Structured-Light Based Acquisition (Part 1)

CS635
Daniel G. Aliaga
Department of Computer Science
Purdue University

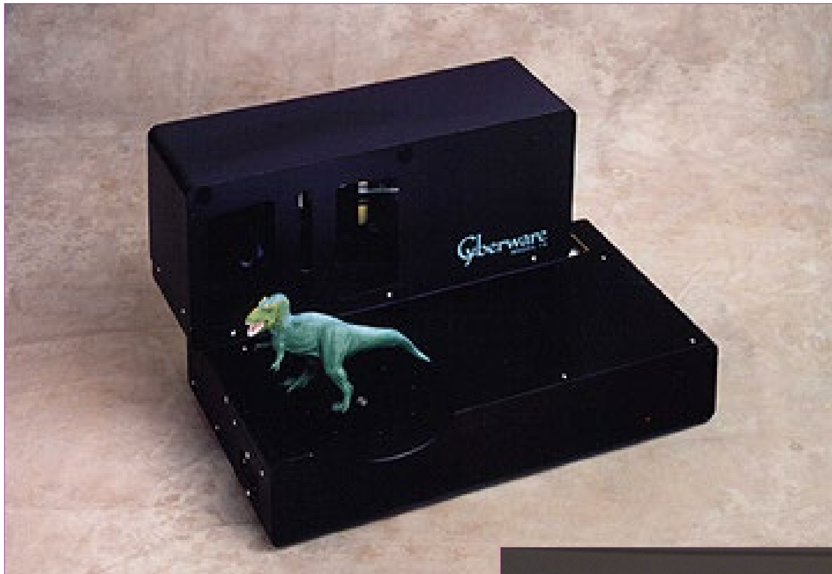
Passive vs. Active Acquisition

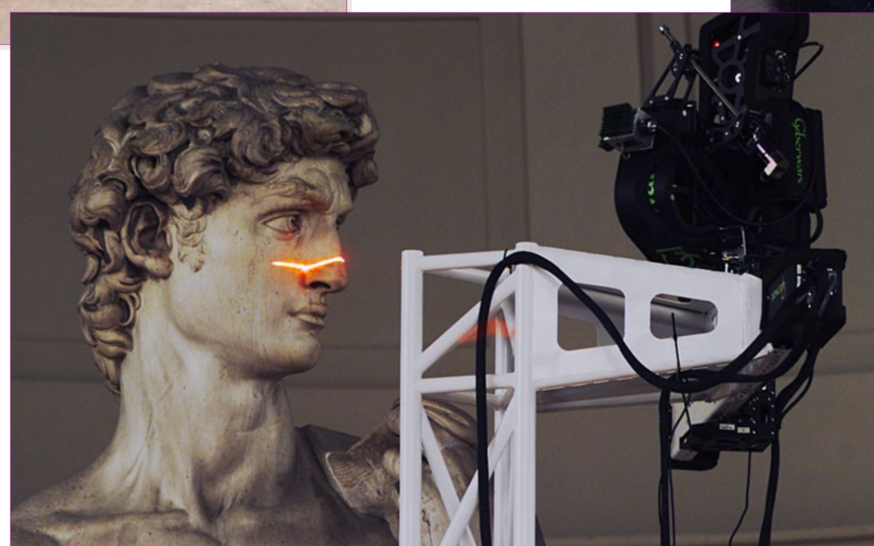
- Passive
  - \+ Just take pictures
  - \+ Does not intrude in the environment (=passive)
  - – Some surfaces cannot be acquired
  - – Robustness is problematic
- Active
  - \+ Emit “light” into the scene so as to force the generation of robust correspondence
  - – Environment is intruded (=active)

Active Acquisition

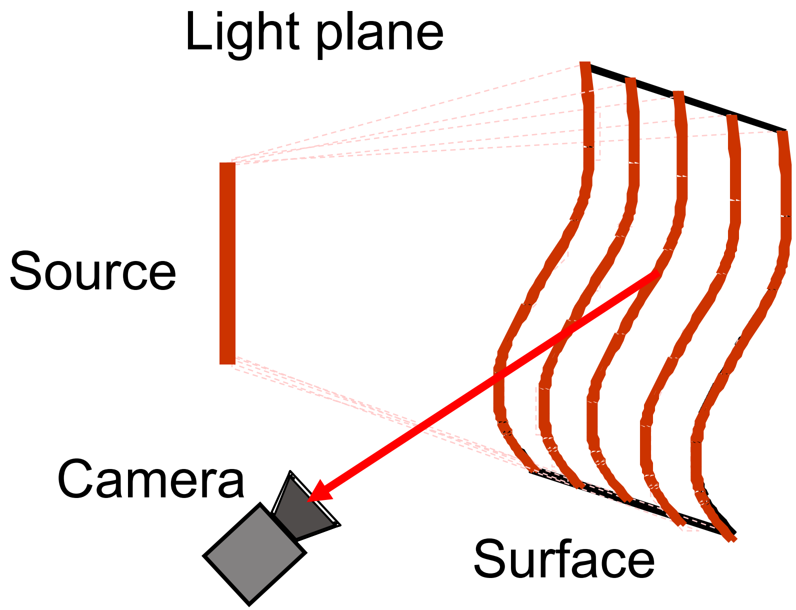
- Some options:
  - Laser scanning
  - “Structured Light”

Laser Scanning



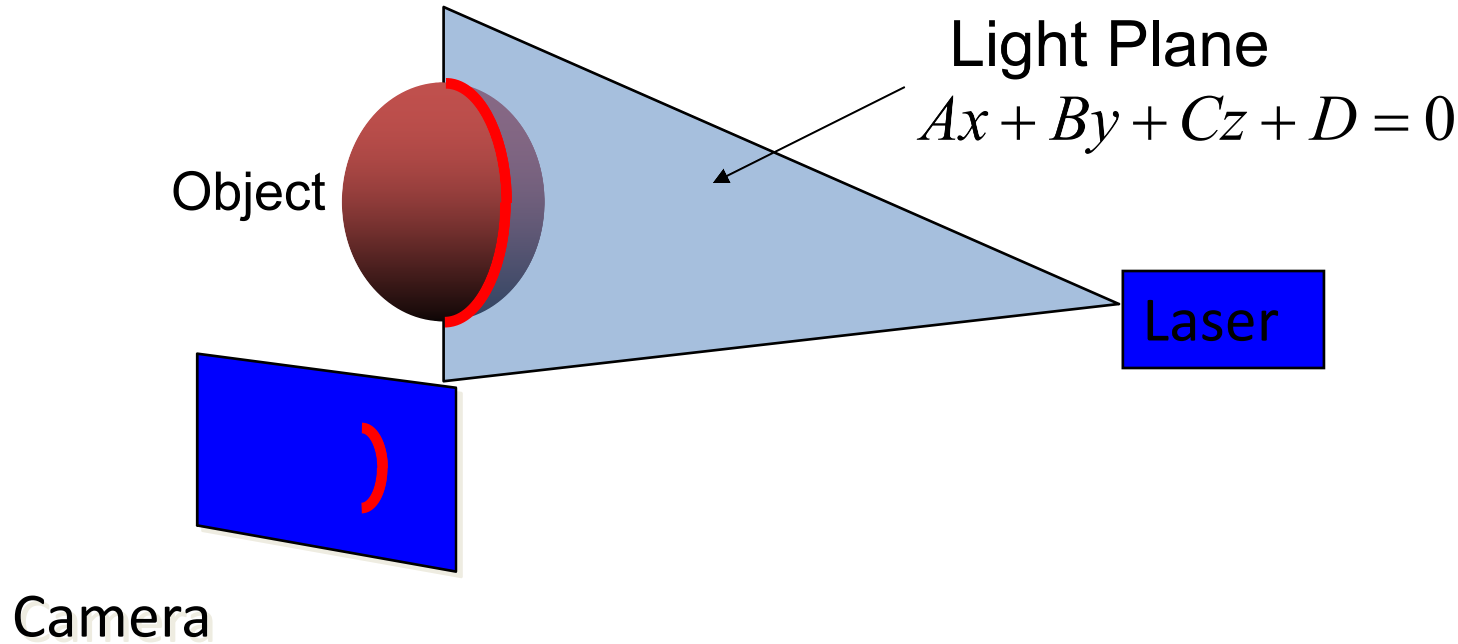


Light Stripe Scanning (Single Stripe)



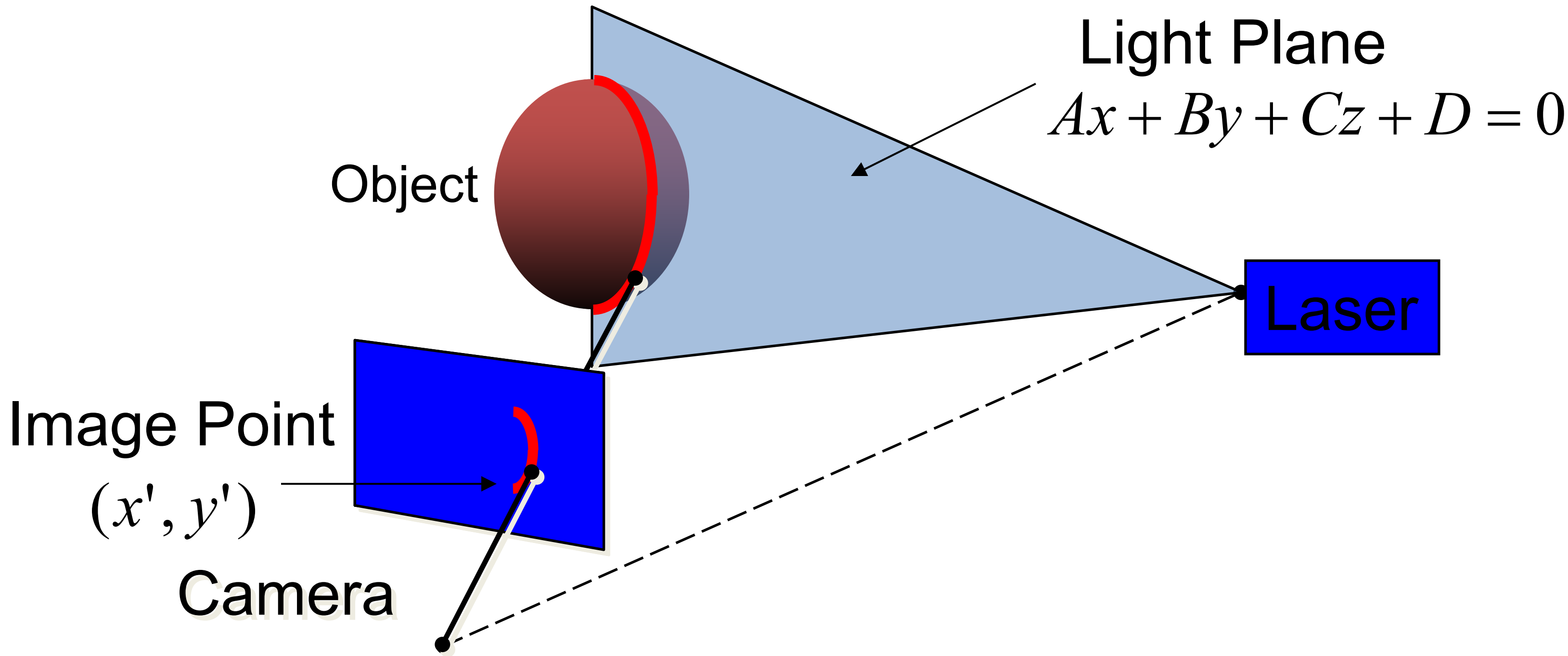
- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning
  - Good for high resolution 3D, but needs many images and takes time

