

Computational Illumination

Computational Photography

CSE 273

Lecture 9

Announcements

- Assignment 4 is due today, 11:59 PM
- Initial project proposal is due Feb 18
- Revised project proposal is due Feb 25
- Initial project presentation slides are due Feb 25
- Draft project report is due Mar 10
- Final project report is due Mar 17
- Final project presentation slides are due Mar 20

Computational illumination

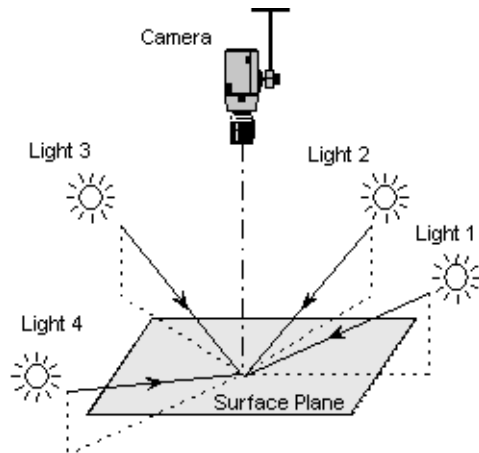
- Controlling photographic illumination in a structured manner, then processing the captured images to create new images

Computational illumination

- Example methods
 - Photometric stereo
 - Structured illumination
 - Time of flight

Computational illumination

- Photometric stereo



surface
(albedo texture map)

