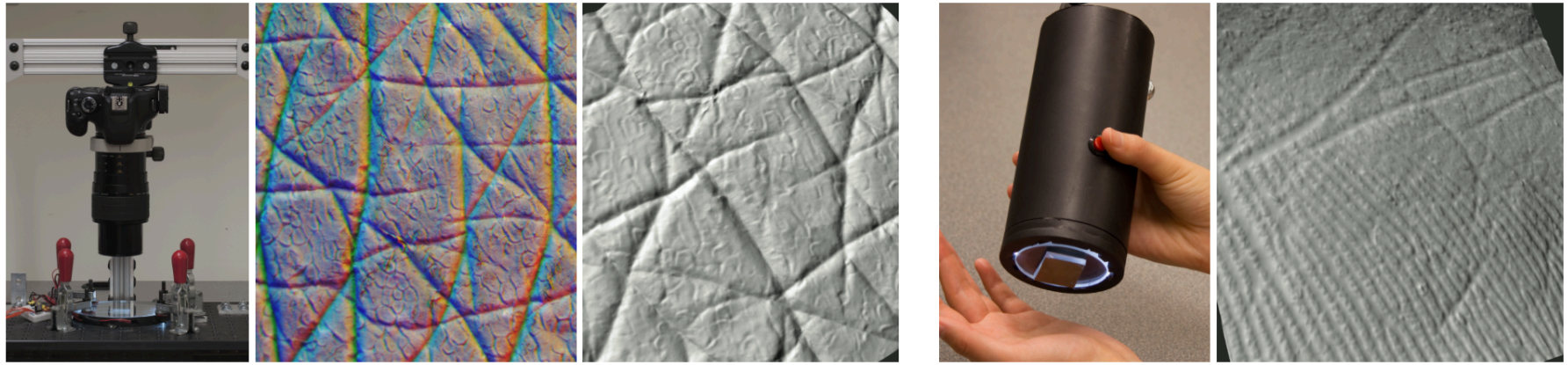
Johnson and Adelson, 2009


OREO


OREO

Lights, camera, action





(a) bench configuration (b) captured (c) reconstruction (d) portable configuration (e) reconstruction

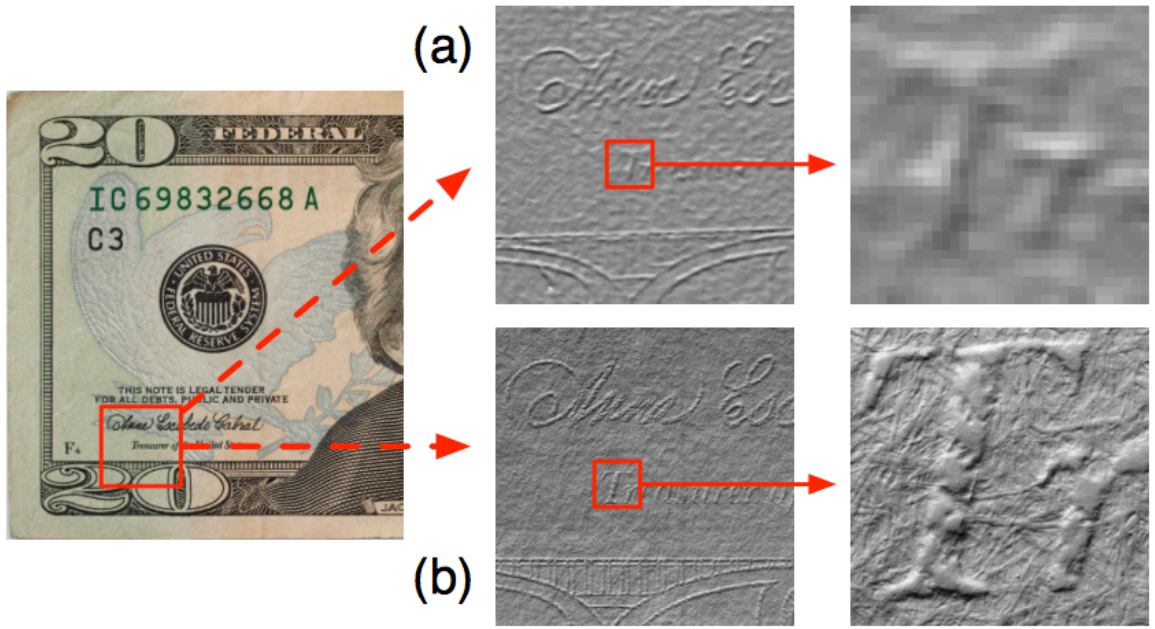


Figure 7: *Comparison with the high-resolution result from the original retrographic sensor. (a) Rendering of the high-resolution $20 bill example from the original retrographic sensor with a close-up view. (b) Rendering of the captured geometry using our method.*