Stripe Triangulation



- Project laser stripe onto object

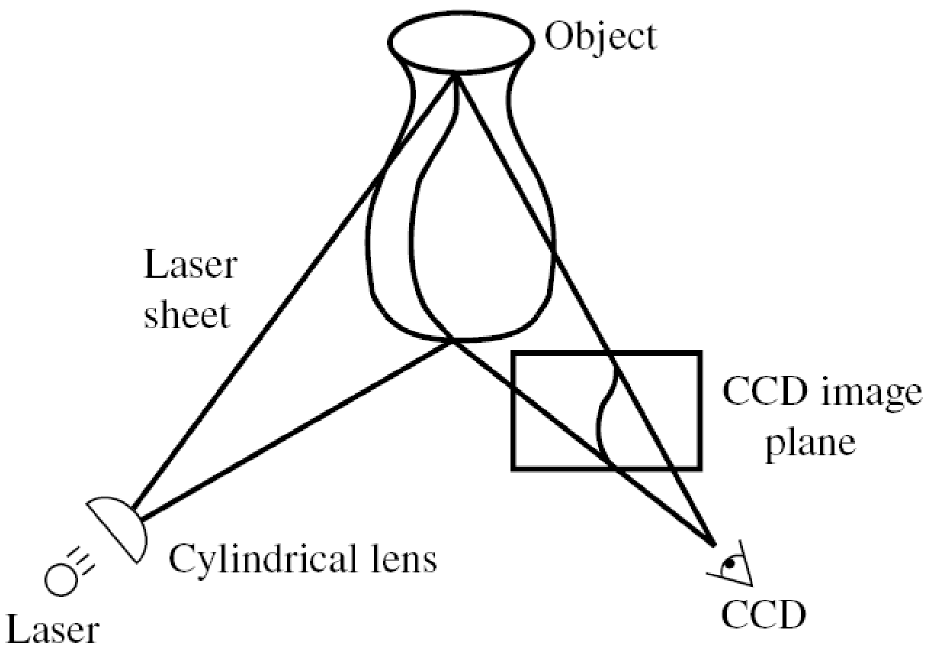
Stripe Triangulation



- Depth from ray-plane triangulation:
  - Intersect camera ray with light plane

$$x = x'z / f \qquad z = \frac{-Df}{Ax' + By' + Cf}$$
$$y = y'z / f$$

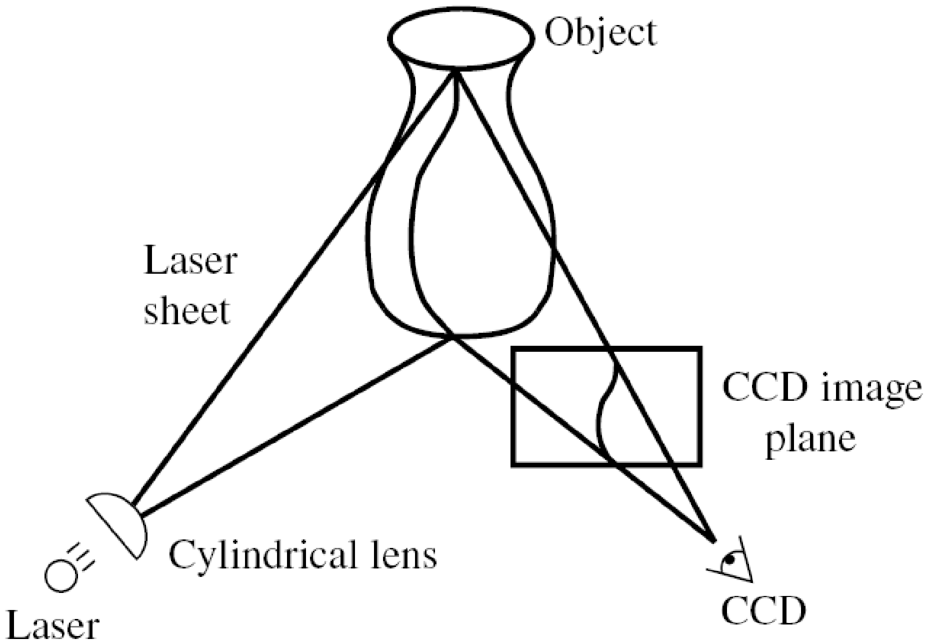
Example: Laser scanner

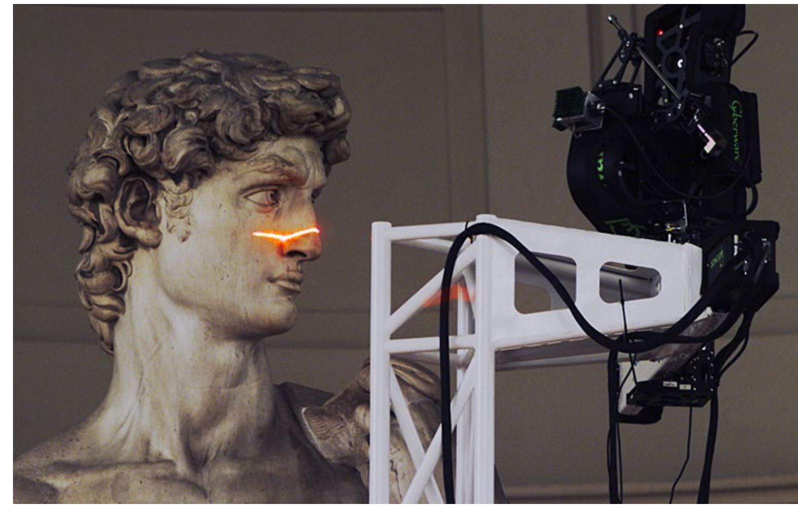


- \+ very accurate < 0.01 mm
- − more than 10sec per scan

Cyberware® face and head scanner

Example: Laser scanner

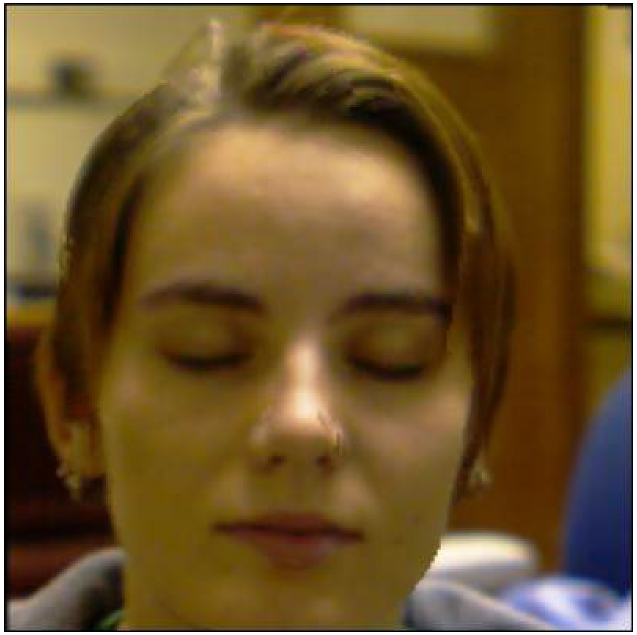




Digital Michelangelo Project

http://graphics.stanford.edu/projects/mich/

Example: Laser scanner



Portable scanner by Minolta

Digital Projector Structured Light

- Goal: generate correspondences so as to enable a robust 3D reconstruction

Digital Projector Structured Light

- Method:
  - Use the projector as a “pattern” generator
  - Have the camera see the “pattern” and generate 1 or more corresponded points

Digital Projector Structured Light

- What are possible patterns?
  - Spatial patterns
  - Temporal patterns
  - Color patterns
  - And combinations of the above

Digital Projector Structured Light

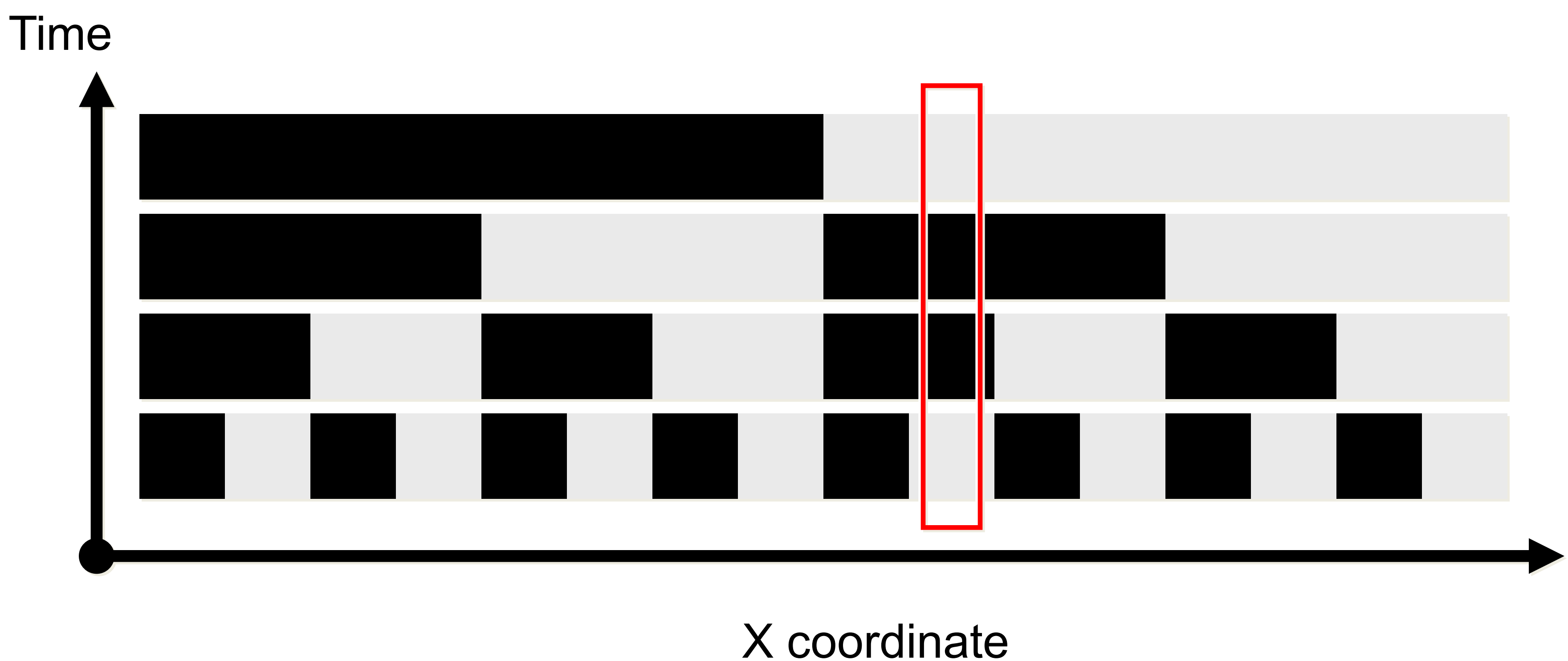
			Scene applicability: Static	Scene applicability: Moving	Pixel depth: Binary	Pixel depth: Grey levels	Pixel depth: Colour	Coding strategy: Periodical	Coding strategy: Absolute
Time-multiplexing	Binary codes	Posdamer et al.	√		√				√
		Inokuchi et al.	√		√				√
		Minou et al.	√		√				√
		Trobina	√		√				√
		Valkenburg and McIvor	√		√				√
		Skocaj and Leonardis	√		√				√
		Rocchini et al.	√				√		√
	n-ary codes	Caspi et al.	√				√		√
		Horn and Kiryati	√			√			√
	Gray code + Phase shifting	Bergmann	√			√			√
		Sansoni et al.	√		√				√
		Wiora	√			√			√
		Gühring	√		√				√
	Hybrid methods	Kosuke Sato	√		√	√			√
		Hall-Holt and Rusinkiewicz		√	√				√
Spatial Neighborhood	Non-formal codification	Maruyama and Abe		√	√			√	
		Durdle et al.		√		√		√	
		Ito and Ishii		√		√			√
		Boyer and Kak		√			√		√
		Chen et al.		√			√		√
	De Bruijn sequences	Hügli and Maître		√			√		√
		Monks et al.		√			√		√
		Vuylsteke and Oosterlinck		√	√				√
		Salvi et al.		√			√		√
		Lavoie et al.		√			√		√
		Zhang et al.	√	√			√		√
	M-arrays	Morita et al.	√		√				√
		Petriu et al.		√	√				√
		Kiyasu et al.		√	√				√
		Spoelder et al.		√	√				√
		Griffin and Yee		√	√		√		√
		Davies and Nixon		√			√		√
		Morano et al.	√		√		√		√
Direct coding	Grey levels	Carrihill and Hummel	√			√			√
		Chazan and Kiryati	√			√			√
		Hung		√		√		√	
	Colour	Tajima and Iwakawa	√				√		√
		Smutny and Pajdla	√				√		√
		Geng		√			√		√
		Wust and Capson		√			√	√	
		Tatsuo Sato	√				√	√	

Lets focus on binary striped patterns…

Binary Pattern Structured Light

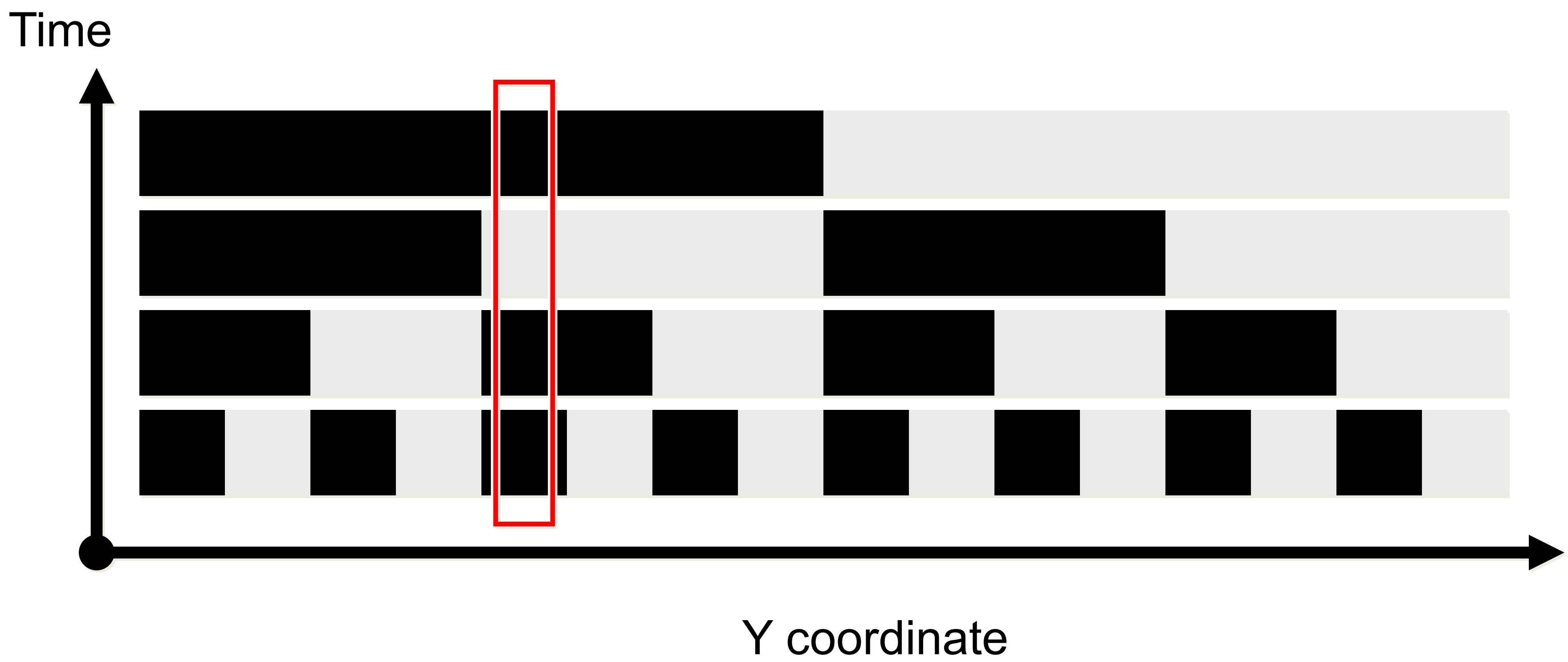
Binary Coding

- Assign each pixel a unique illumination code over time [Posdamer 82]