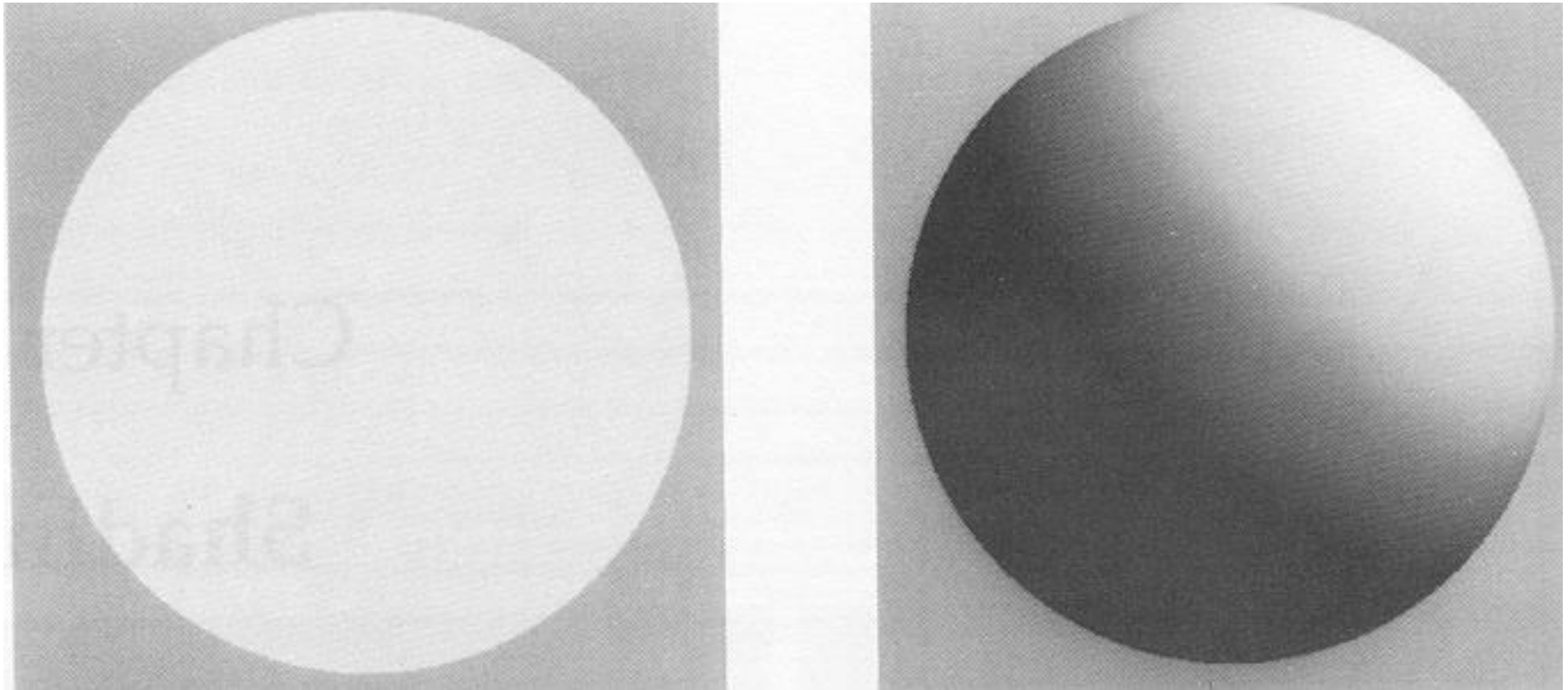


albedo

+ surface normals

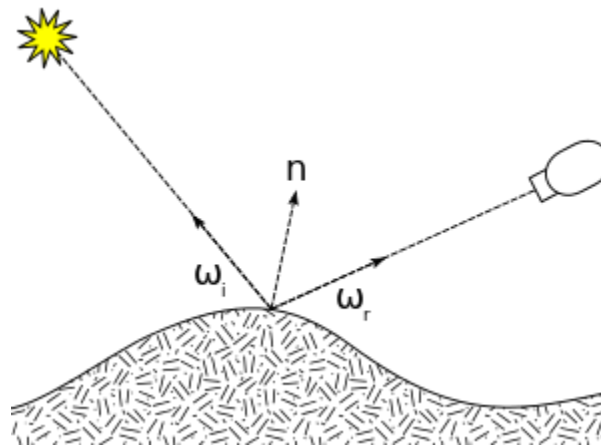


Shading reveals 3D surface geometry

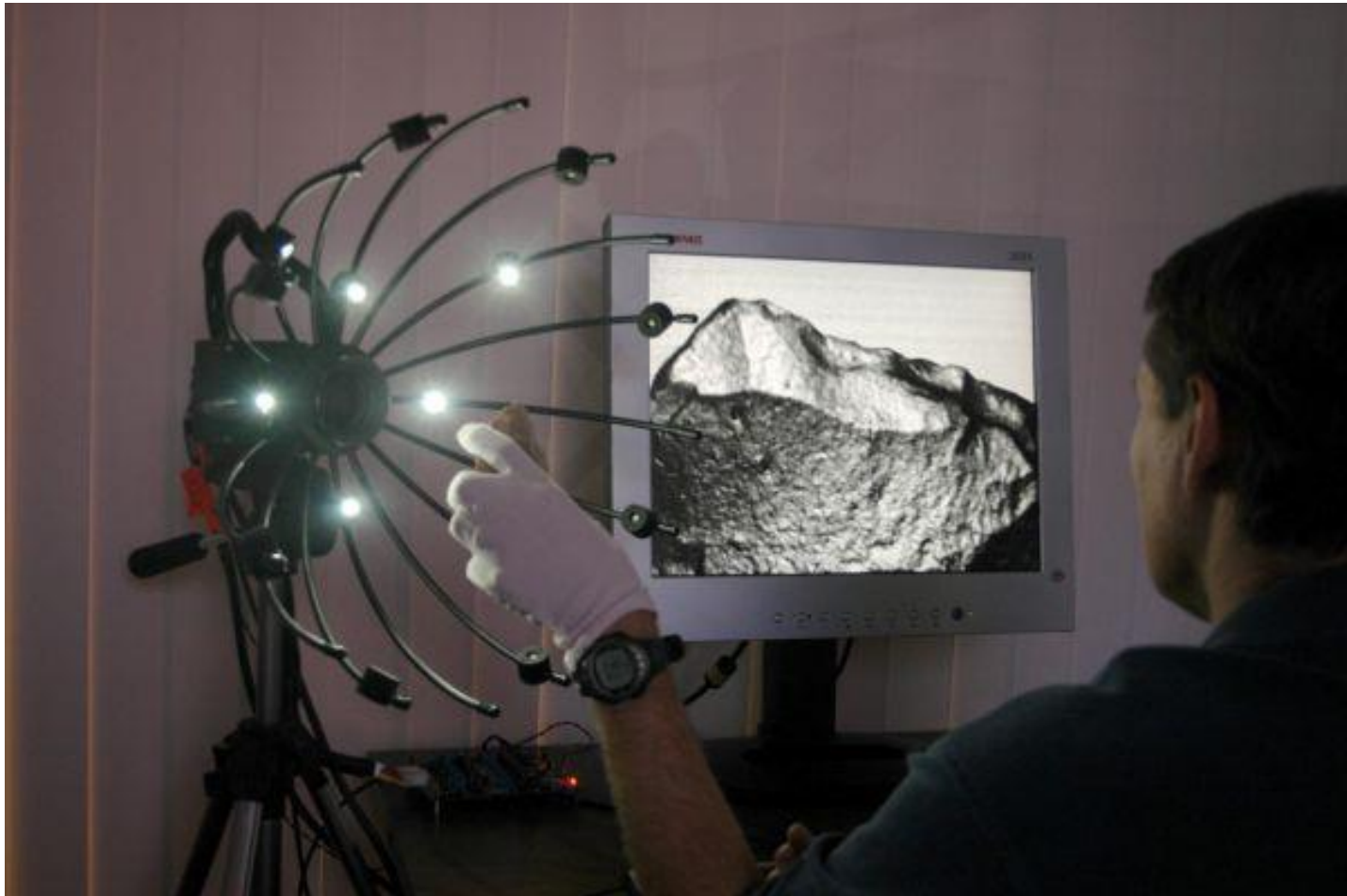


Two shape from X methods that use shading

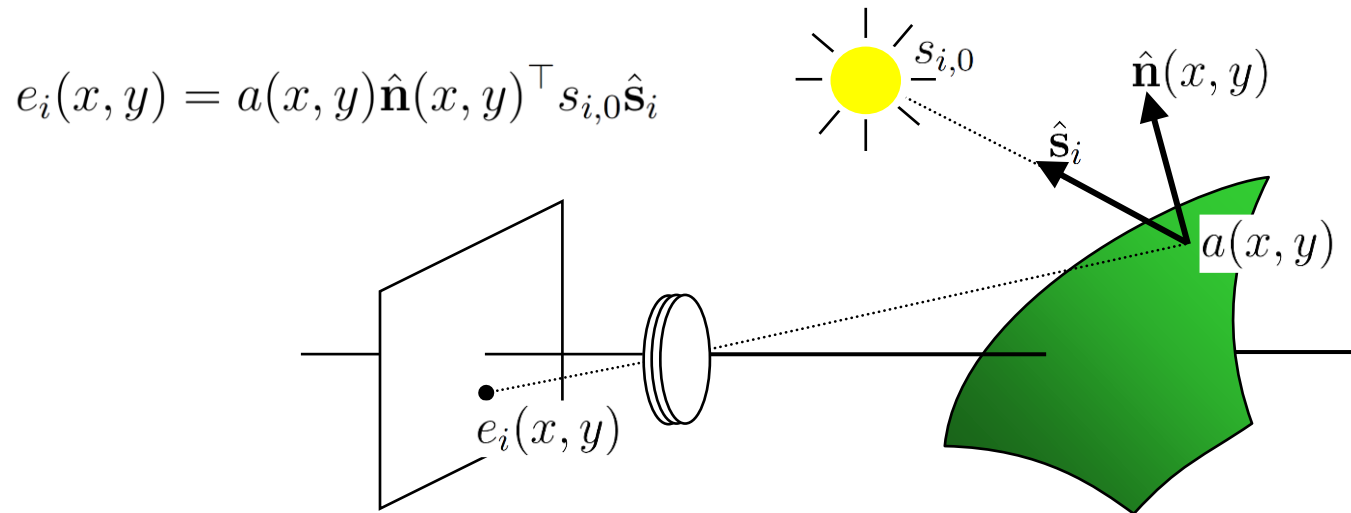
- Shape from shading
 - Use just one image to recover shape. Requires knowledge of light source direction and BRDF everywhere. Too restrictive to be useful.
- Photometric stereo
 - Single viewpoint, multiple images under different lighting.