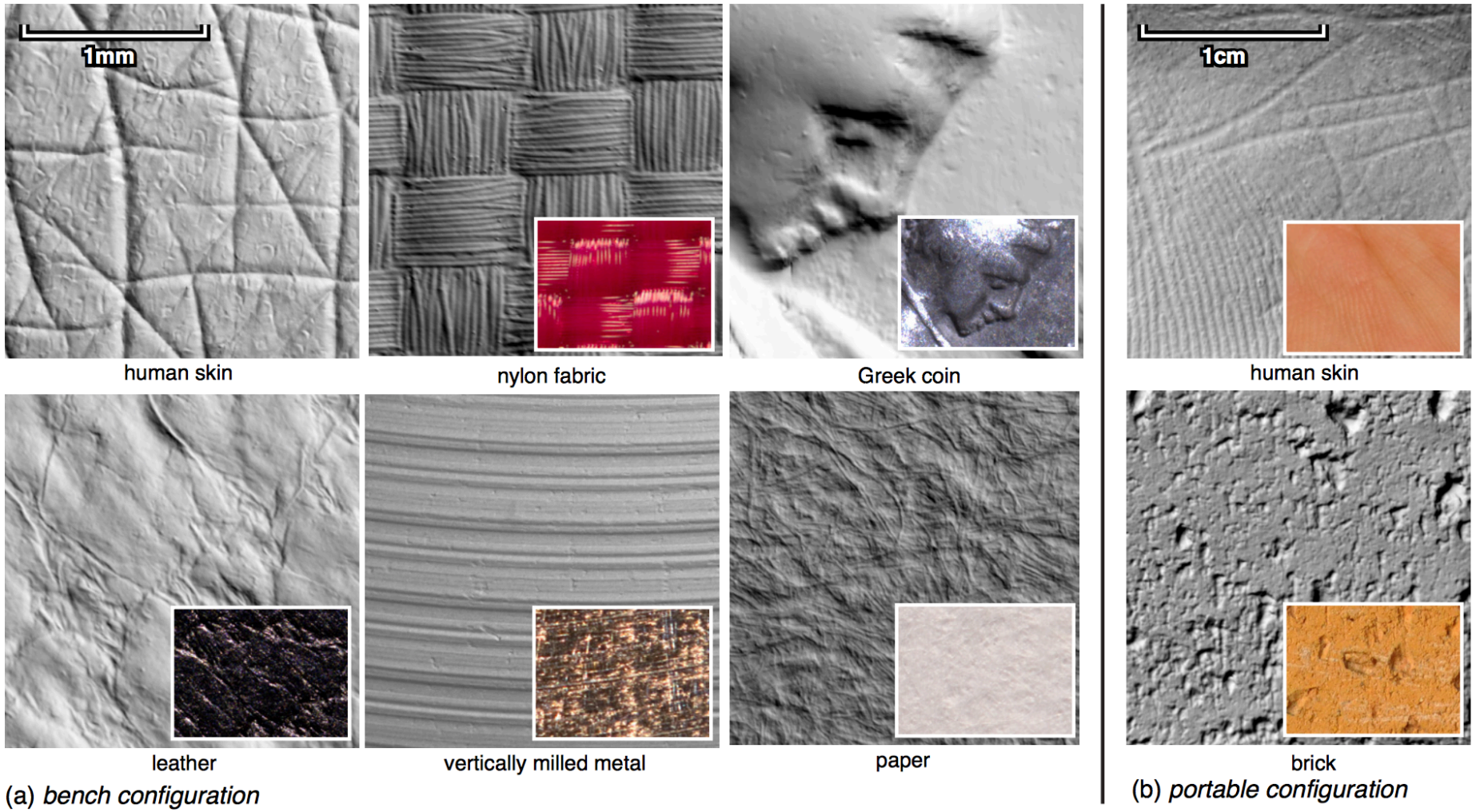