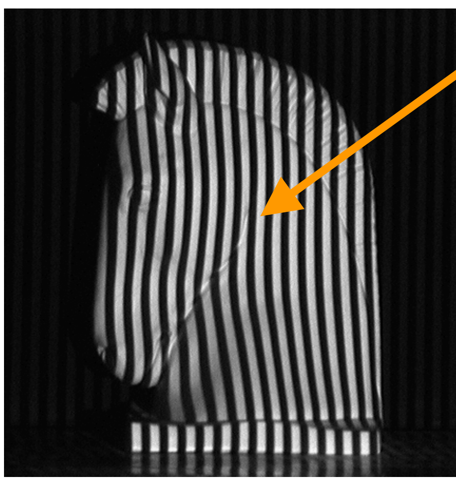
Binary Coding

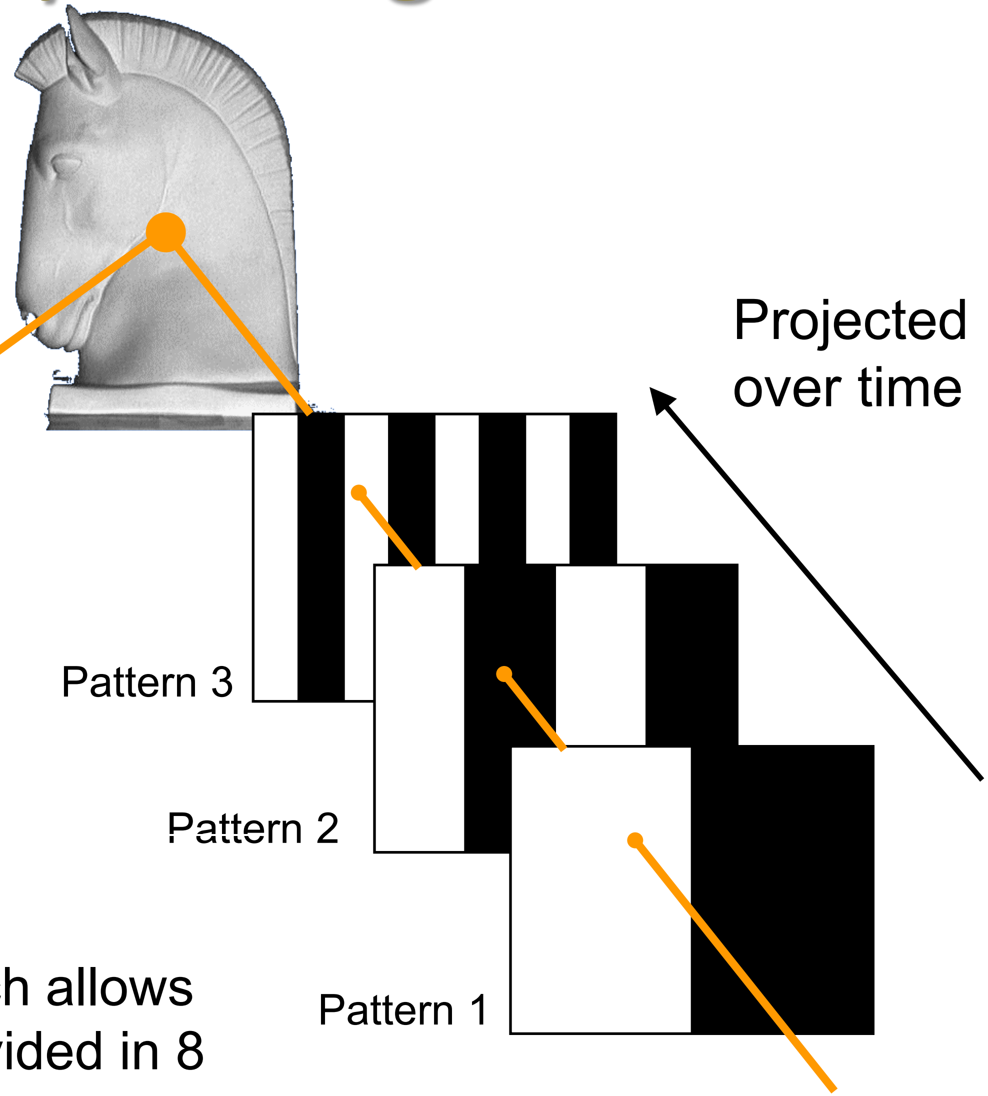
- Assign each pixel a unique illumination code over time [Posdamer 82]



Binary Coding

$2^n - 1$ stripes in n images





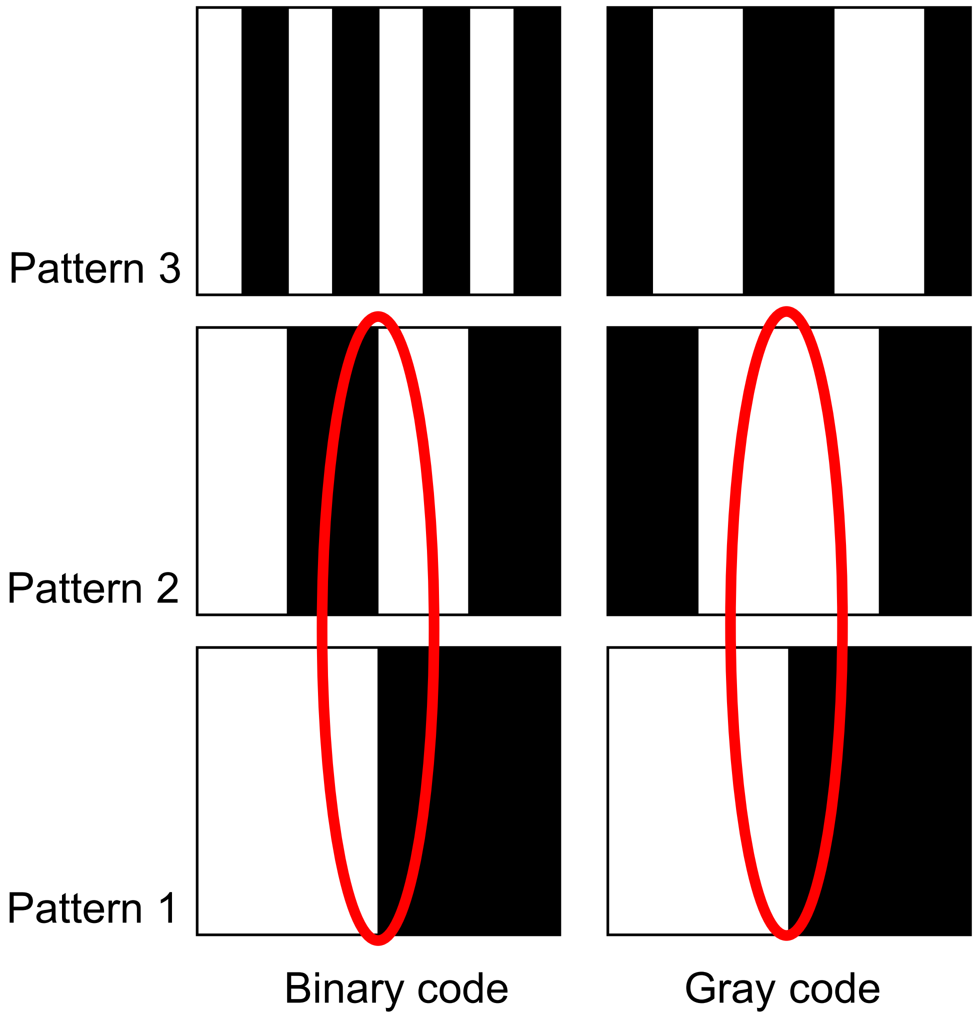
Example:

3 binary-encoded patterns which allows the measuring surface to be divided in 8 sub-regions