Photometric stereo rig



Photometric stereo, Lambertian surface



$s_{i,0}$ is the intensity of the i -th light source

$\hat{s}_i$ is the i -th light source direction at the surface point

$\hat{n}(x, y)$ is the normal at the surface point projected to the image coordinates (x, y)

$a(x, y)$ is the albedo of the surface point projected to the image coordinates (x, y)

$e_i(x, y)$ is the intensity of the light reflected from the surface point projected to the i -th image coordinates (x, y)

Photometric stereo, Lambertian surface

For each pixel

$$e_i(x, y) = a(x, y) \hat{\mathbf{n}}(x, y)^\top s_{i,0} \hat{\mathbf{s}}_i, \text{ solve for } a(x, y) \text{ and } \hat{\mathbf{n}}(x, y)$$

$$e_i(x, y) = \mathbf{b}(x, y)^\top \mathbf{s}_i, \text{ where } \mathbf{b}(x, y) = a(x, y) \hat{\mathbf{n}}(x, y) \text{ and } \mathbf{s}_i = s_{i,0} \hat{\mathbf{s}}_i$$

$$e_i(x, y) = \mathbf{s}_i^\top \mathbf{b}(x, y)$$

$$\begin{bmatrix} e_1(x, y) \\ e_2(x, y) \\ \vdots \\ e_n(x, y) \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1^\top \\ \mathbf{s}_2^\top \\ \vdots \\ \mathbf{s}_n^\top \end{bmatrix} \mathbf{b}(x, y)$$

$n \geq 3$ light sources and associated images

$$\mathbf{e}(x, y) = \mathbf{S} \mathbf{b}(x, y), \text{ solve for } \mathbf{b}(x, y)$$

$$\mathbf{S}^\top \mathbf{e}(x, y) = \mathbf{S}^\top \mathbf{S} \mathbf{b}(x, y)$$

$$(\mathbf{S}^\top \mathbf{S})^{-1} \mathbf{S}^\top \mathbf{e}(x, y) = \mathbf{b}(x, y)$$

$$\mathbf{S}^+ \mathbf{e}(x, y) = \mathbf{b}(x, y), \text{ where } \mathbf{S}^+ = (\mathbf{S}^\top \mathbf{S})^{-1} \mathbf{S}^\top$$

Same for all
pixel coordinates

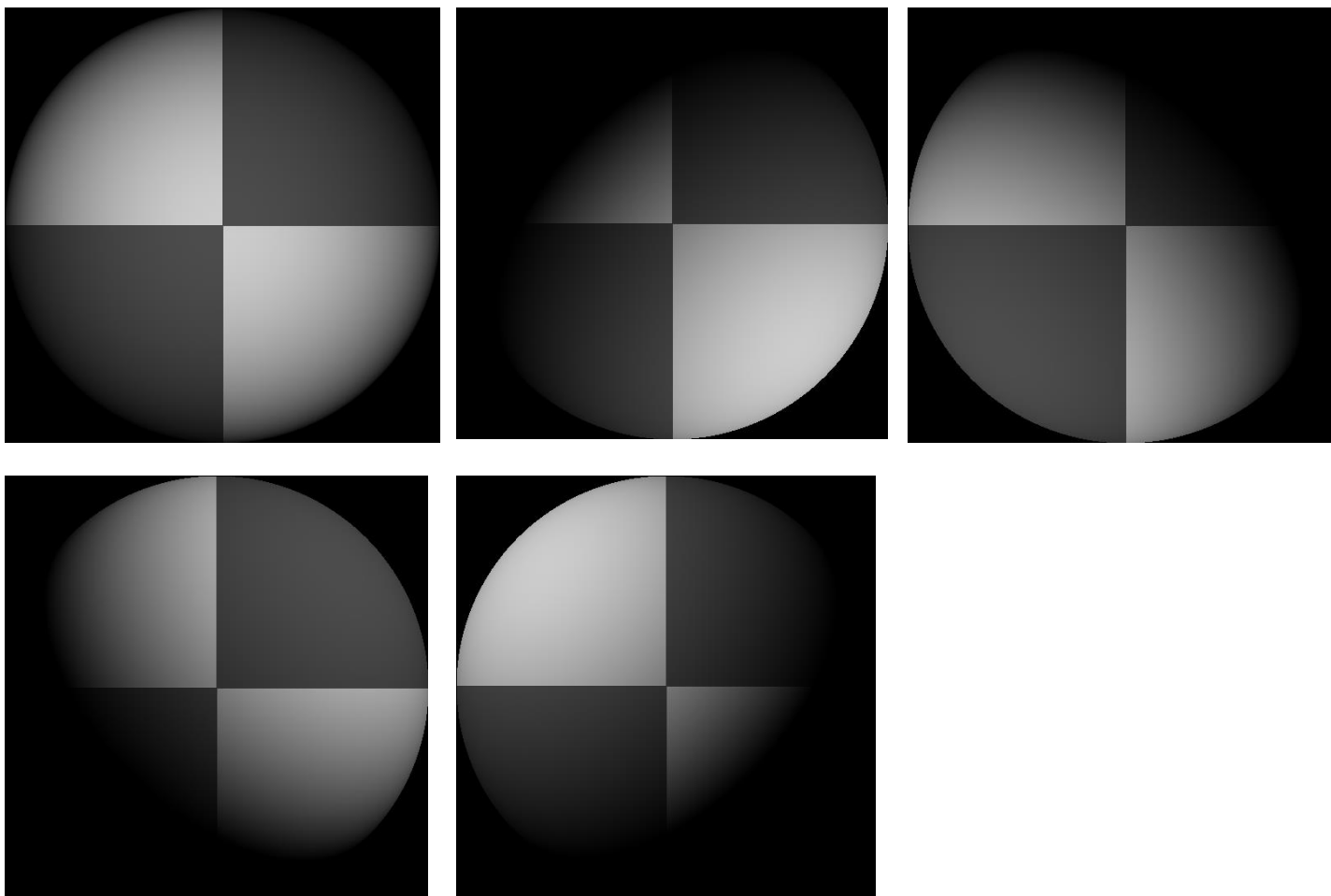
$$\mathbf{b}(x, y) = a(x, y) \hat{\mathbf{n}}(x, y), \text{ where } a(x, y) = \|\mathbf{b}(x, y)\| \text{ and } \hat{\mathbf{n}}(x, y) = \frac{\mathbf{b}(x, y)}{\|\mathbf{b}(x, y)\|}$$

