Image Formation II

Semester 2, 2025

Kris Ehinger

Outline

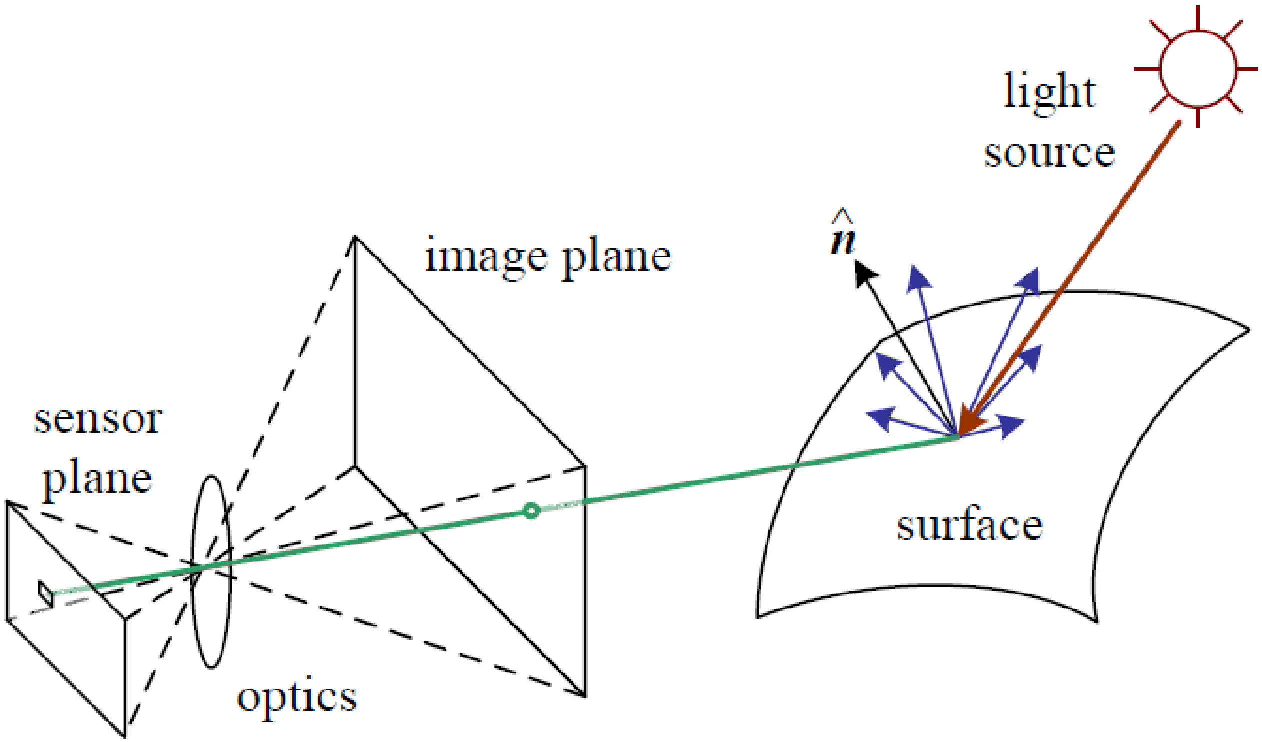
- Image formation, continued
- Colour
- Shading and surfaces
- Feature invariance

Learning outcomes

- Implement the diffuse reflectance model
- Explain how trichromatic colour values are computed and implement colour space transforms
- Explain the problems involved in recreating surface properties from a single image
- Explain “invariance” in the context of object recognition and identify invariant features

Image formation

Image formation model



Computer Vision, Richard Szeliski (2021)

Diffuse (Lambertian) reflectance

$$I_D(x) = I_L R \mathbf{N}(x) \cdot \mathbf{L}$$

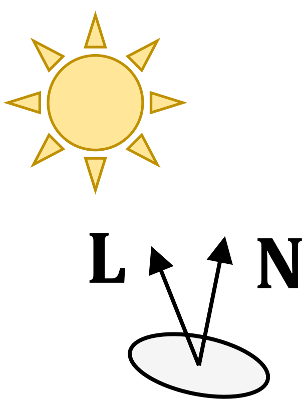
Intensity of reflected light

Light source intensity

Reflectance (Colour)

Surface's normal vector

Vector to light source



Goal of vision

$$I_D(x) = I_L R \mathbf{N}(x) \cdot \mathbf{L}$$

Intensity of reflected light

Surface's normal vector

Reflectance (Colour)