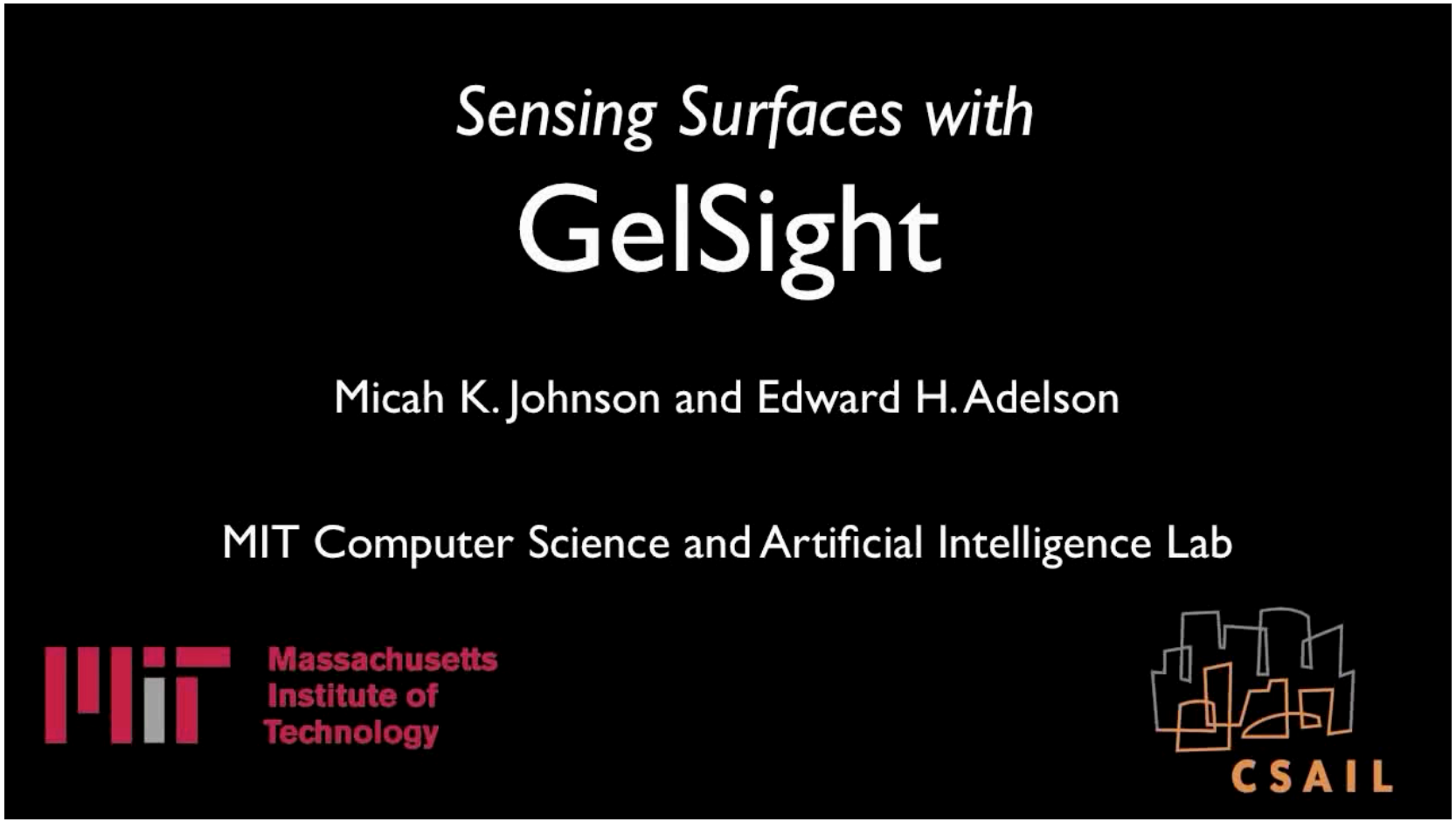