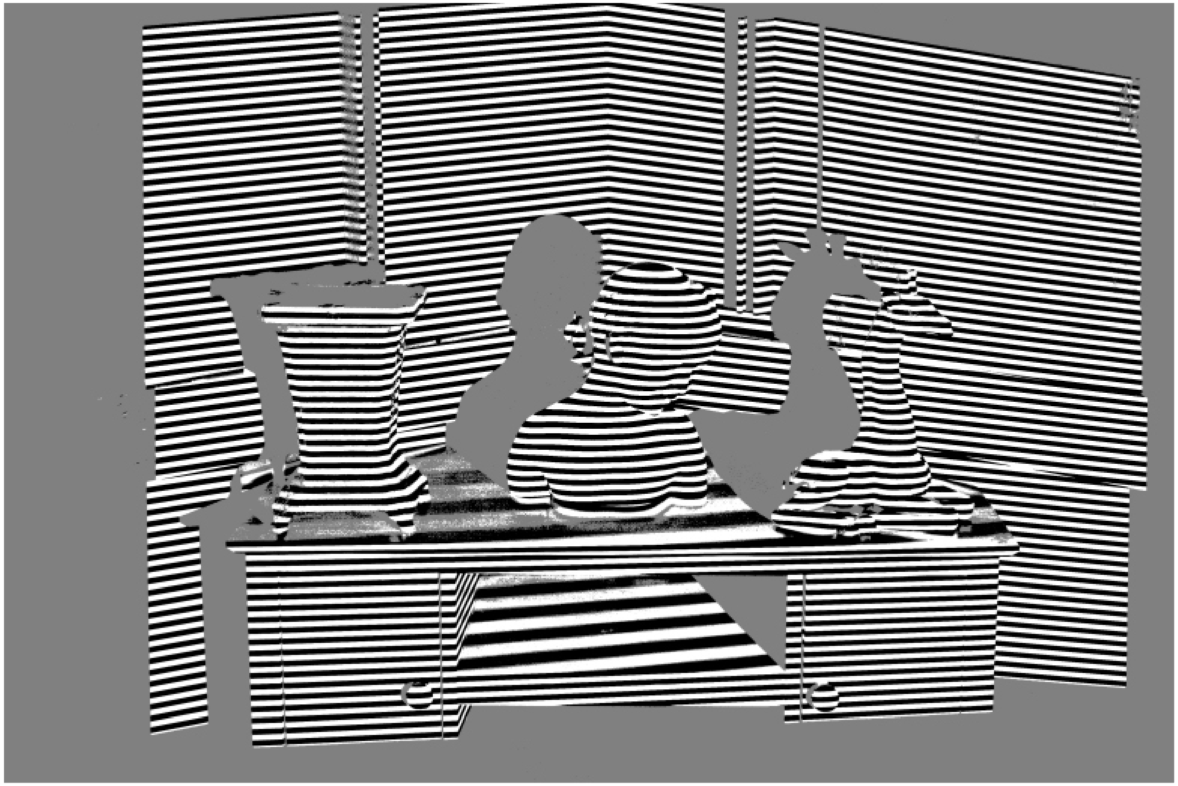
Binary vs Gray Codes

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Binary vs Gray Codes



Standard Pixel Classification

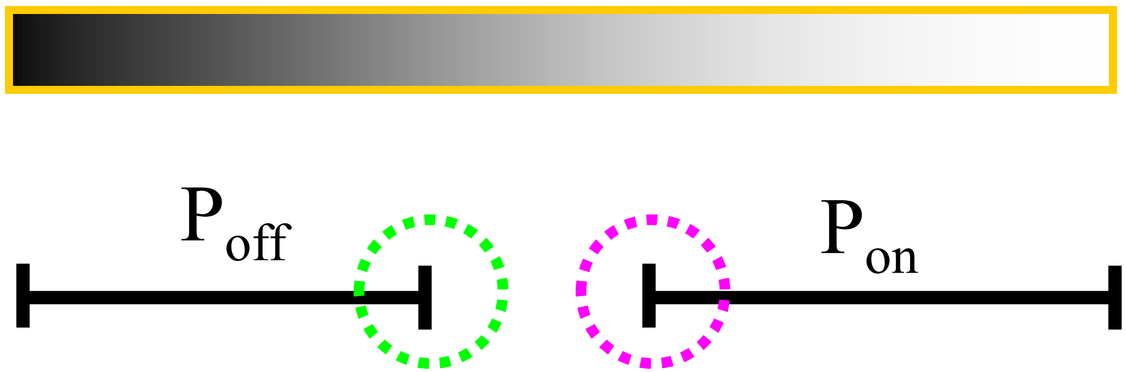


ON(1) OFF(0) Uncertain

Pixel Classification Challenges

Standard Pixel Classification

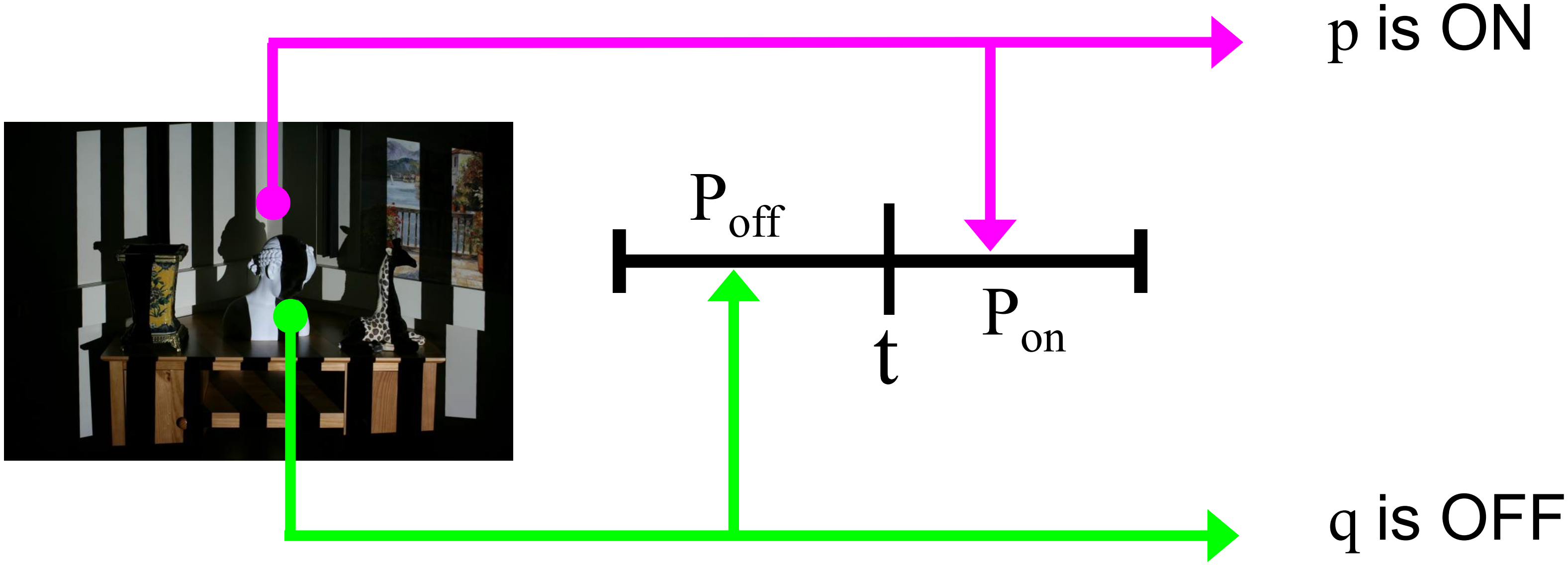
- Interval



- Common methods
  - Simple threshold
  - Albedo threshold
  - Dual pattern

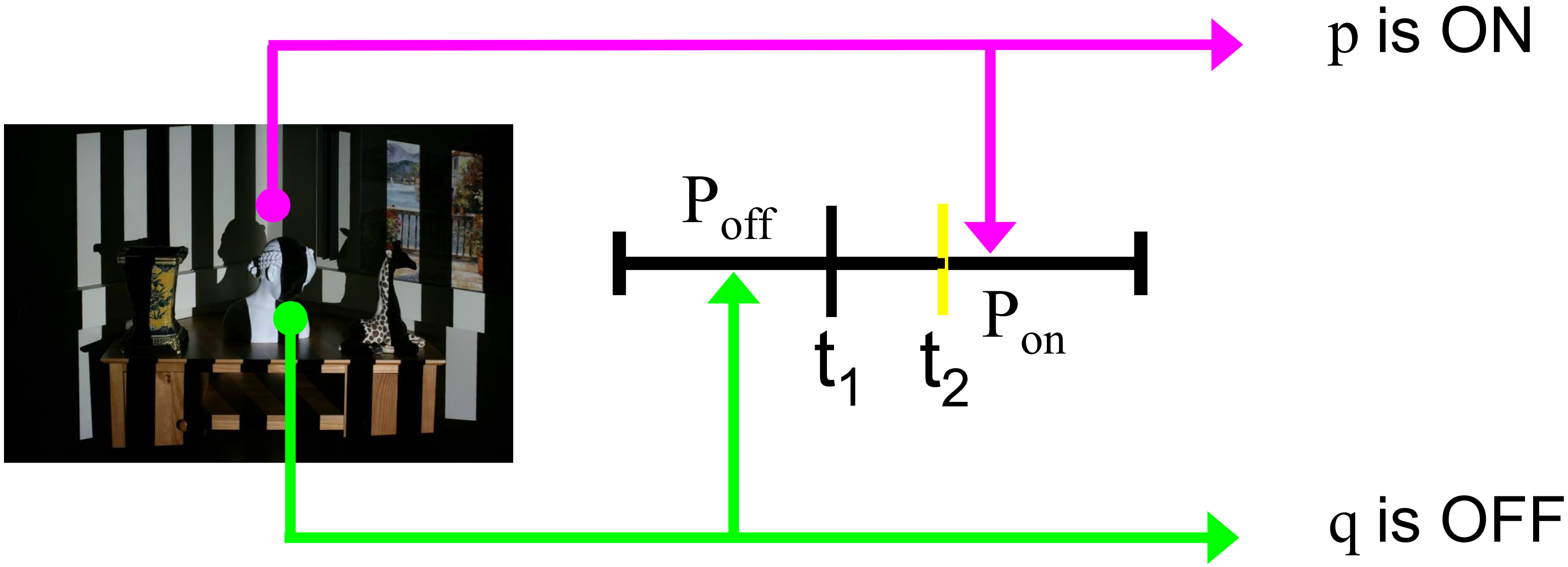
1. Simple Threshold

User specifies one threshold t for all pixels



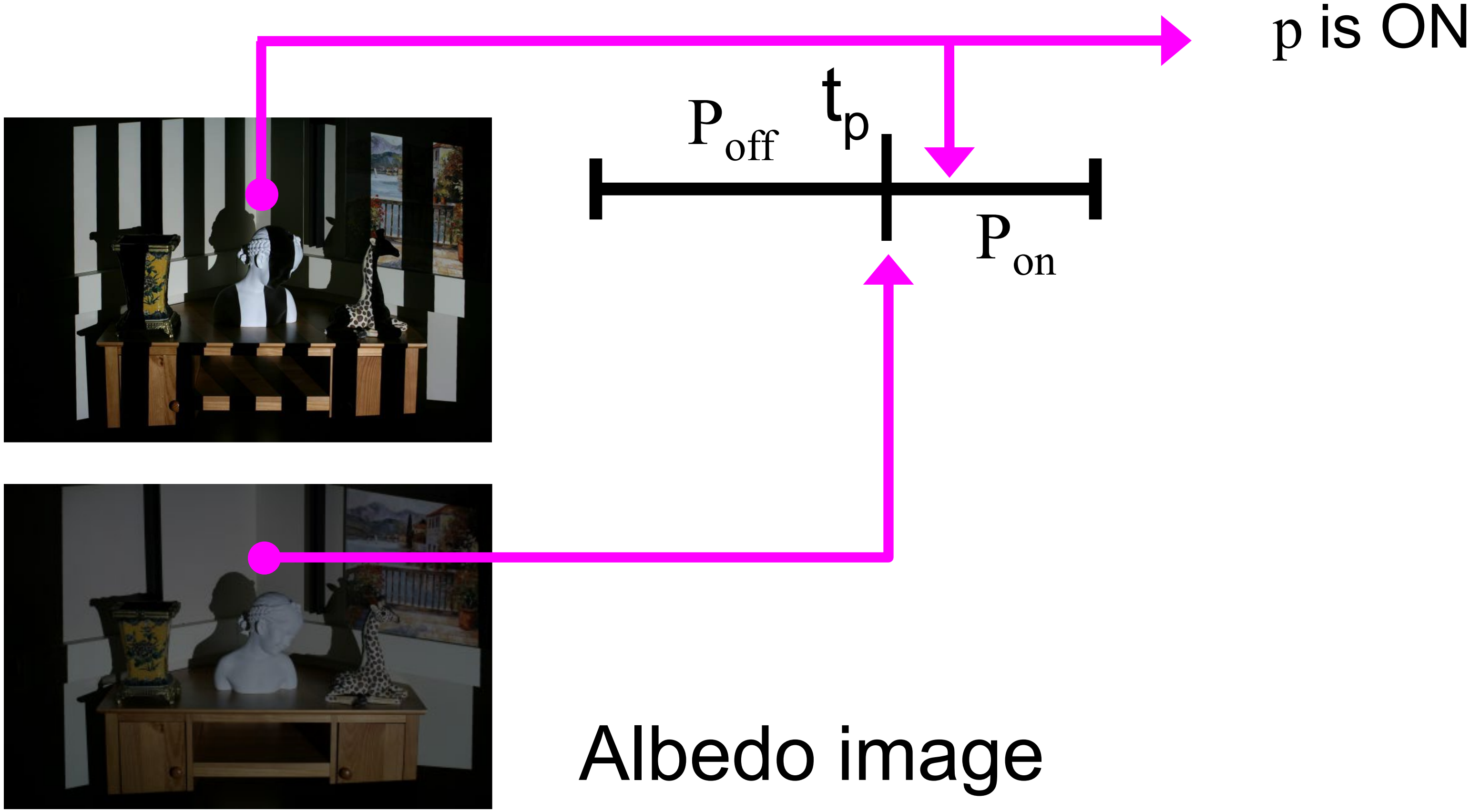
1. Simple Threshold

User specifies two thresholds t_1, t_2



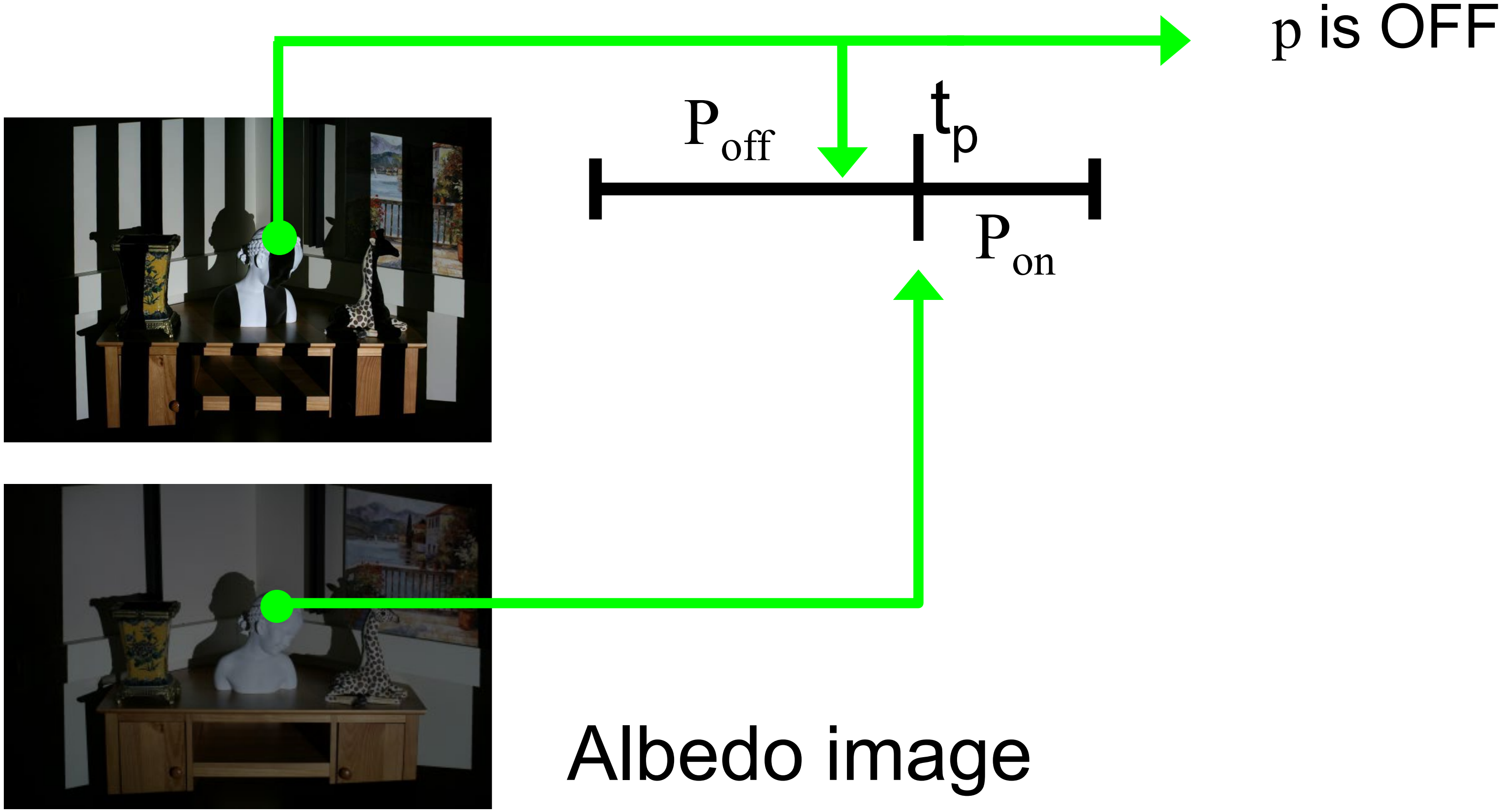
2. Albedo Threshold

Compute the albedo t_p for each pixel



2. Albedo Threshold

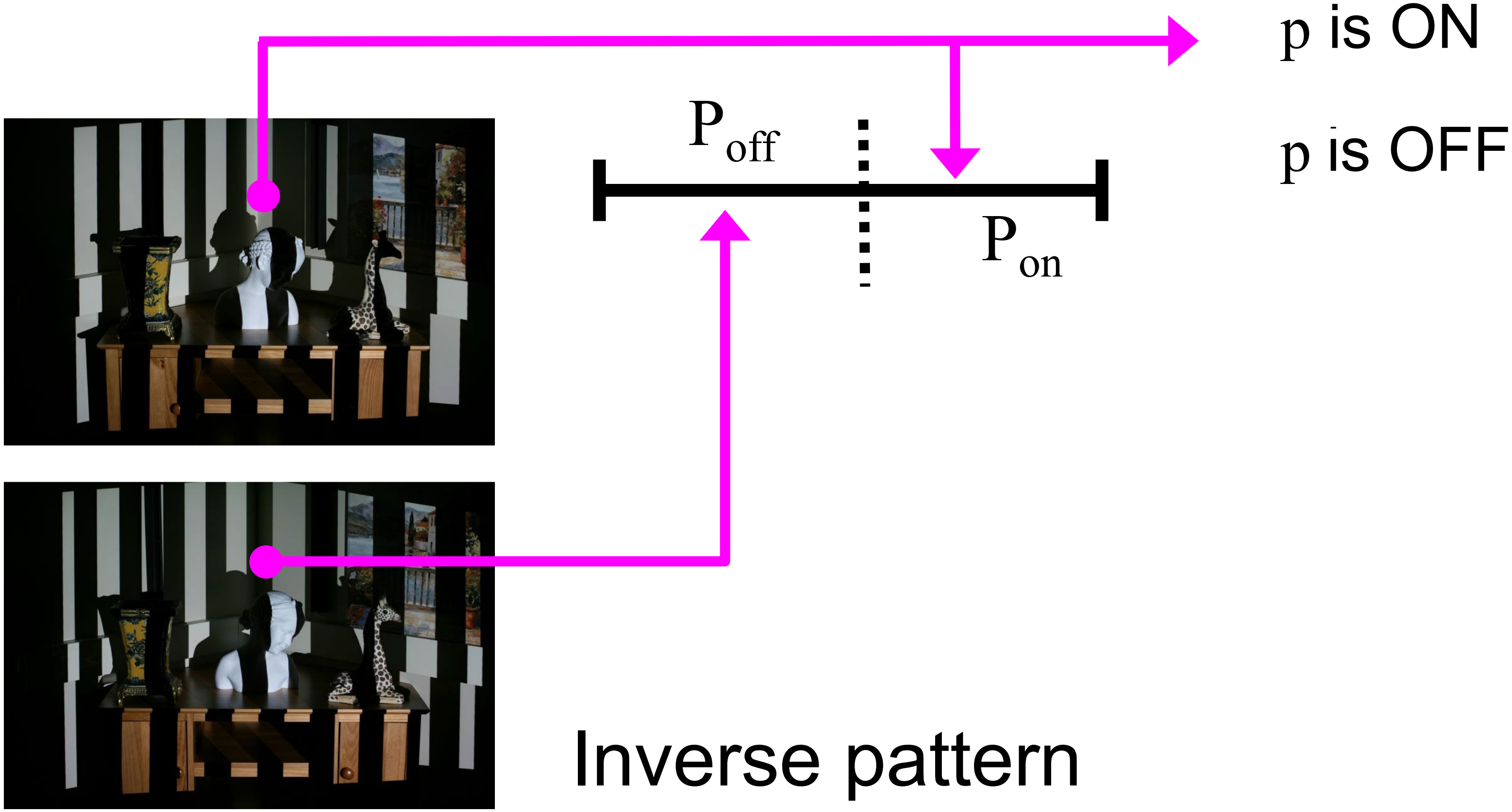
Compute the albedo t_p for each pixel



Albedo image

3. Dual Pattern: Pattern and Inverse

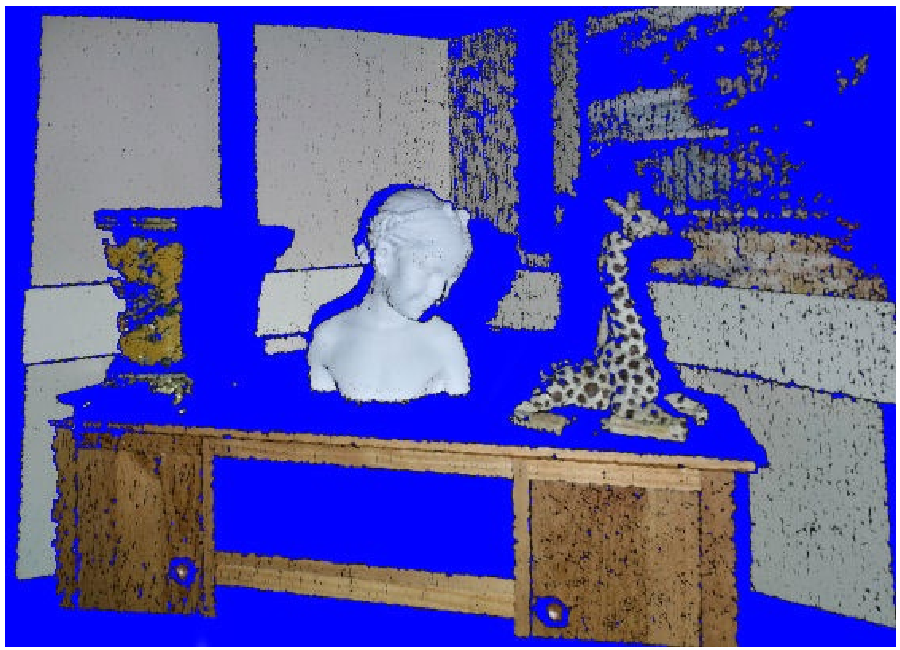
Without explicitly computing t_p



Limitations of Standard Methods

- All three methods assume the two intervals do not overlap
  - This is incorrect when there is strong indirect (global) light
- Haven’t actually established the correct bounds