Recover surface colour and normal from reflected light

Colour

Visible light

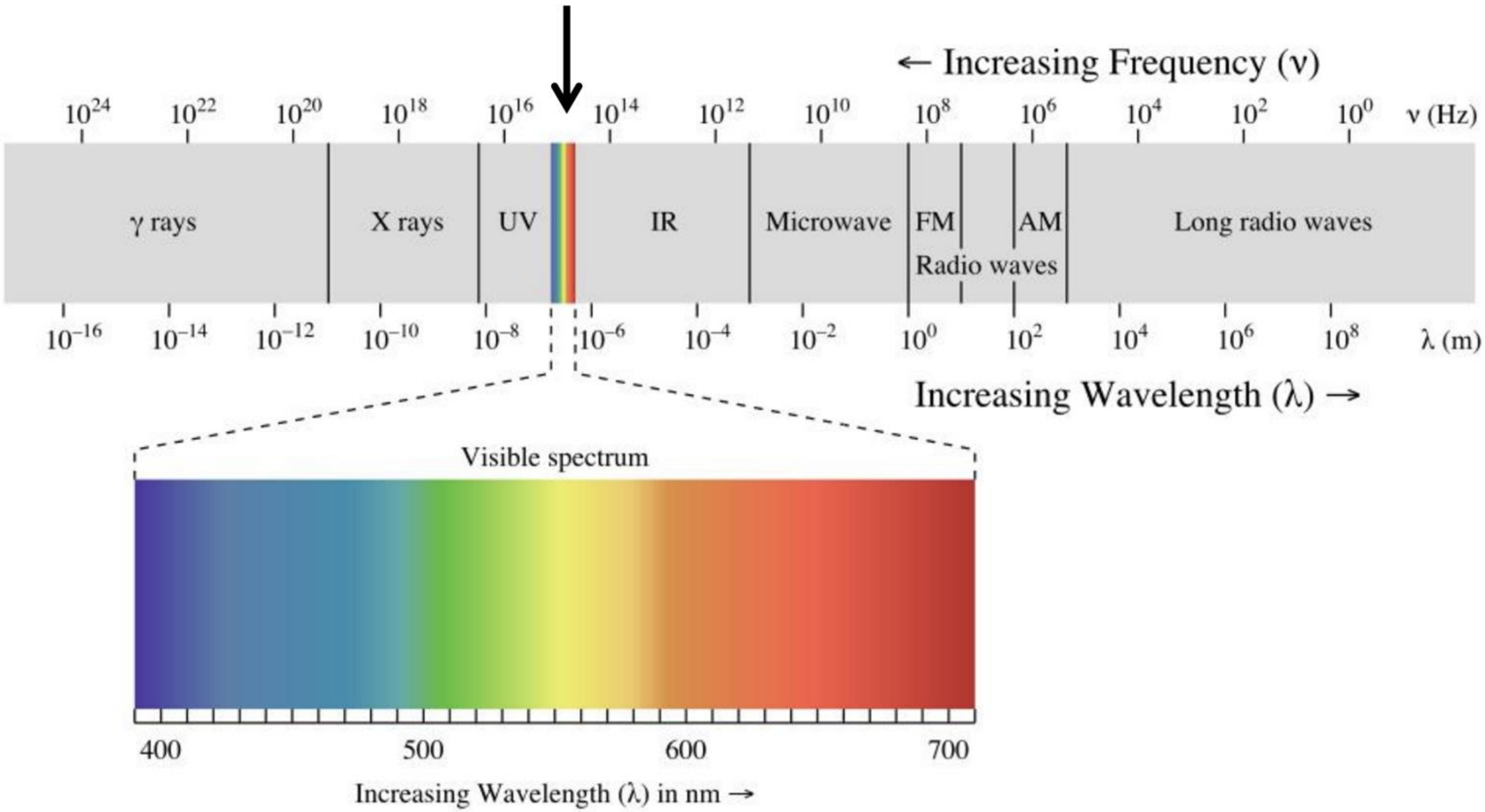
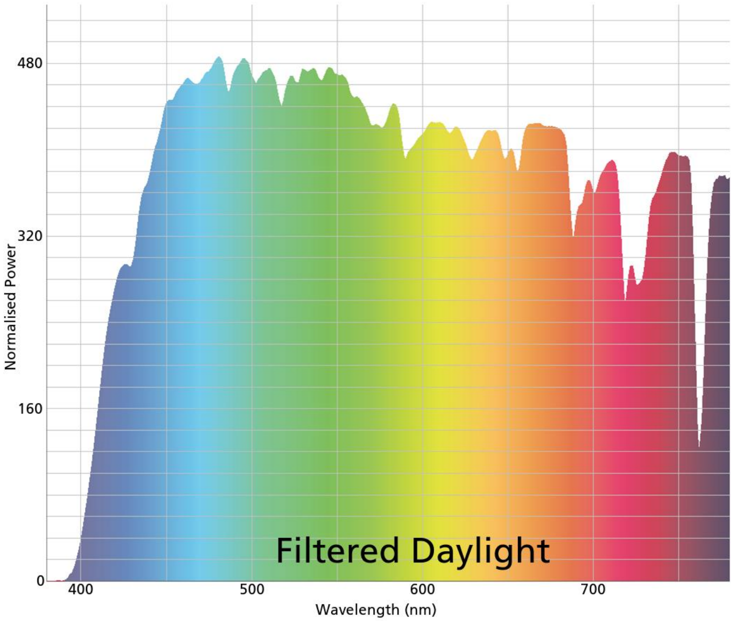