albedo

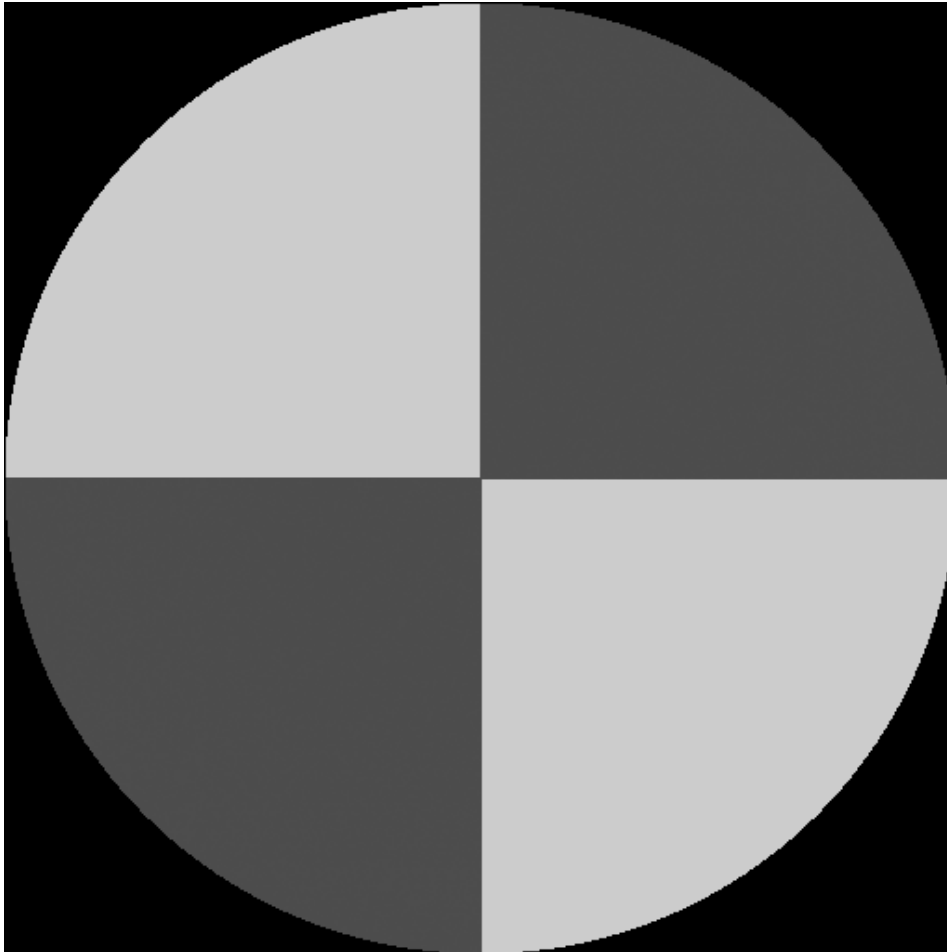


+ surface normals

Example: input images

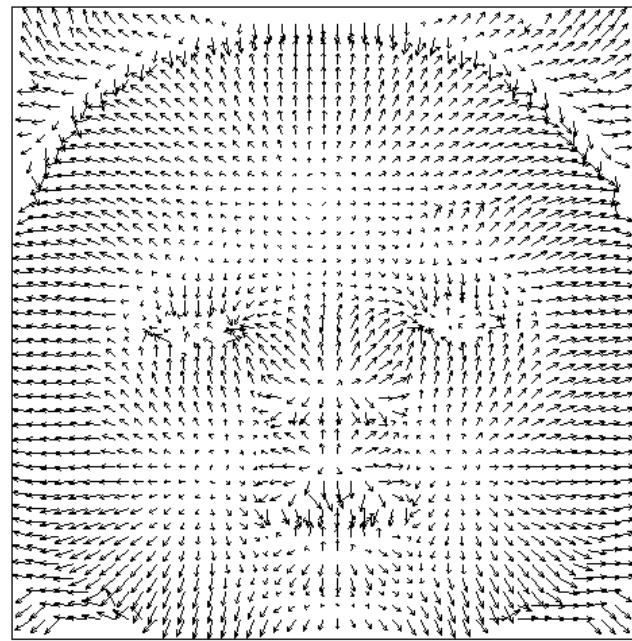


Example: albedo



Photometric stereo, color images

- Lambertian surface
 - Use multichannel photometric stereo algorithm (preferred) or convert to grayscale and use single channel algorithm



Application: image-based relighting

- Rendering under different illumination



Given images and light source directions

Rendered
image

$$e_i(x, y) = a(x, y) \hat{\mathbf{n}}(x, y)^\top s_{i,0} \hat{\mathbf{s}}_i$$

Photometric stereo, color images

- Non-Lambertian surface

Given color of light source $\mathbf{c} = (R_s, G_s, B_s)^\top$, calculate $\mathbf{R} \in \text{SO}(3)$ such that $(1, 0, 0)^\top = \mathbf{R}\hat{\mathbf{c}}$

$$\hat{\mathbf{c}} = \frac{\mathbf{c}}{\|\mathbf{c}\|}$$

$$\mathbf{R} = \begin{bmatrix} \hat{\mathbf{c}}^\top \\ [\hat{\mathbf{c}}]^\perp \end{bmatrix}, \text{ where } [\hat{\mathbf{c}}]^\perp \hat{\mathbf{c}} = 0 \text{ (i.e., } [\hat{\mathbf{c}}]^\perp \text{ is left null space of } \hat{\mathbf{c}})$$

if $\det(\mathbf{R}) = -1$, then negate last row of $\mathbf{R}$

- Use $\mathbf{R}$ to map RGB to SUV color space, where S is specular and U and V are diffuse
- Use U and V as input to Lambertian surface photometric stereo algorithm

Left null space of a vector

Let $\mathbf{x} = (x_1, x_2, \dots, x_n)^\top$ be any vector, let $\mathbf{e}_1 = (1, 0, \dots, 0)^\top$, and let $\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\| \mathbf{e}_1$. Calculate the Householder matrix $\mathbf{H}_{\mathbf{v}}$ such that $\mathbf{H}_{\mathbf{v}} \mathbf{x} = \mp \|\mathbf{x}\| \mathbf{e}_1 = (\mp \|\mathbf{x}\|, 0, \dots, 0)^\top$.