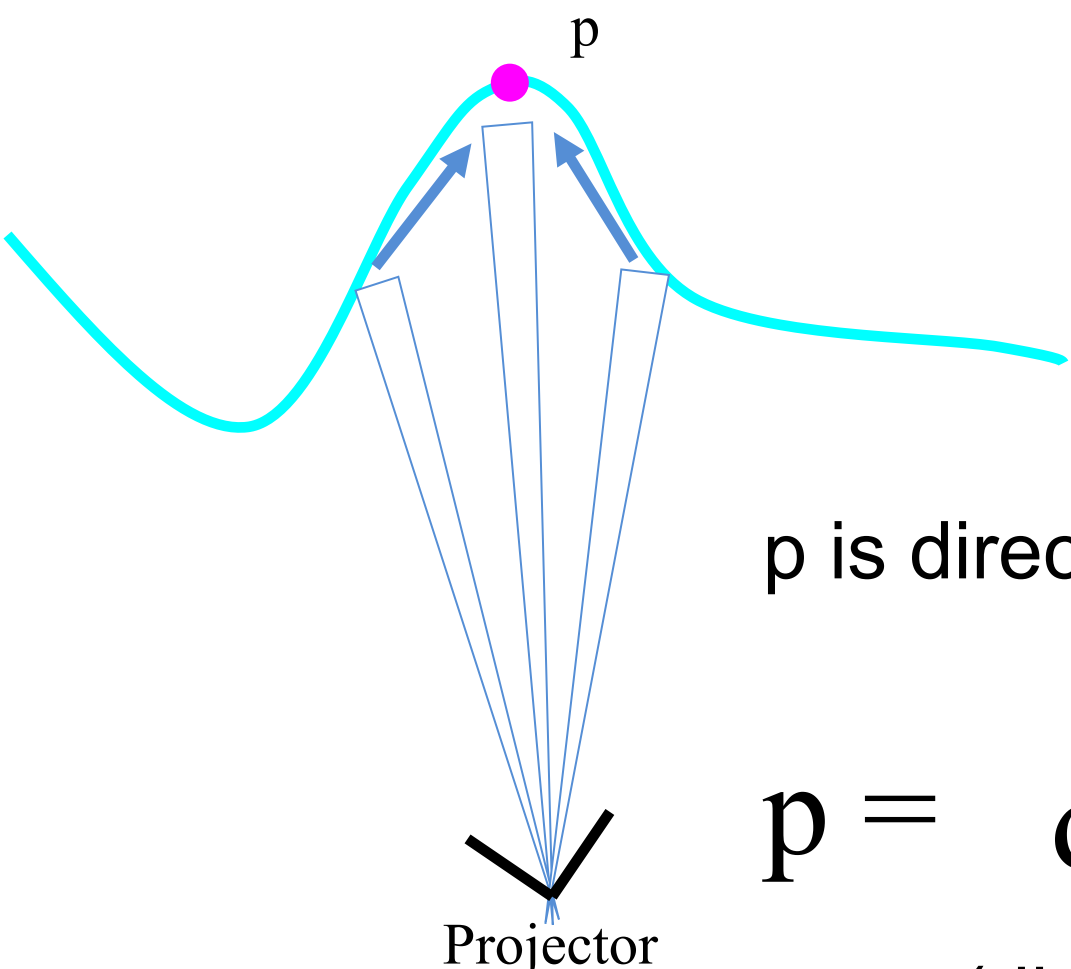
Example Comparison



Using light-separated pixel classification

Using standard pixel classification

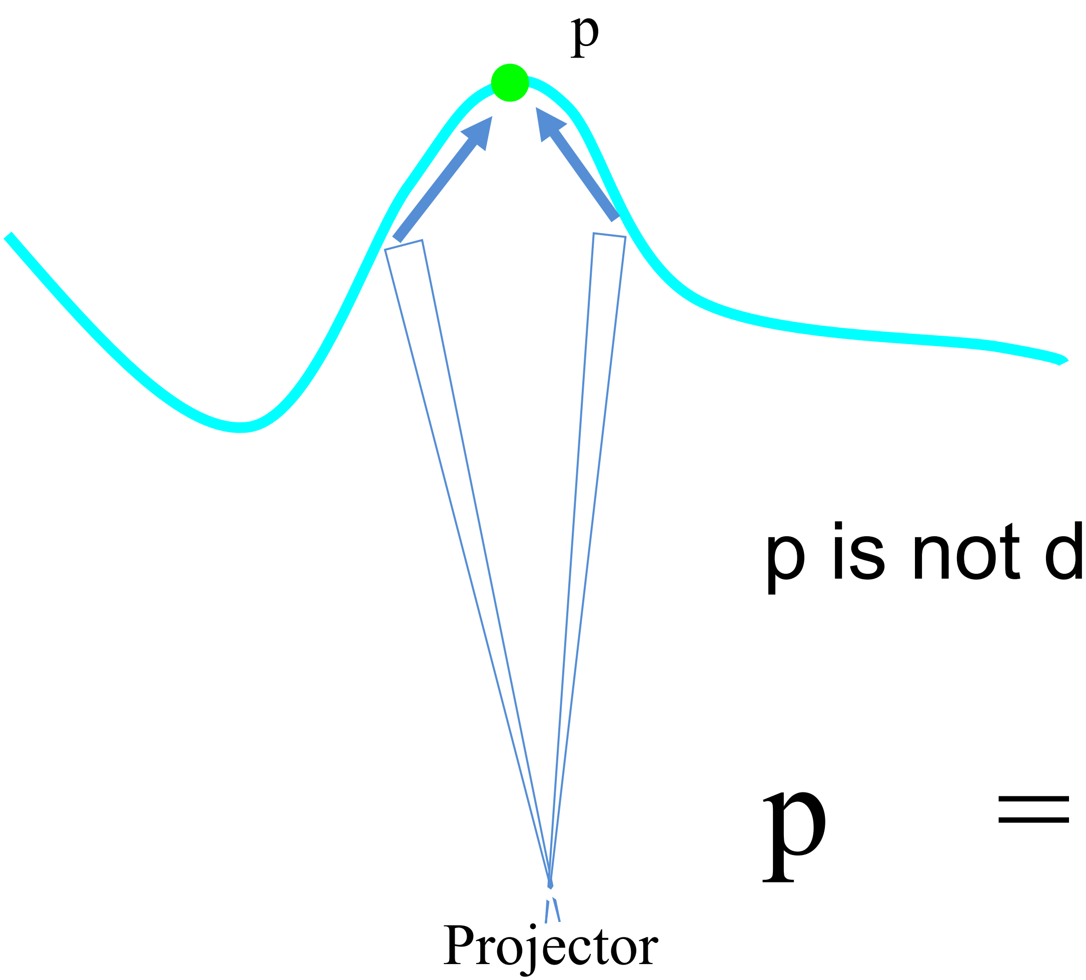
Pixel Intensity (p is ON)



p is directly illuminated

$$p = \underbrace{d}_{\text{(direct)}} + \underbrace{i_{on}}_{\text{(indirect)}}$$

Pixel Intensity (*p* is OFF)



p is not directly illuminated

$$p = i_{off}$$

(indirect)

Pixel Intensity

$$p = \begin{cases} d + i_{on} & \text{If } p \text{ is ON} \\ i_{off} & \text{If } p \text{ is OFF} \end{cases}$$

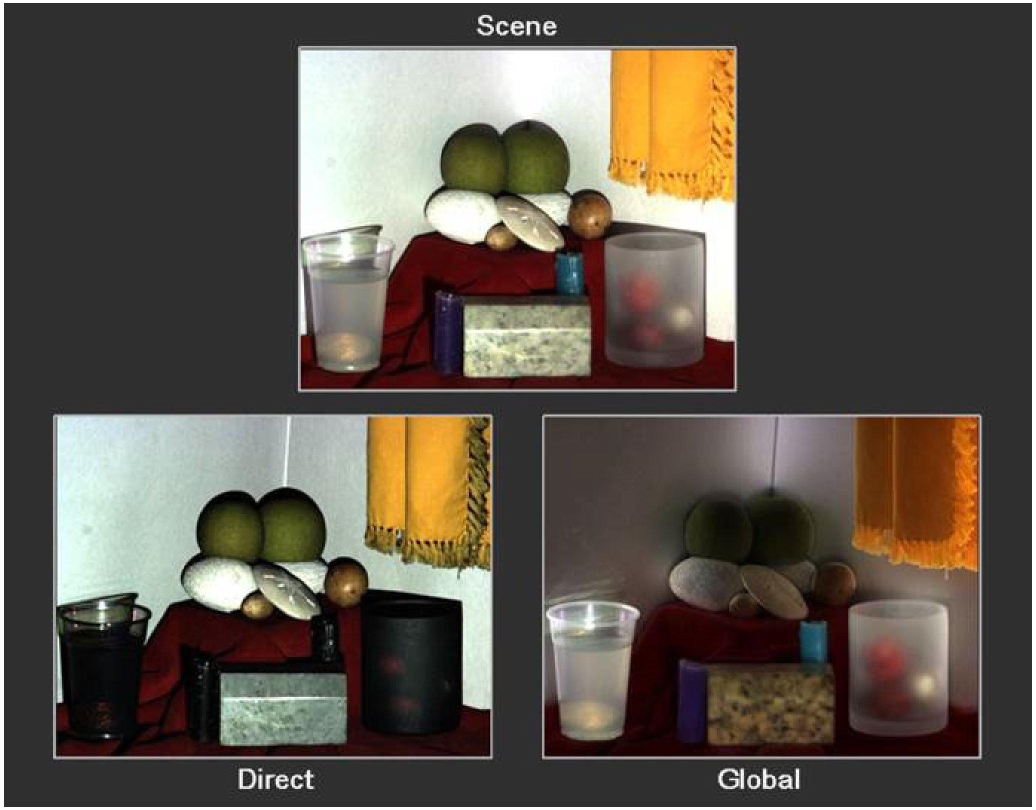
- Chicken and egg problem is
  - Need to know d, i_{on}, i_{off} to classify a pixel.
  - ~~Need to classify a pixel to know~~ $\cancel{d, i_{on}, i_{off}}$

ALL white pattern

(all projector pixels on)

Direct and Indirect Separation

- Direct and indirect (global) components of each pixel can be separated easily (Nayar et al. SIGGRAPH'06).

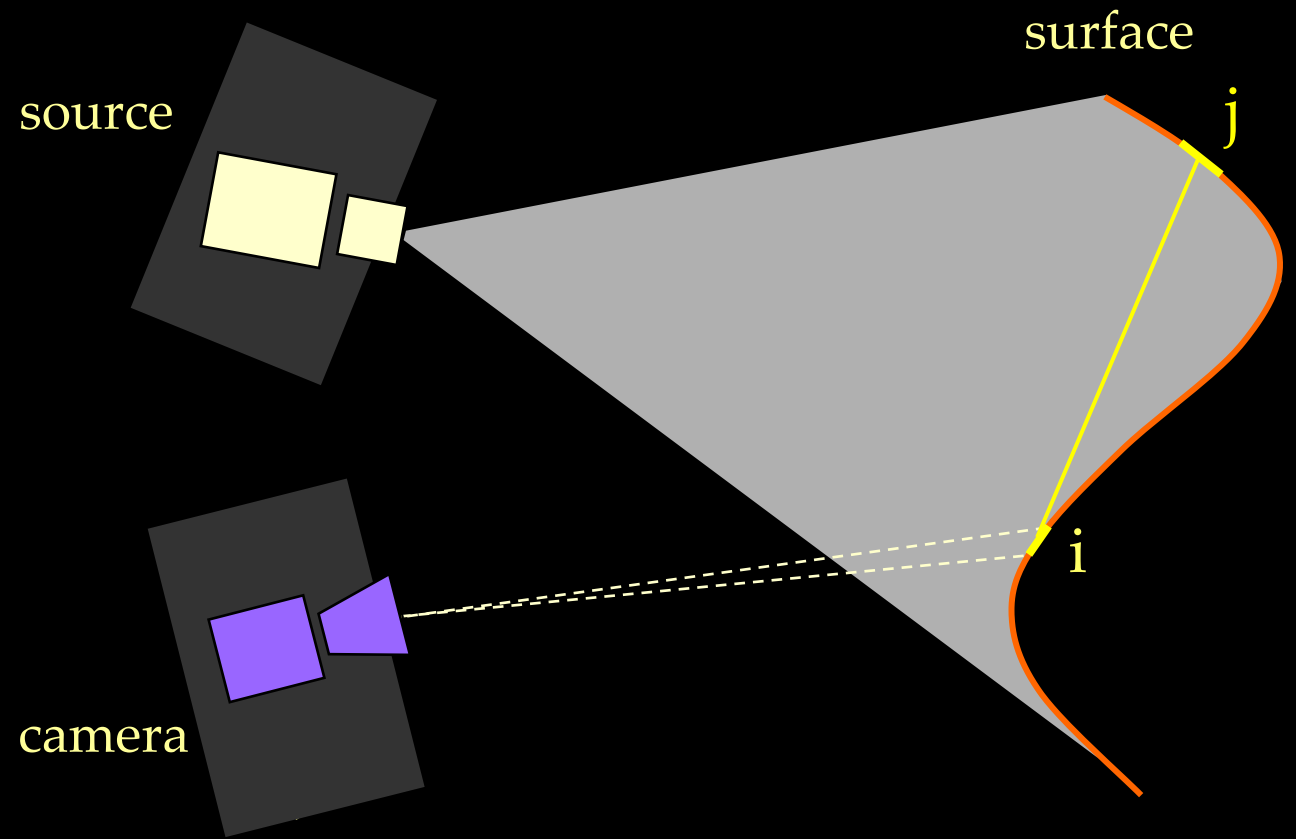


Direct and Indirect Separation

- Direct and indirect (global) components of each pixel can be separated easily (Nayar et al. SIGGRAPH'06).