Figure: M. Brown

Visible light

- Spectral power distribution (SPD) = relative amount of each wavelength reflected by a surface (or produced by a light source)



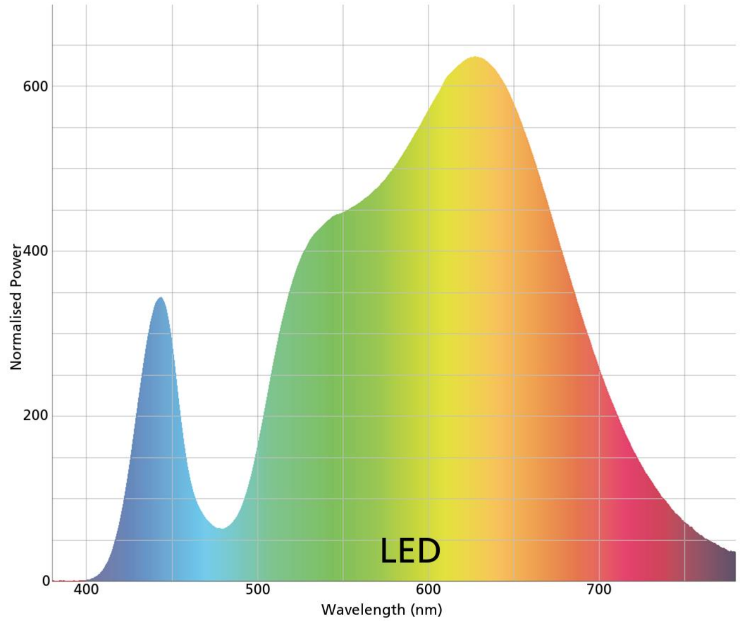
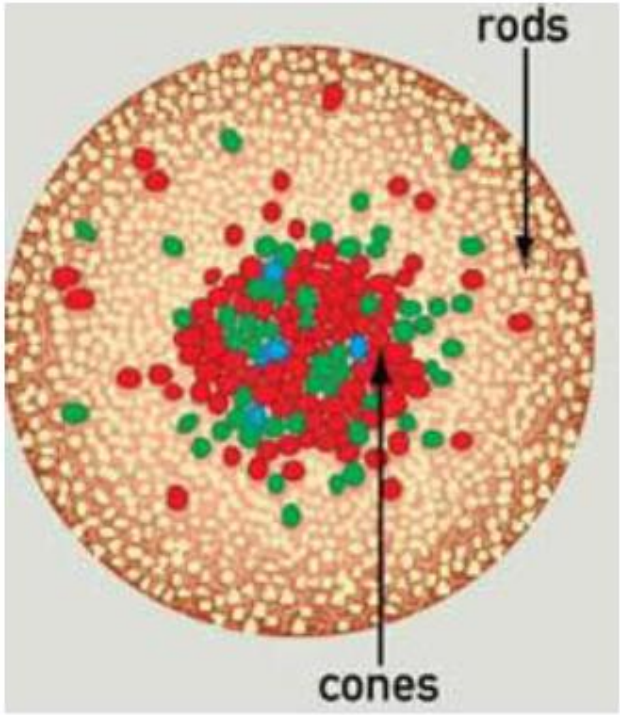


Figure: https://research.ng-london.org.uk/scientific/spd/

Perceived colour

- Human colour perception is based on 3 types of colour-sensitive cells (cones)