$$\mathbf{H}_{\mathbf{v}} = \mathbf{I} - 2 \frac{\mathbf{v} \mathbf{v}^\top}{\mathbf{v}^\top \mathbf{v}}$$

Note that $\mathbf{H}_{\mathbf{v}}$ is symmetric and orthogonal. $\mathbf{v}$ is called the Householder vector for $\mathbf{x}$. For stability reasons, the sign ambiguity in defining $\mathbf{v}$ should be resolved by setting

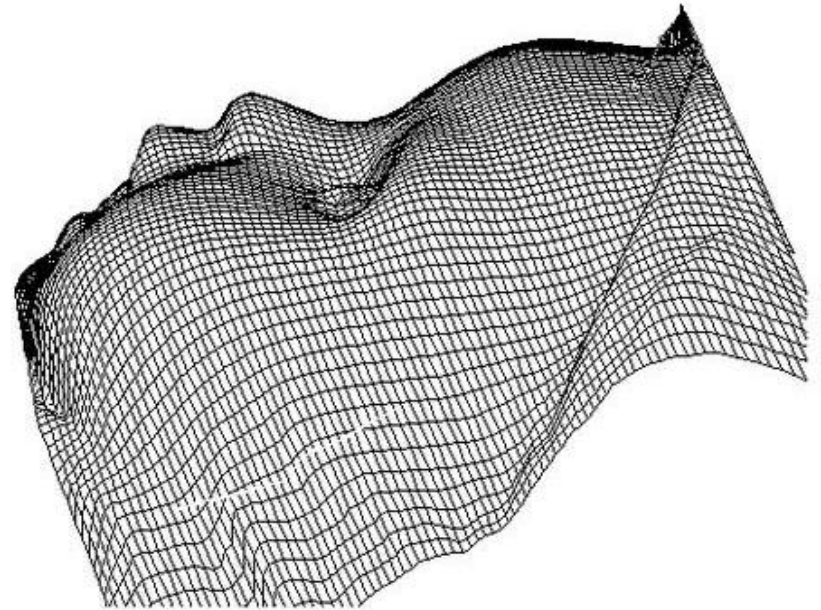
$$\mathbf{v} = \mathbf{x} + \text{sign}(x_1) \|\mathbf{x}\| \mathbf{e}_1$$

In $[\mathbf{x}]^\perp \mathbf{x} = \mathbf{0}$, $[\mathbf{x}]^\perp$ is the left null space of $\mathbf{x}$ and may be calculated from $\mathbf{H}_{\mathbf{v}}$ as follows.

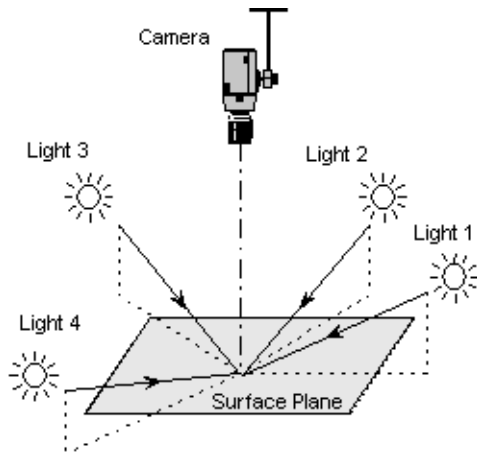
$$\mathbf{H}_{\mathbf{v}} = \begin{bmatrix} \mathbf{h}_{\mathbf{v}}^{1\top} \\ [\mathbf{x}]^\perp \end{bmatrix}$$

(i.e., $[\mathbf{x}]^\perp$ is $\mathbf{H}_{\mathbf{v}}$ with the first row omitted). Further, note that $([\mathbf{x}]^\perp)^\top$ is the (right) null space of $\mathbf{x}^\top$ (i.e., $\mathbf{x}^\top ([\mathbf{x}]^\perp)^\top = \mathbf{0}^\top$).

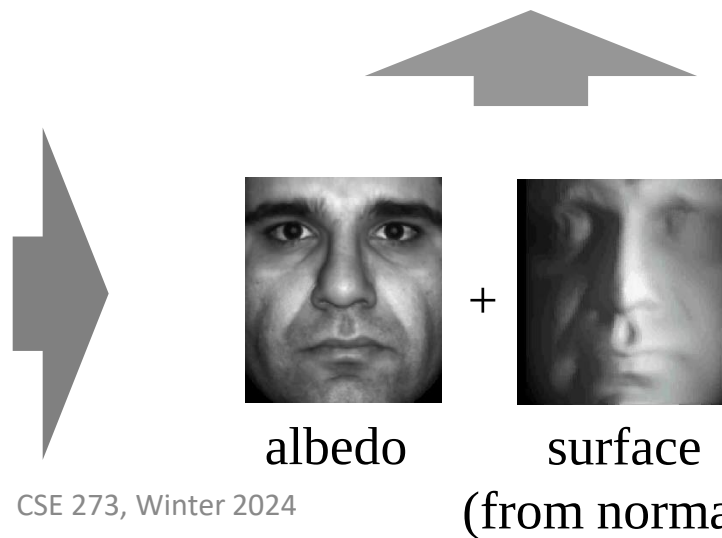
Application: 3D surface from normals



Application: 3D reconstruction



3D surface
(albedo texture map)

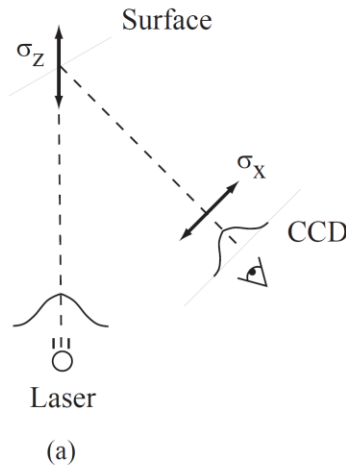


Computational illumination

- Structured illumination
 - The projection of a known pattern onto a scene. The scene is reconstructed from one or more images of the scene and deformed pattern.