https://cave.cs.columbia.edu/gallery/separation/RWO

Direct and Global Components: Interreflections



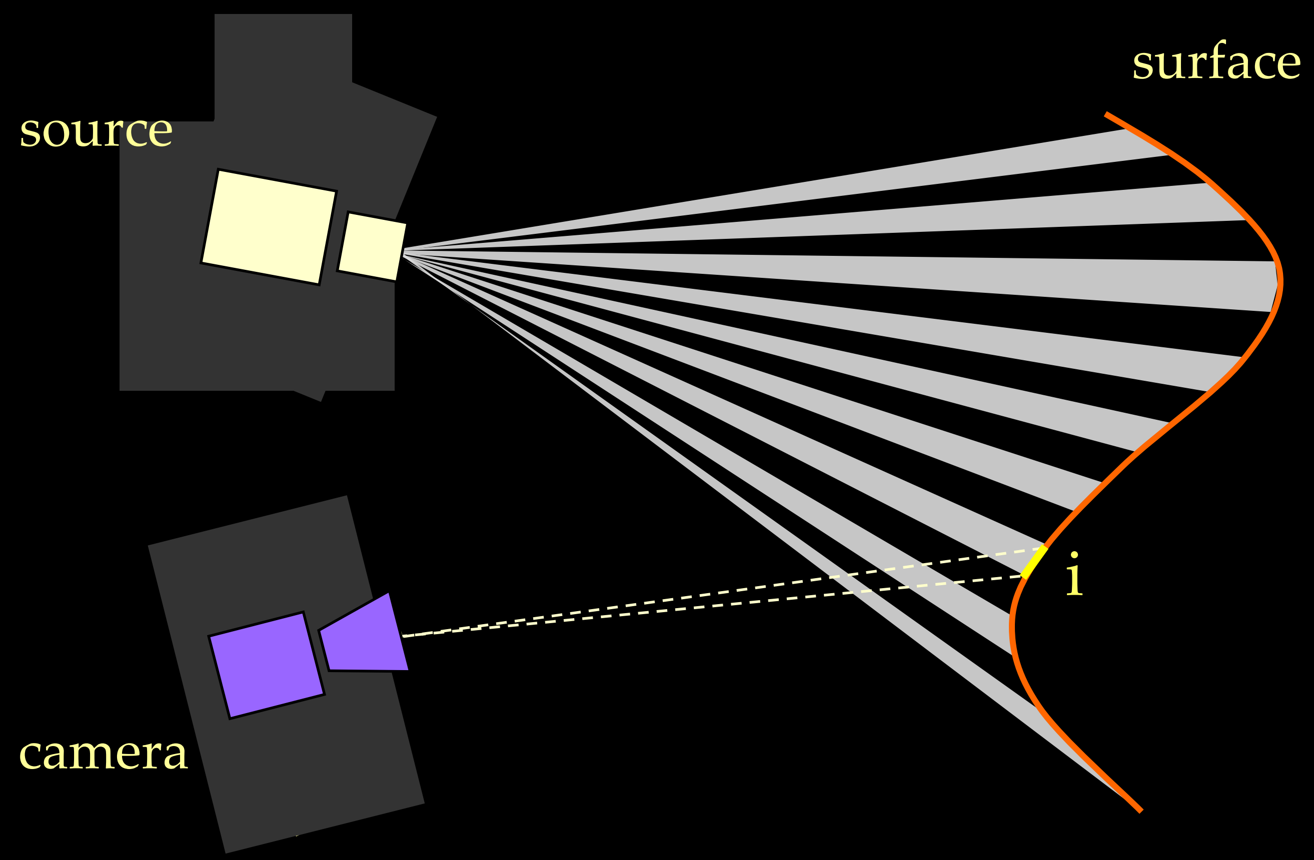
$$L[c,i] = L_d[c,i] + L_g[c,i]$$

radiance direct global

$$L_g[c,i] = \sum_P A[i,j]\, L[i,j]$$

BRDF and geometry

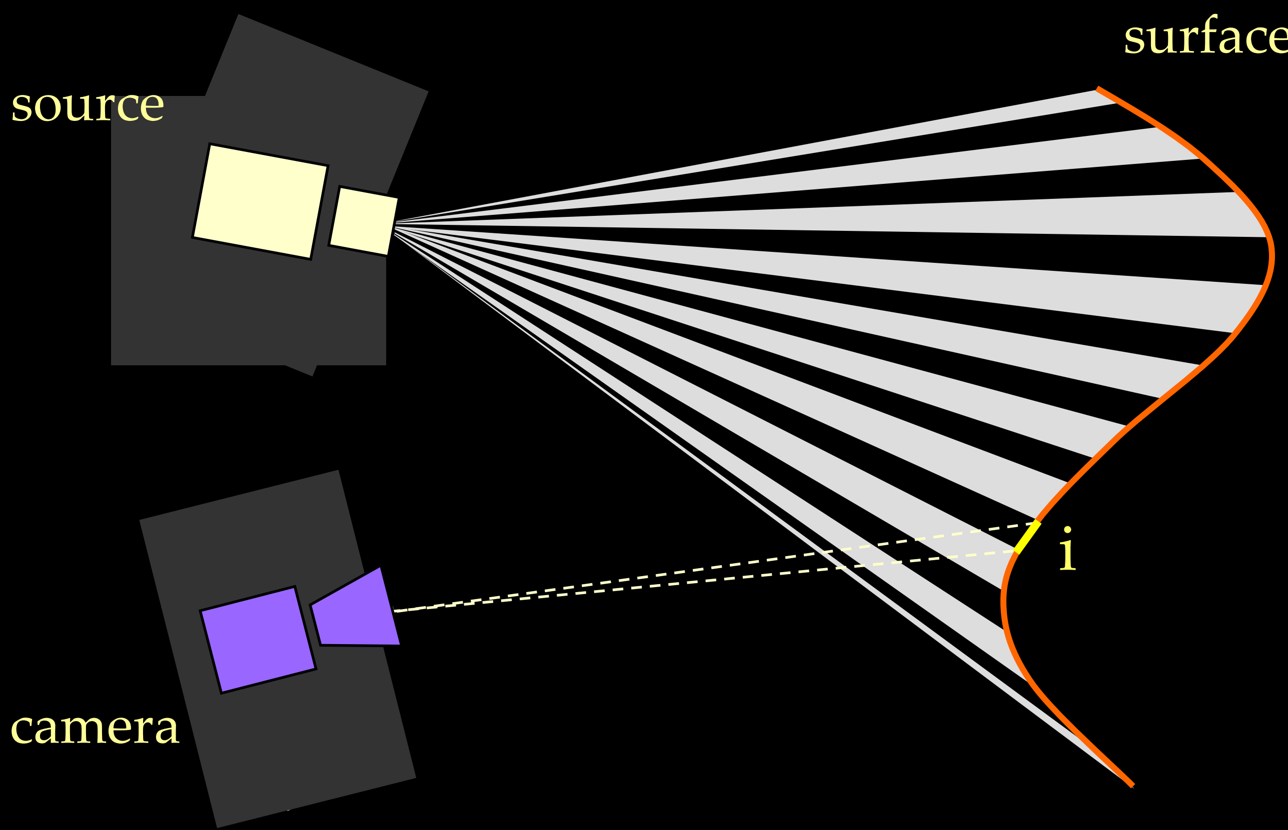
High Frequency Illumination Pattern



$$L^{+}[c,i] = L_d[c,i] + \alpha\, L_g[c,i]$$

fraction of activated source elements

High Frequency Illumination Pattern



$$L^{+}[c,i] = L_d[c,i] + \alpha L_g[c,i]$$

$$L^{-}[c,i] = (1-\alpha) L_g[c,i]$$

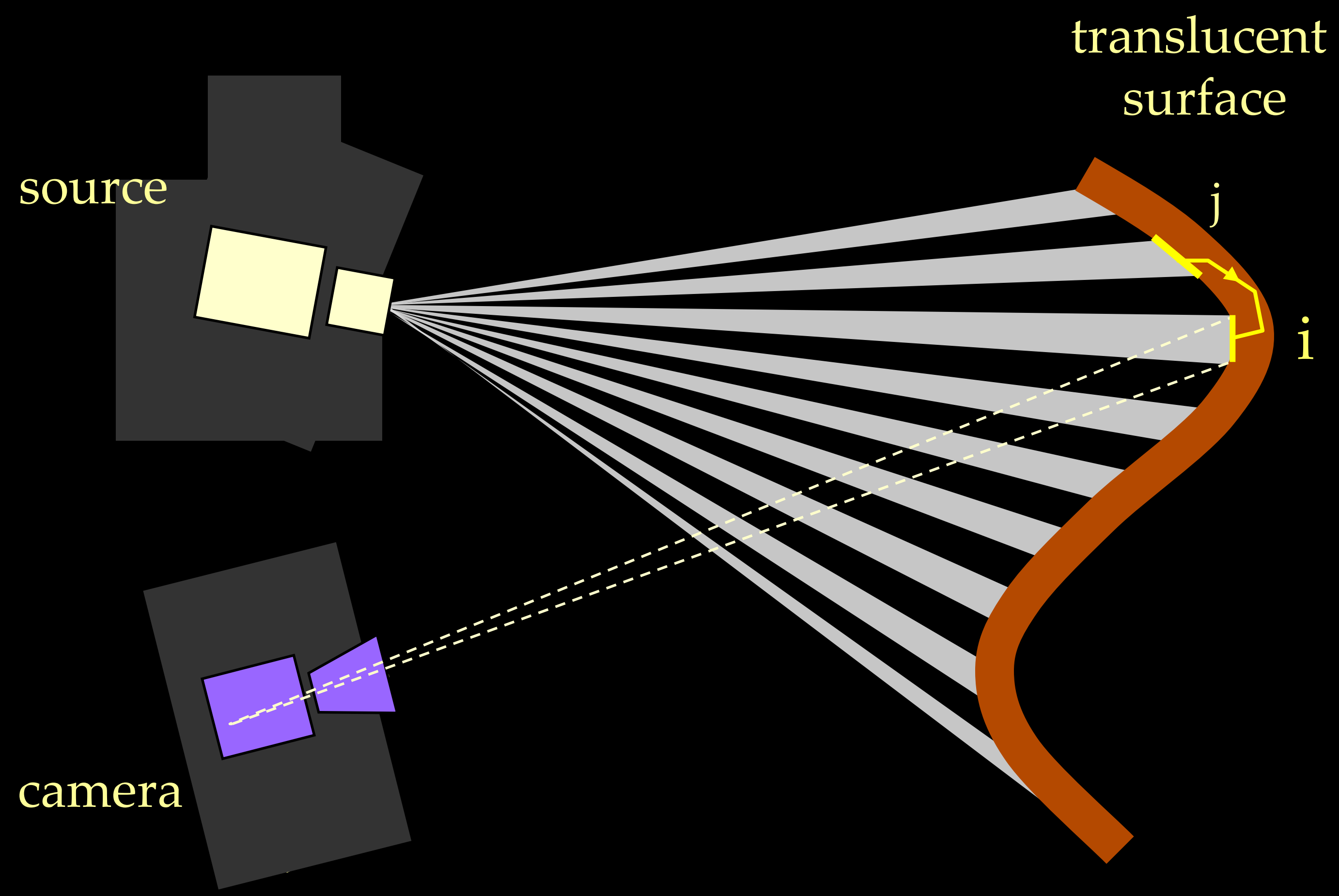
fraction of activated source elements

Separation from Two Images

$$\alpha = \frac{1}{2}: \quad \boxed{L_d = L_{\max} - L_{\min}, \quad L_g = 2L_{\min}}$$

direct global

Other Global Effects: Subsurface Scattering



Direct and Indirect Separation

- Direct and indirect (global) components of each pixel under ALL white pattern can be separated easily (Nayar et al. SIGGRAPH'06).

$$p \quad = \quad d_{total} \quad + \quad i_{total}$$

Direct and Indirect Separation

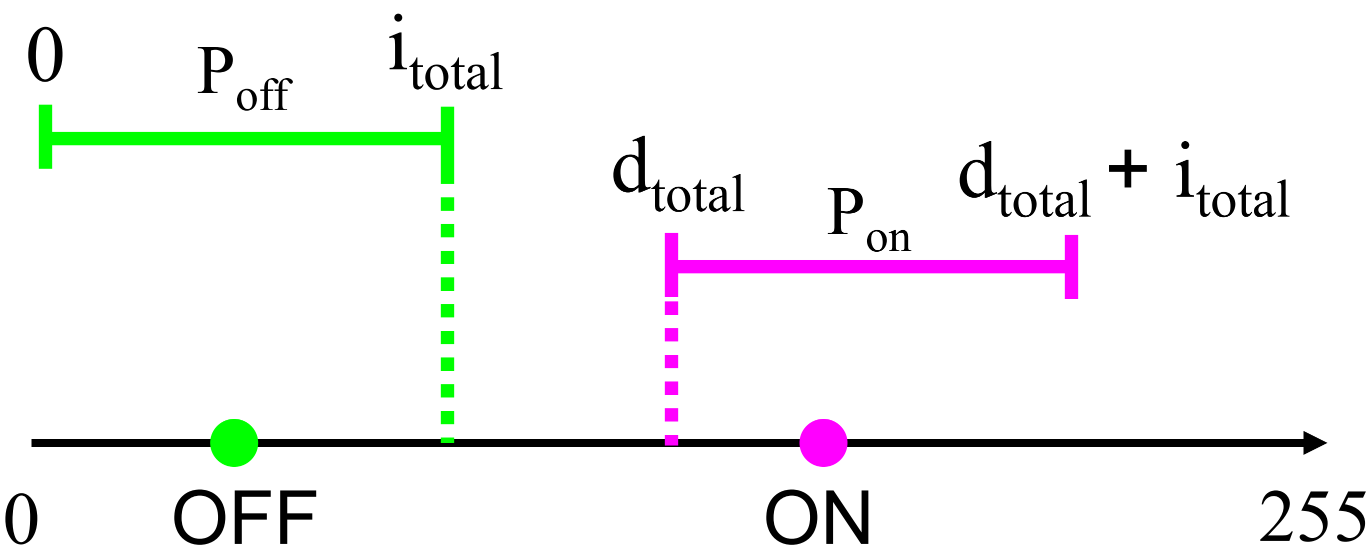
- Project high frequency binary pattern and its inverse to separate light components.