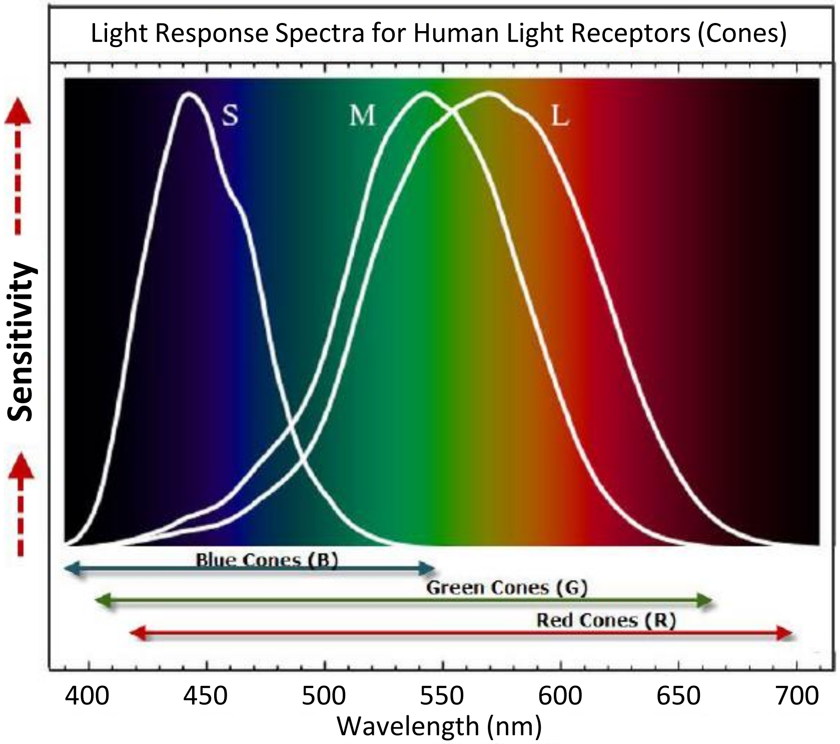
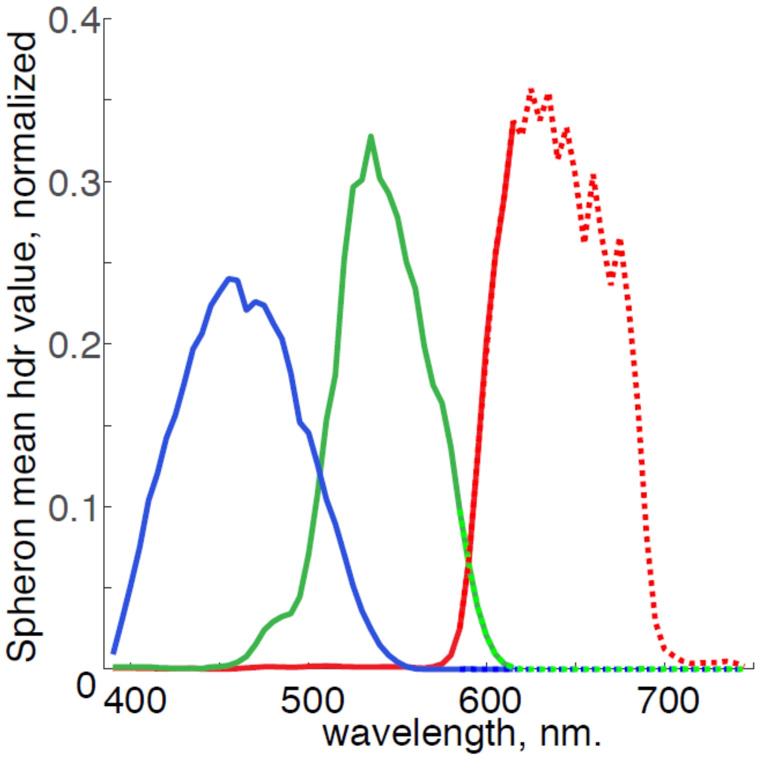


Figure: M. Brown

Perceived colour

- Standard cameras also have 3 colour sensors, each with a different spectral sensitivity

Spheron SpheroCam HDR



Adams, et al. (2016), https://syns.soton.ac.uk/

Perceived colour

- Most surfaces reflect a range of wavelengths, but perceived colour is a function of cone response