Structured illumination

Projection of point



Projection of line

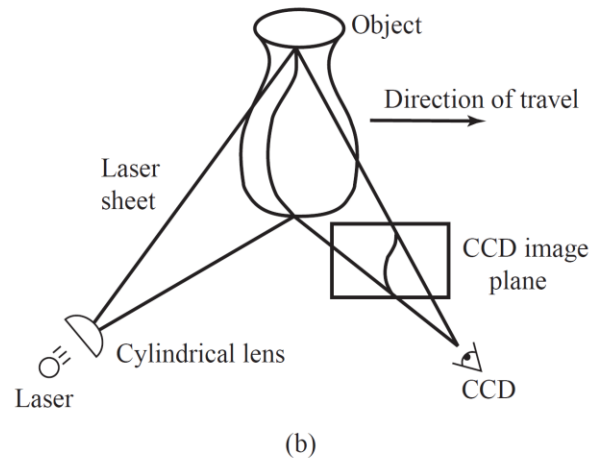


Figure 12.5 Range data scanning (Curless and Levoy 1996) © 1996 ACM: (a) a laser dot on a surface is imaged by a CCD sensor; (b) a laser stripe (sheet) is imaged by the sensor (the deformation of the stripe encodes the distance to the object); (c) the resulting set of 3D points are turned into (d) a triangulated mesh.

Structured illumination

- Calibration
 - Mathematical model for projector
 - Mathematical model for camera
 - Use basic pinhole model with lens distortion for both
- 3D scene reconstruction
 - Triangulation

Triangulation

- Linear estimation of 3D point using the direct linear transformation (DLT) algorithm

Given $n \geq 2$ point correspondences over n views where $\mathbf{X}$ is visible

$\mathbf{x}^{(j)} = \mathbf{P}^{(j)} \mathbf{X} \forall j$ where $\mathbf{X}$ is visible, solve for $\mathbf{X}$

$$[\mathbf{x}^{(j)}]^\perp \mathbf{P}^{(j)} \mathbf{X} = \mathbf{0} \forall j$$

$$\begin{bmatrix} [\mathbf{x}^{(1)}]^\perp \mathbf{P}^{(1)} \\ \vdots \\ [\mathbf{x}^{(n)}]^\perp \mathbf{P}^{(n)} \end{bmatrix} \mathbf{X} = \mathbf{0}$$

$$\mathbf{A} \mathbf{X} = \mathbf{0}, \text{ where } \mathbf{A} = \begin{bmatrix} [\mathbf{x}^{(1)}]^\perp \mathbf{P}^{(1)} \\ \vdots \\ [\mathbf{x}^{(n)}]^\perp \mathbf{P}^{(n)} \end{bmatrix}$$

But, data normalization must be used

Triangulation

$$\mathbf{A}\mathbf{X} = \mathbf{0}, \text{ where } \mathbf{A} = \begin{bmatrix} [\mathbf{x}^{(1)}]^\perp \mathbf{P}^{(1)} \\ \vdots \\ [\mathbf{x}^{(n)}]^\perp \mathbf{P}^{(n)} \end{bmatrix}$$

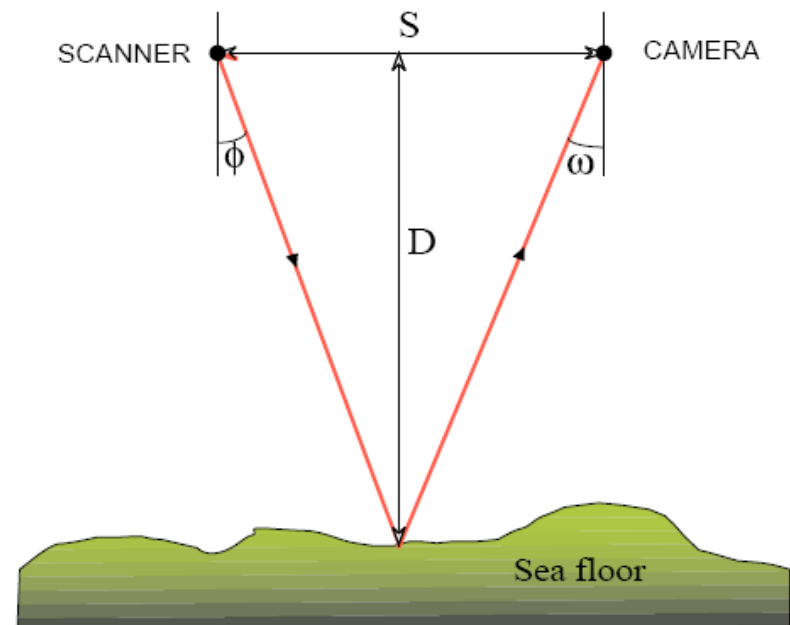
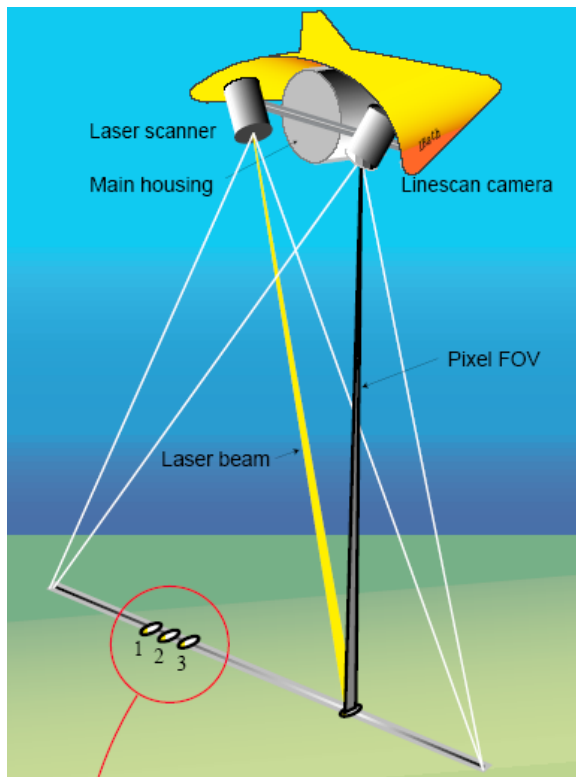
In general, due to error in measurements, the (right) null space of the matrix $\mathbf{A}$ will be empty (i.e., $\mathbf{A}$ will be full rank). Instead, solve for the vector $\mathbf{X}$ such that $\|\mathbf{A}\mathbf{X}\|$ is minimized.

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

The solution vector $\mathbf{X}$ is the column of $\mathbf{V}$ corresponding to the smallest singular value (i.e., the last column of $\mathbf{V}$ (or last row of $\mathbf{V}^\top$)).