- Structured light patterns include the separation patterns.
  - Thus, separation can be applied to previously captured data to obtain per pixel

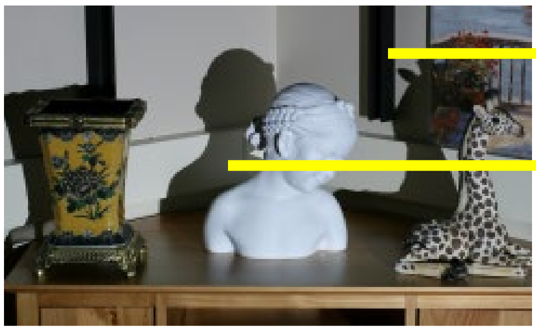
$$p = d_{total} + i_{total}$$

Pixel Classification Scenarios

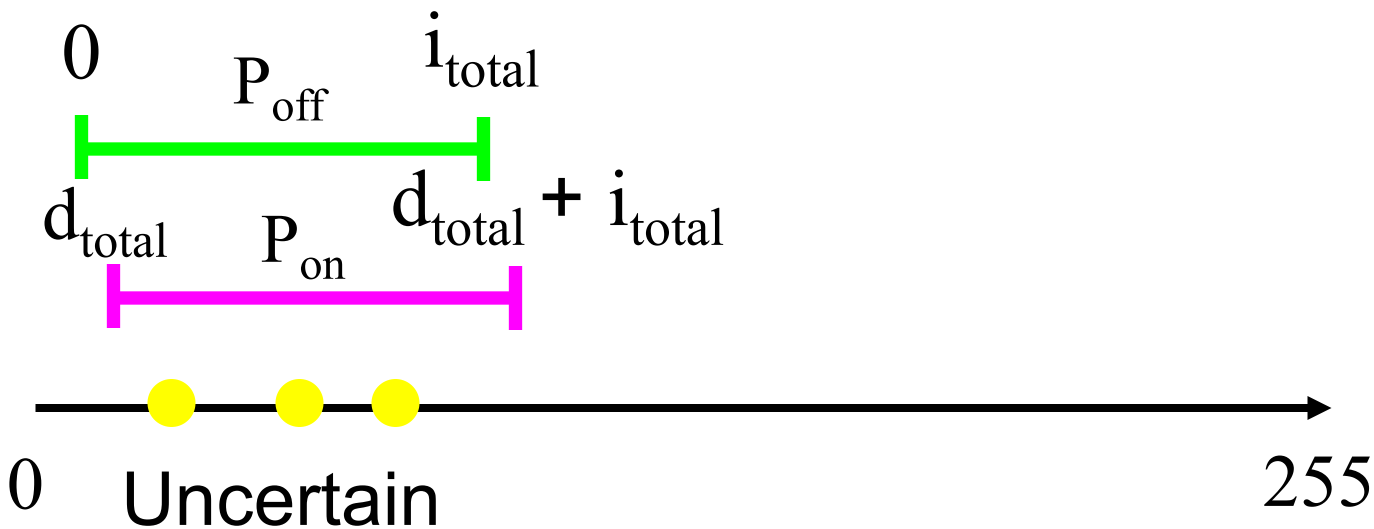
$d_{total} > i_{total}$



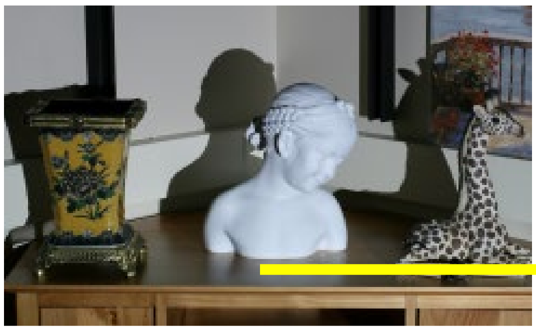
Pixel Classification Scenarios



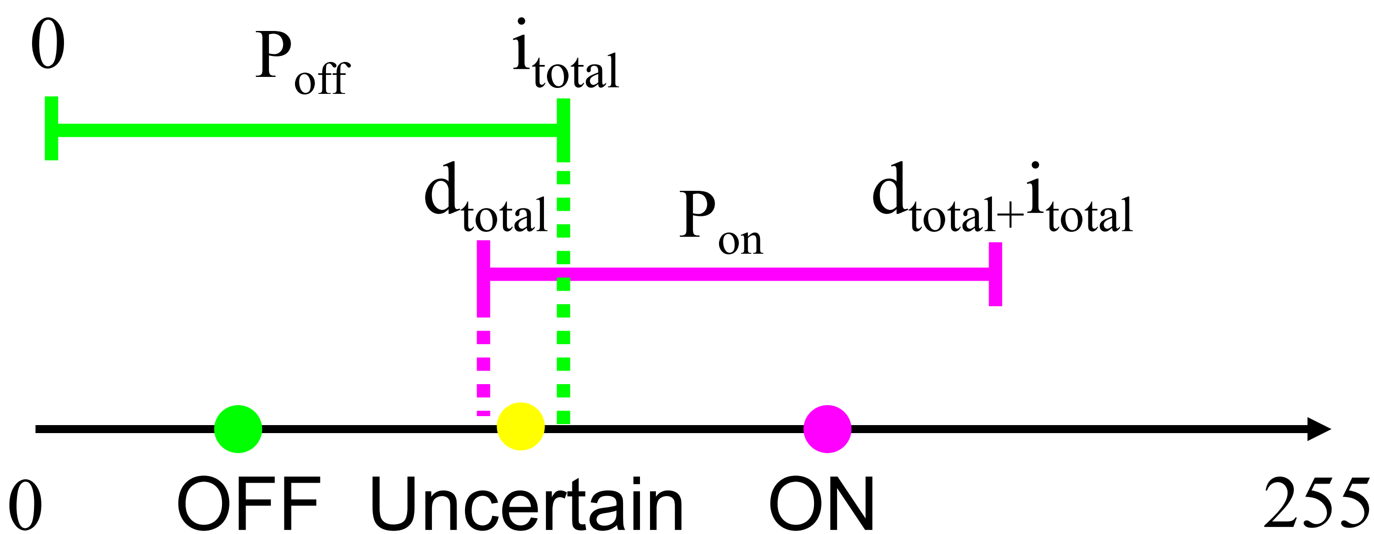
$d_{total} \approx 0$



Pixel Classification Scenarios



$$d_{total} \le i_{total}$$

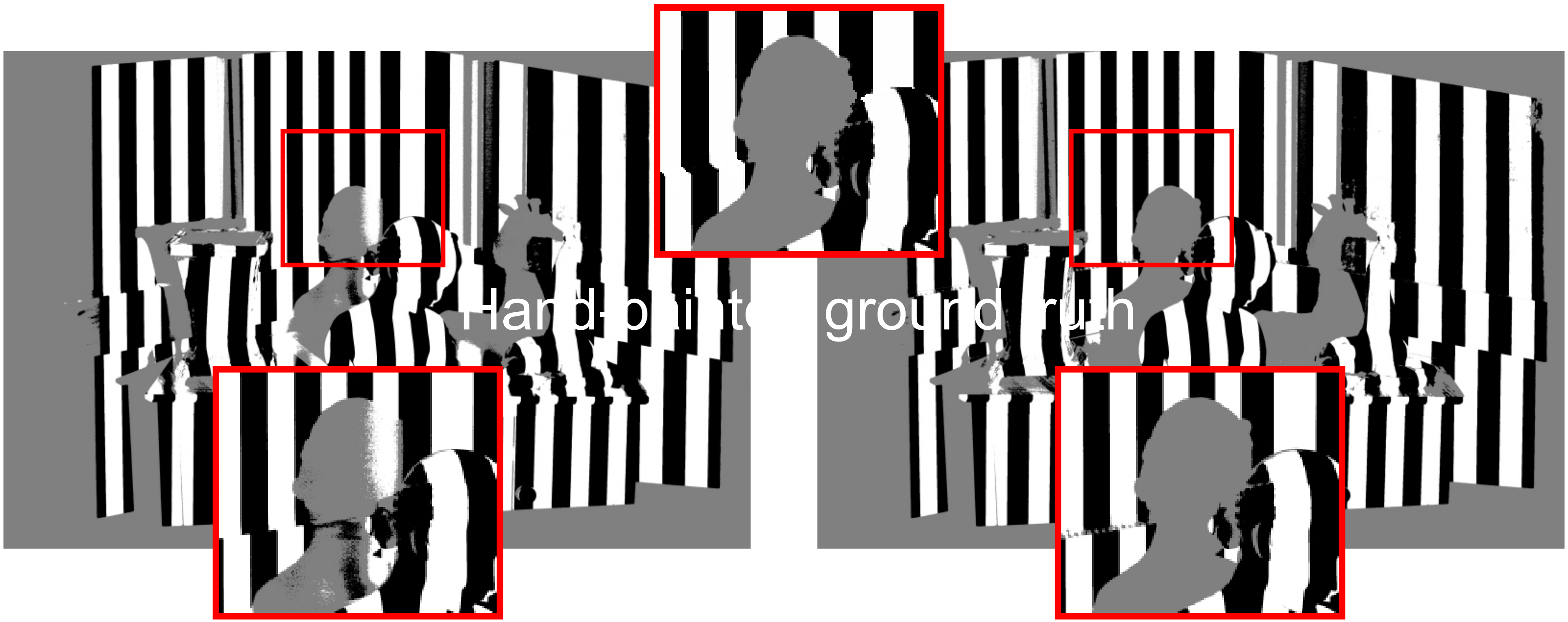


Single Pattern Classification Rules

- $\boldsymbol{d}_{\textbf{total}} < \boldsymbol{m}$ → pixel is *uncertain*
- $\boldsymbol{p} < \min(\boldsymbol{d}_{\boldsymbol{total}}, \boldsymbol{i}_{\boldsymbol{total}})$ → pixel is *off*
- $\boldsymbol{p} > \max(\boldsymbol{d}_{\boldsymbol{total}}, \boldsymbol{i}_{\boldsymbol{total}})$ → pixel is *on*
- otherwise → pixel is *uncertain*