Figure 9: *Example geometry measured with the bench and portable configurations. Outer image: rendering under direct lighting. Inset: macro photograph of original sample. Scale shown in upper left. Color images are shown for context and are to similar, but not exact scale.*



Sensing Surfaces with GelSight

kimoatmit

Subscribe 64

138,850 views

https://www.youtube.com/watch?v=S7gXih4XS7A

InverseRenderNet: Learning single image inverse rendering

Ye Yu and William A. P. Smith
Department of Computer Science, University of York, UK
{yy1571,william.smith}@york.ac.uk

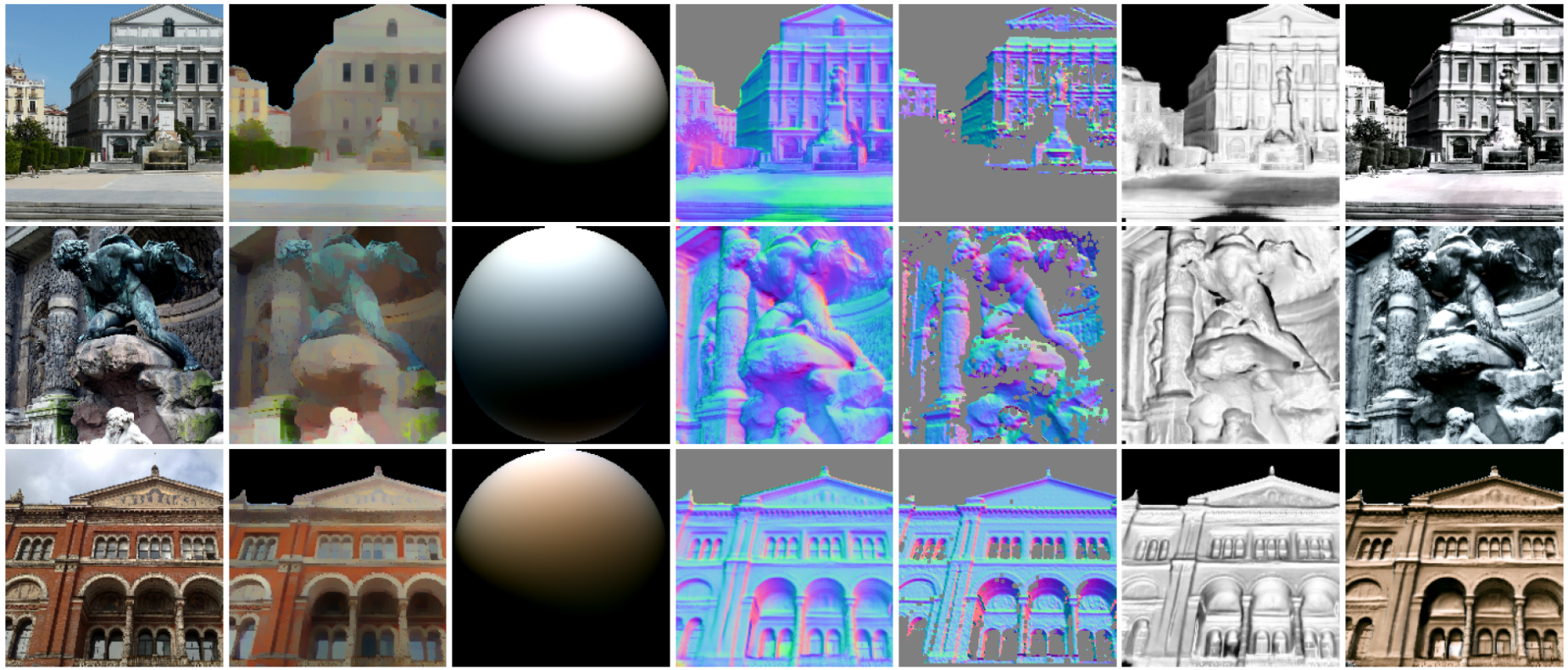


Figure 1: From a single image (col. 1), we estimate albedo and normal maps and illumination (col. 2-4); comparison multi-view stereo result from several hundred images (col. 5); re-rendering of our shape with frontal/estimated lighting (col. 6-7).

Questions?