Structured illumination

- Application: bathymetry

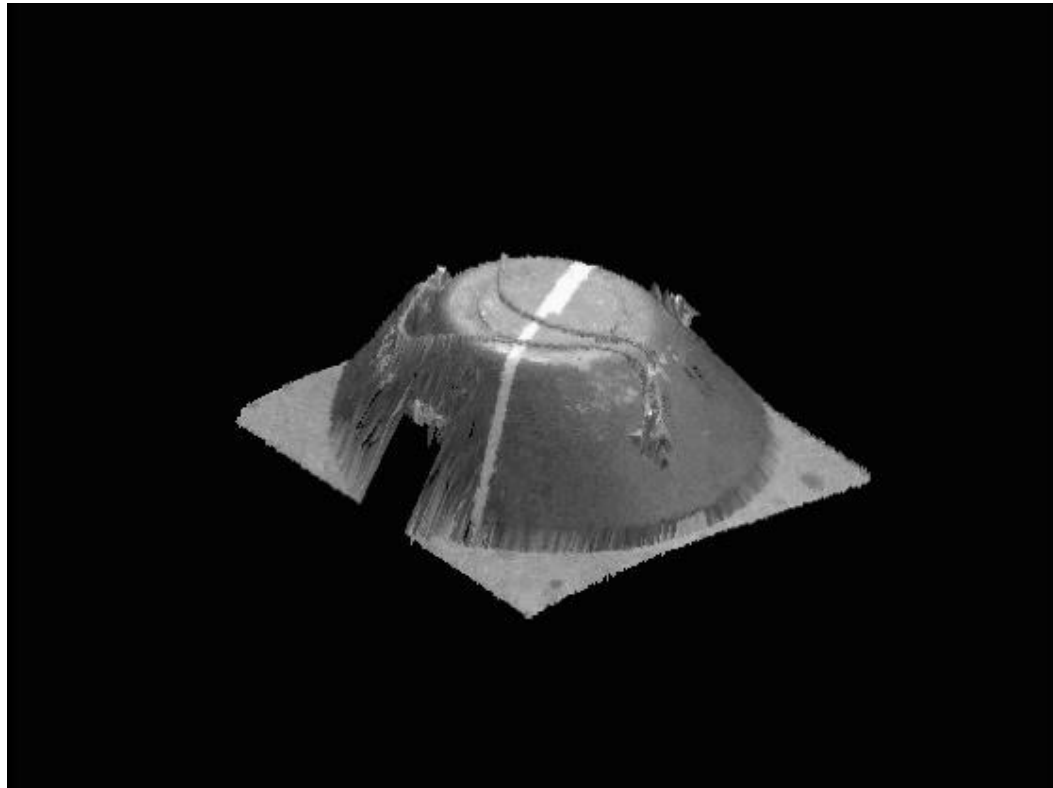


$$D = \frac{S}{\tan \phi - \tan \omega}$$

Triangulation

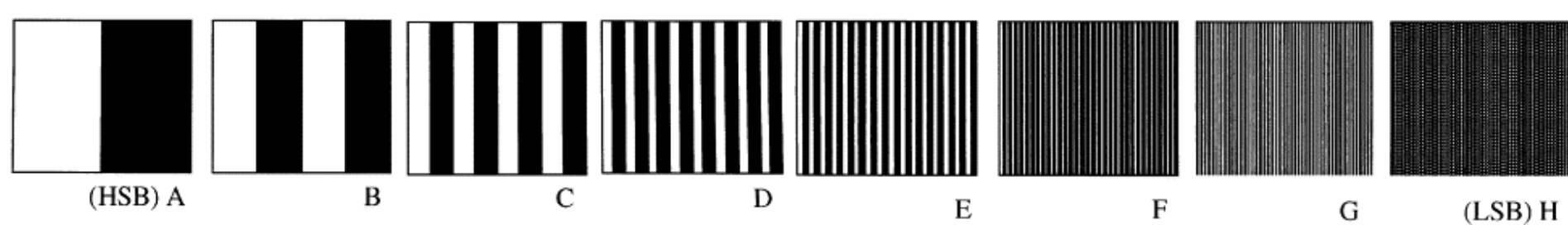
Structured illumination

- Application: bathymetry



Structured illumination

- Video projector, static scene
 - Reflected binary code (i.e., Gray code) sequences
 - Projector has N pixels
 - Each projector pixel has unique binary number
 - Number of images (i.e., bits) is $\lceil \log_2(N) \rceil$
 - For example, horizontal and vertical stripe patterns