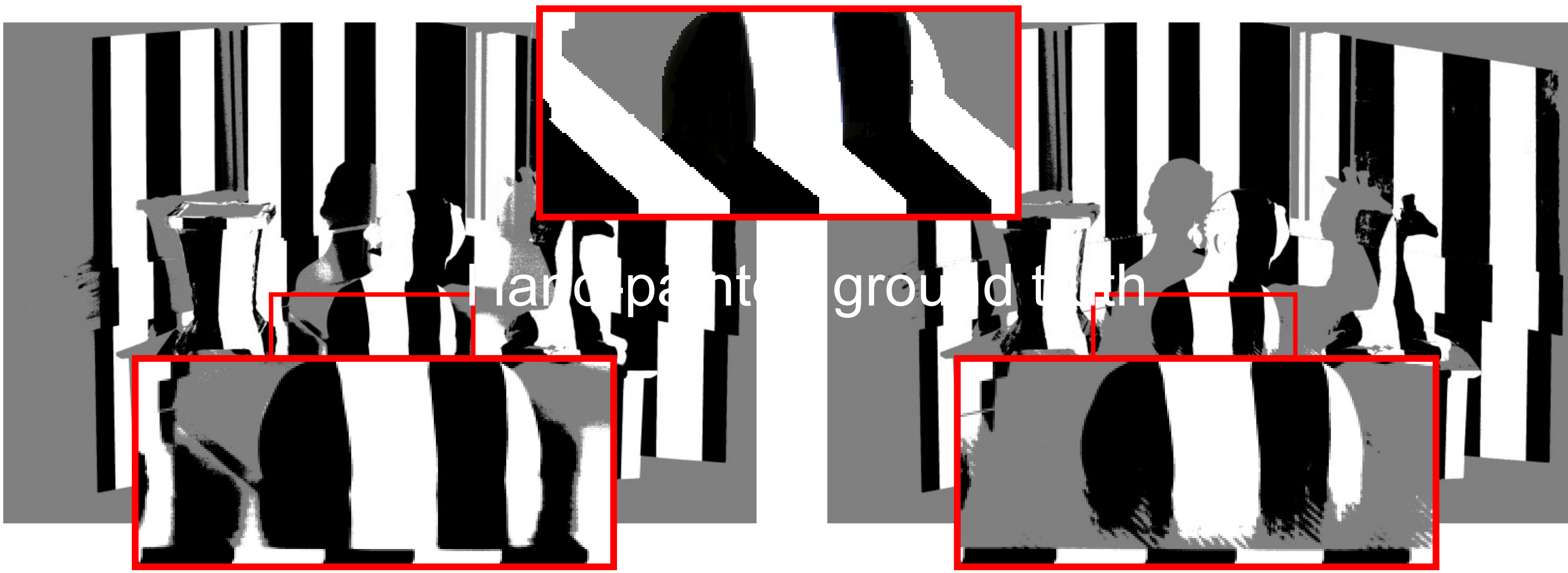
Classification Results

ON(1) OFF(0) Uncertain

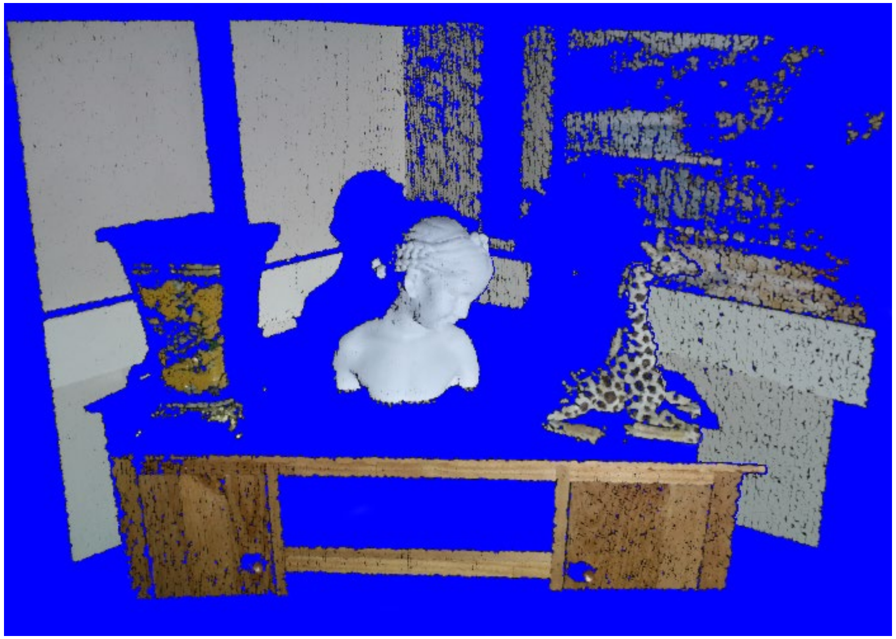


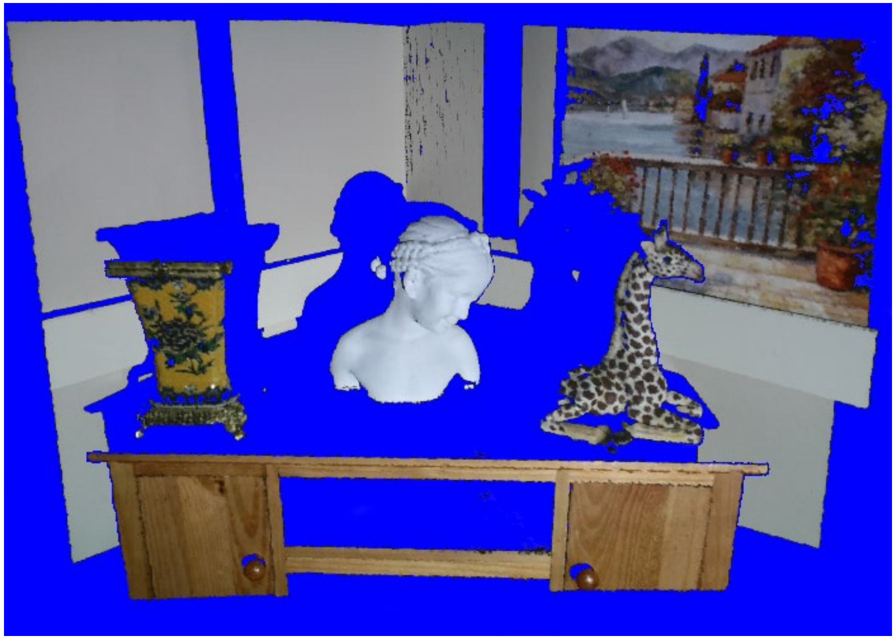
Classification Results

ON(1) OFF(0) Uncertain

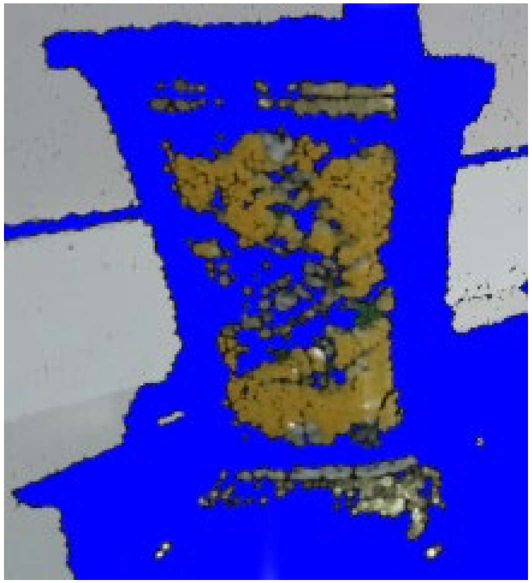


Increased Reconstructed Points

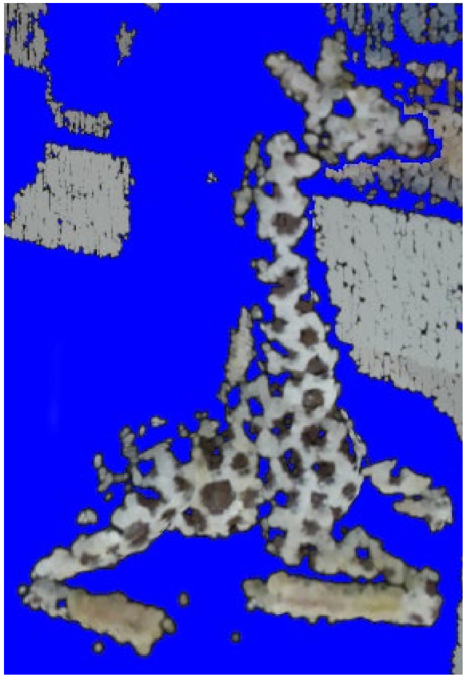




Increased Reconstructed Points

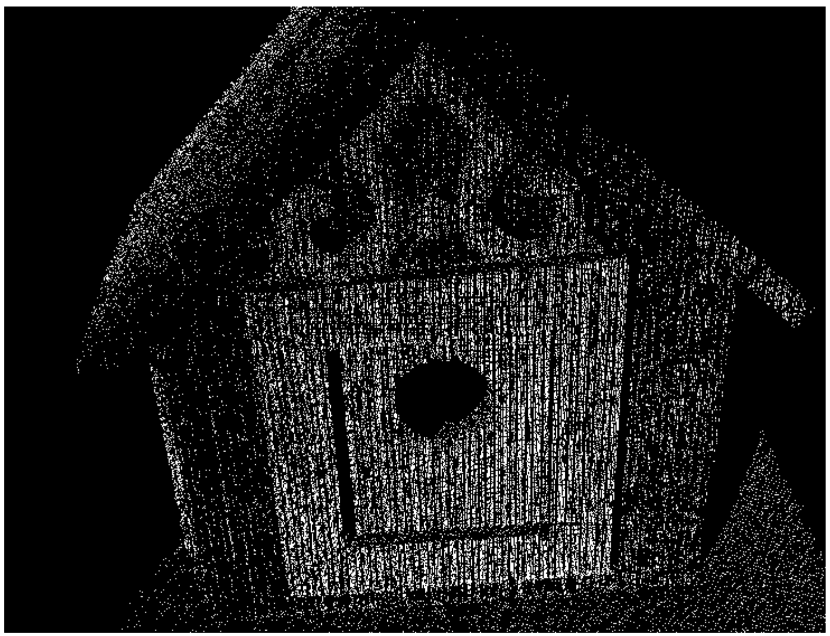


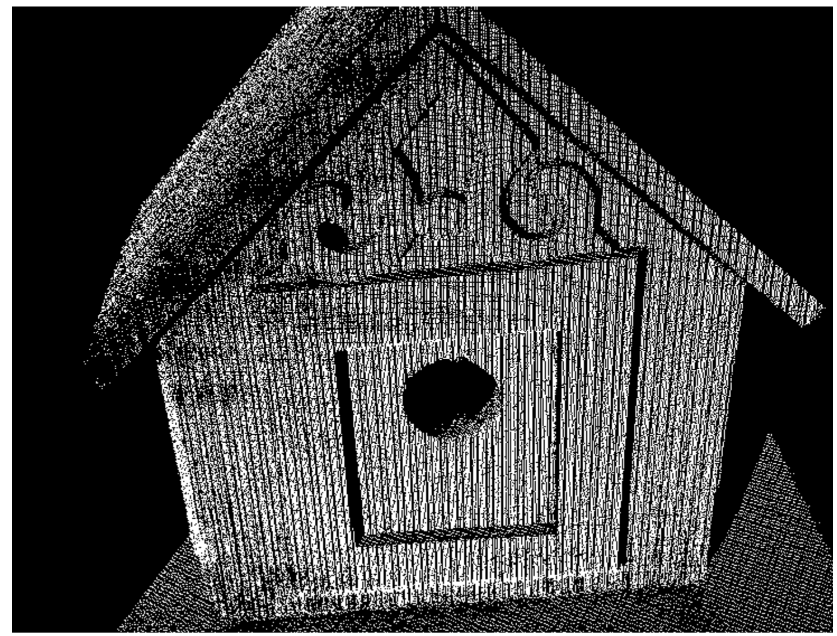
Increased Reconstructed Points



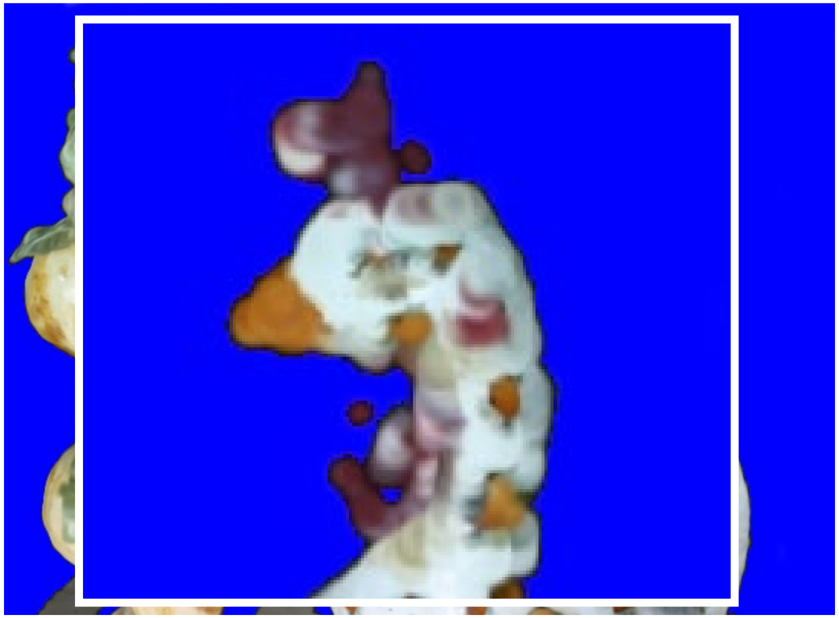
Increased Reconstructed Points

Increased Reconstructed Points





Increased Reconstructed Points



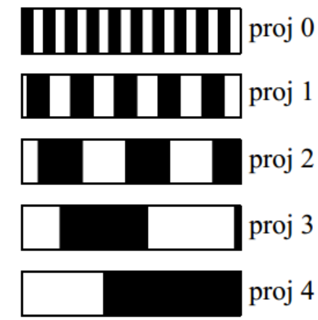
Another issue…

Another issue…

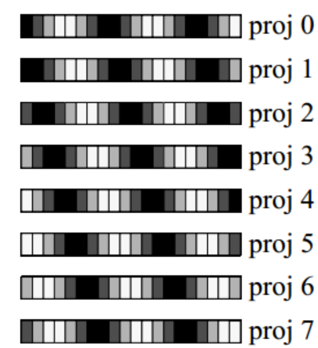
- Classify…
  - pixels at stripe boundaries?
  - pixels at strip middle?
  - all pixels?

Structured Light using Phase shifting

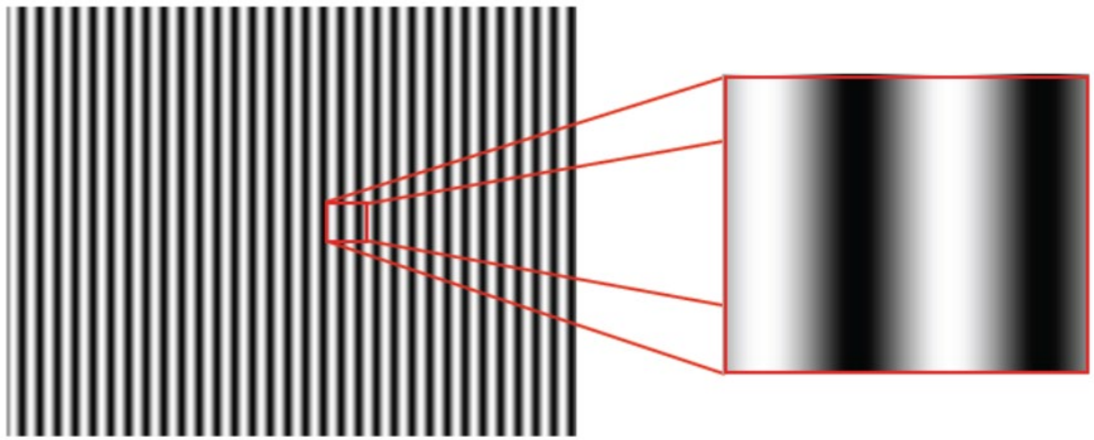
- Binary/gray code structured light
  - limited to (integer) projector resolution
  - has trouble with the LSB bits



- Alternative is **phase shifting**
  - Able to produce subpixel accuracies
  - Able to capture LSB bits more robustly



Phase Shifting



- Project 3 or more "phase shifted" sinusoidal patterns
- For example:

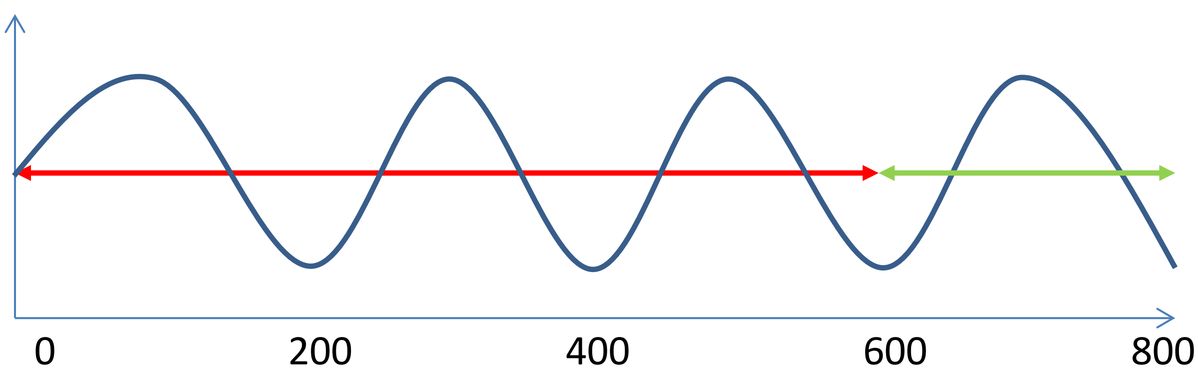
$$I_- = I_b + I_v \cos(\phi - \theta)$$
$$I_* = I_b + I_v \cos(\phi)$$
$$I_+ = I_b + I_v \cos(\phi + \theta)$$

- The variable ϕ is the phase of each column

Phase Shifting

- Since binary/gray code good for MSB (and phase shifting less reliably for large ranges), we define absolute and relative phase shifts:

$$\phi'(x,y) = \underbrace{2\pi k(x,y)}_{\text{(from binary codes)}} + \underbrace{\phi(x,y)}_{\text{(relative phase shift)}} \text{ where } k = [0, N-1]$$



Phase Shifting

$$I_- = I_b + I_v \cos(\phi - \theta)$$
$$I_* = I_b + I_v \cos(\phi)$$
$$I_+ = I_b + I_v \cos(\phi + \theta)$$

- How do you remove dependence on I_b, I_v?
- One option: use 3 images and compute

$$\frac{I_- - I_+}{2I_* - I_- - I_+}$$

Phase Shifting

$$\frac{I_- - I_+}{2I_* - I_- - I_+}$$

$$= \frac{2\sin(\phi)\sin(\theta)}{2\cos(\phi)(1-\cos(\theta))} = \frac{\tan(\phi)}{\tan(\frac{\theta}{2})}$$

thus

$$\phi(0,2\pi) = \arctan(\tan(\frac{\theta}{2})\frac{I_- - I_+}{2I_* - I_- - I_+})$$

Phase-shift projection

- Phase angle from brightness
  - Computed from the three images
  - Ambient light and projector power can be ignored
  - Result tells you location within one period…thus "micro-correspondence" and not "macro correspondence"
  - Empirically, good for last few bits:
    - E.g., to get 10 bits of correspondence, use gray code for 7 msb and phase shifting for 3 lsb
    - Total images: 2 x 7 x 2 + 3 x 2 = 34 instead of 2 x 10 x 2 = 40 (but also maybe more robust)