Structured illumination

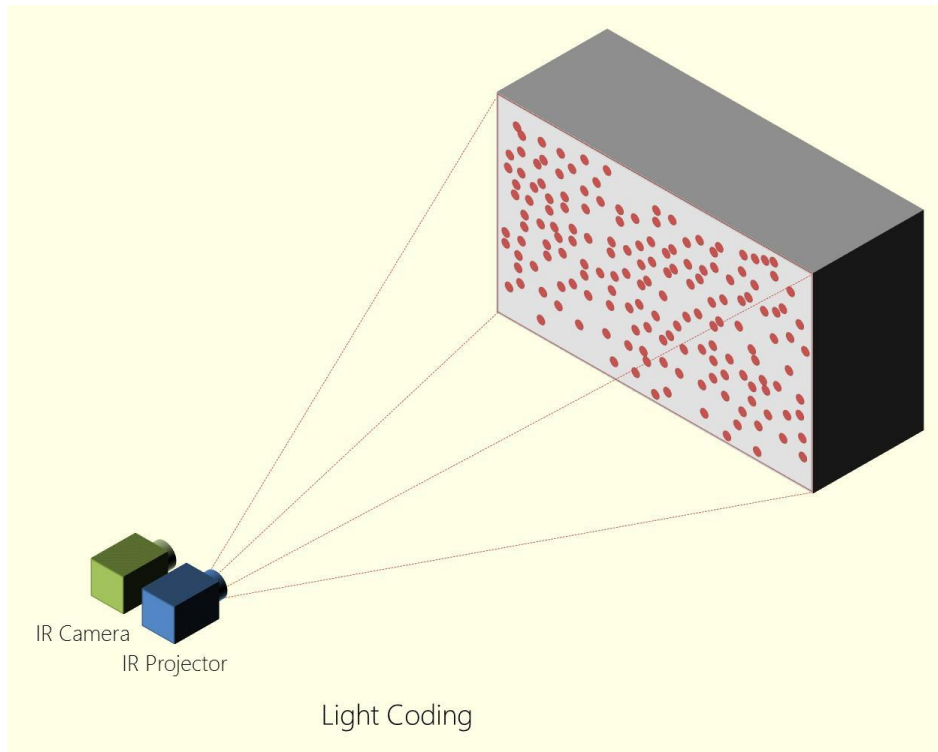
- Video projector, dynamic scene



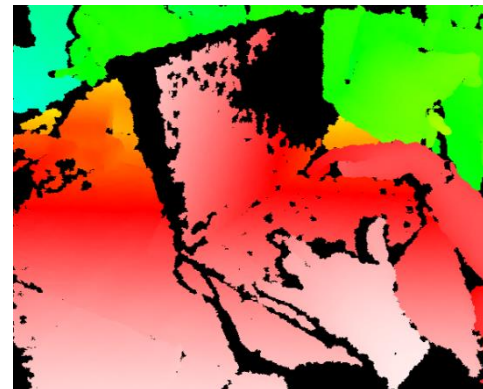
Figure 12.7 Real-time dense 3D face capture using spacetime stereo (Zhang, Snavely, Curless *et al.* 2004) © 2004 ACM: (a) set of five consecutive video frames from one of two stereo cameras (every fifth frame is free of stripe patterns, in order to extract texture); (b) resulting high-quality 3D surface model (depth map visualized as a shaded rendering).

Structured illumination

- Invisible (e.g., infrared) light (e.g., Kinect 1)



IR image



Depth image

Structured illumination

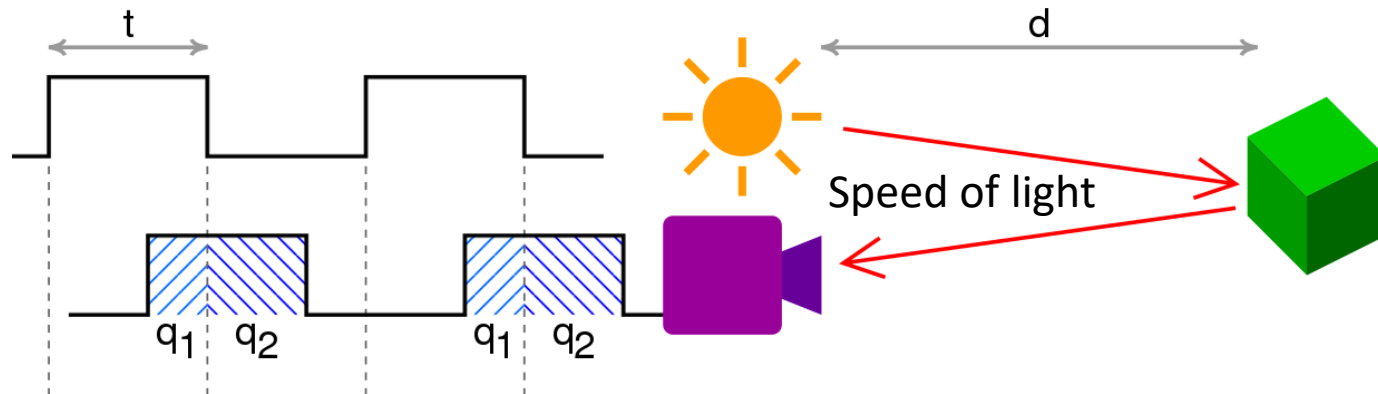
- Kinect 1
 - Released November 2010
 - Claimed the Guinness World Record for “fastest selling consumer electronics device” (sold 8 million units in its first 60 days on the market)



Discontinued

Computational illumination

- Time of flight
 - Measure the round-trip time of a controlled light

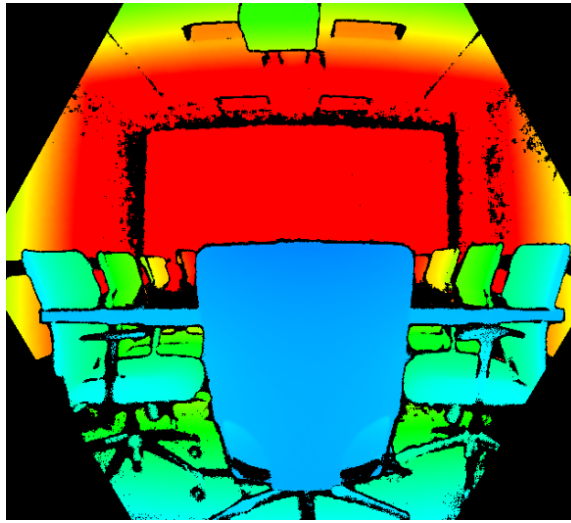


Called lidar, if using laser as light source

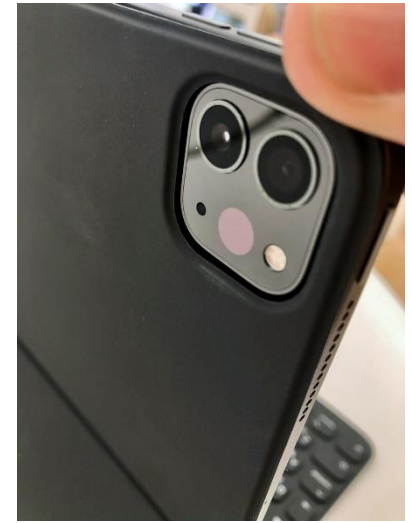
Time of flight



Real-time depth map using a pulsed illumination system (Iddan and Yahav 2001)



Depth image from Azure Kinect DK



Lidar sensor on 4th generation iPad Pro

Time of flight

- Kinect 2



Discontinued

- Azure Kinect DK



Computational illumination summary

- Controlling photographic illumination in a structured manner, then processing the captured images to create new images
- Example methods
 - Photometric stereo
 - Single viewpoint, multiple images under different lighting
 - Structured illumination
 - The projection of a known pattern onto a scene. The scene is reconstructed from one or more images of the scene and deformed pattern.
 - Time of flight
 - Timing is everything

Next lecture

- Group forming meeting