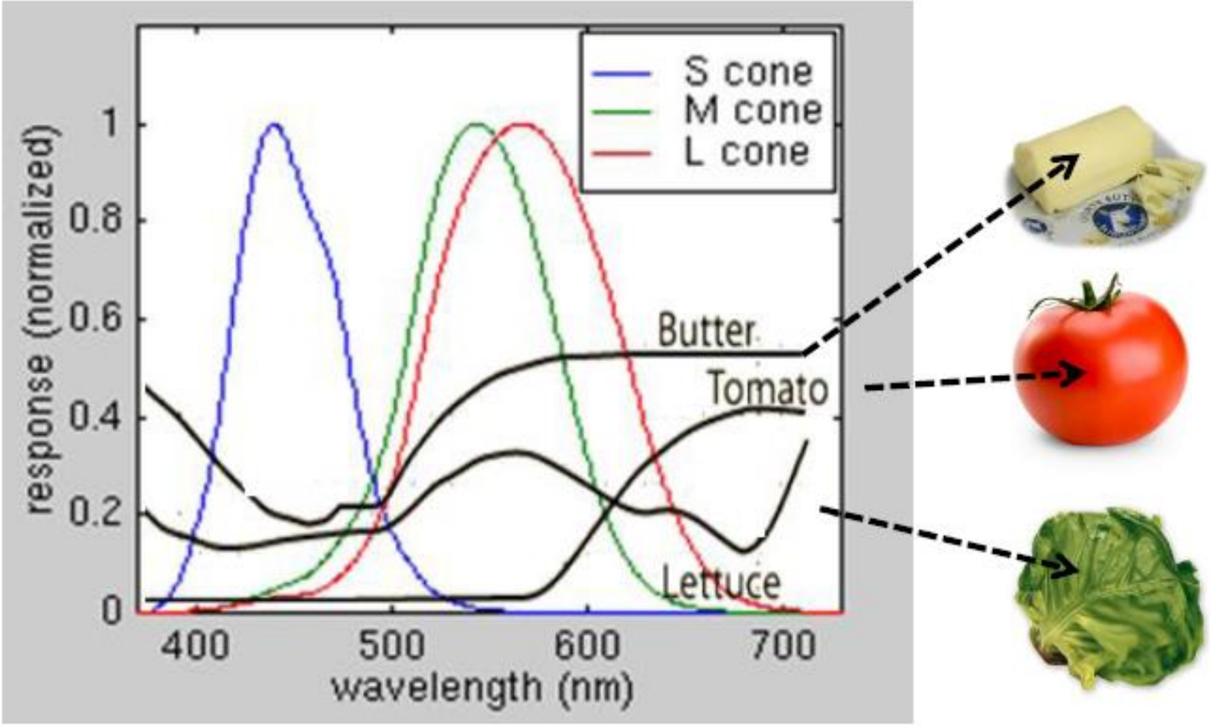


Figure: M. Brown

Perceived colour

- Result: many different spectra appear to be the same colour

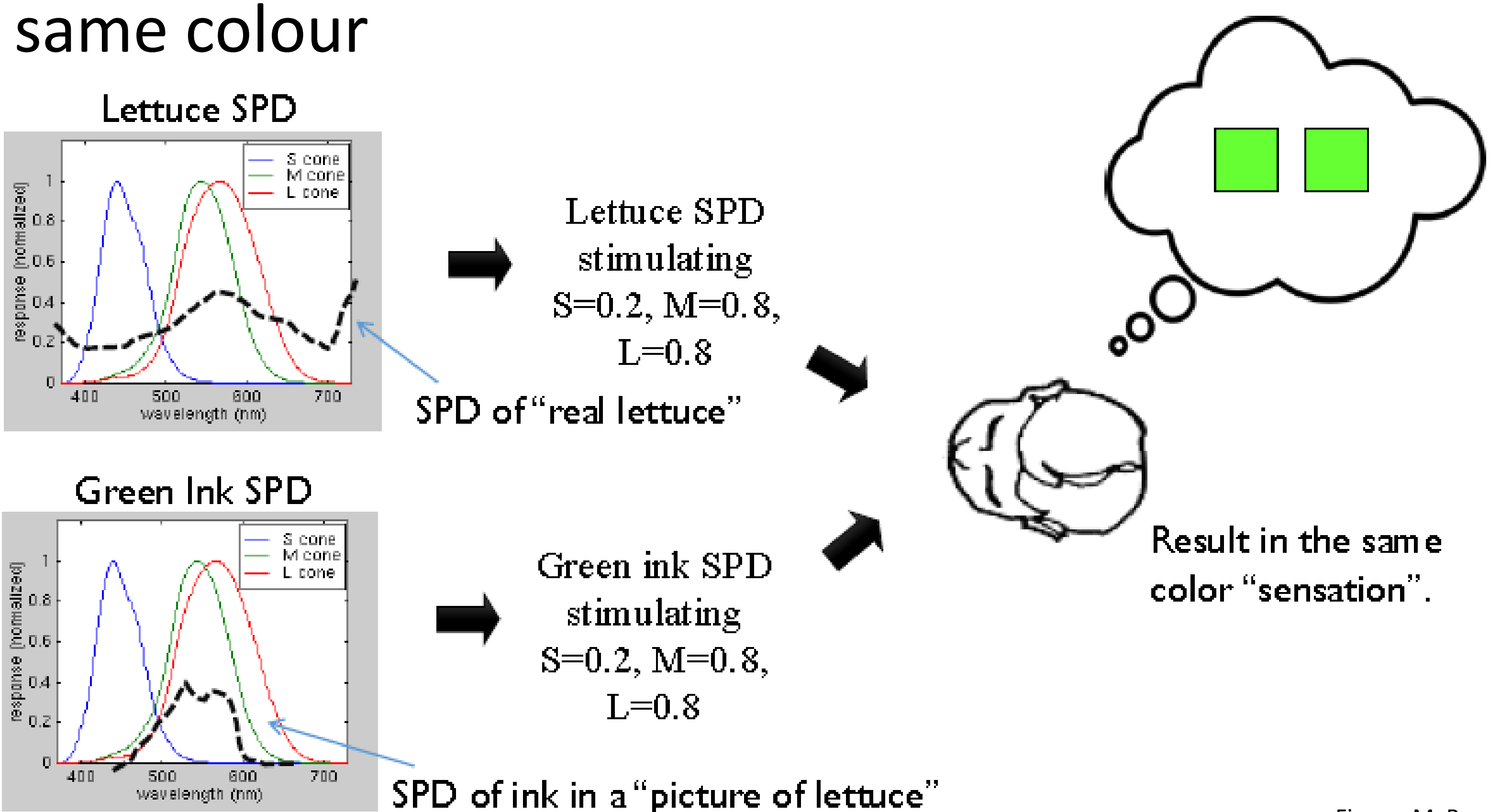


Figure: M. Brown

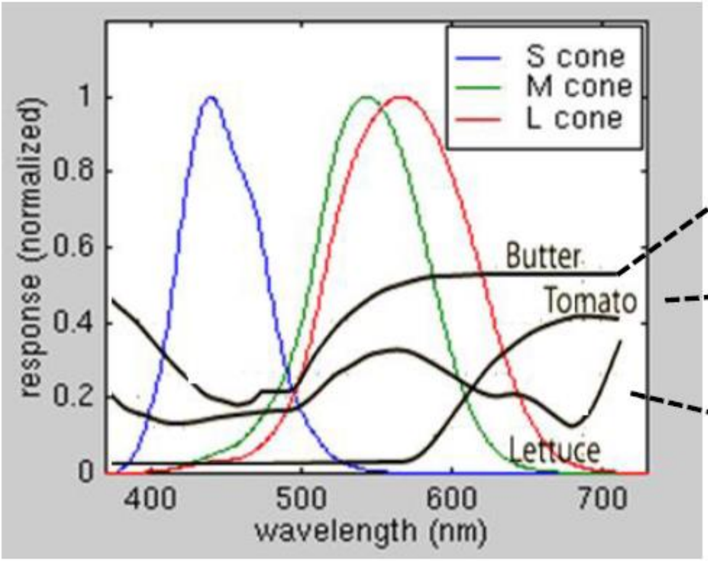
Trichromatic colour response

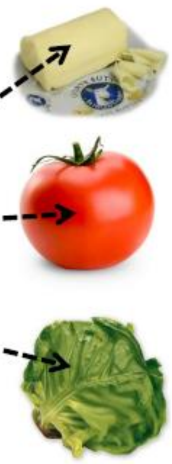
- Sensor response = sensitivity x spectrum, integrated over all wavelengths

$$I_R = \int_{400}^{700} I(\lambda) S_R(\lambda) \partial\lambda$$

$$I_G = \int_{400}^{700} I(\lambda) S_G(\lambda) \partial\lambda$$

$$I_B = \int_{400}^{700} I(\lambda) S_B(\lambda) \partial\lambda$$



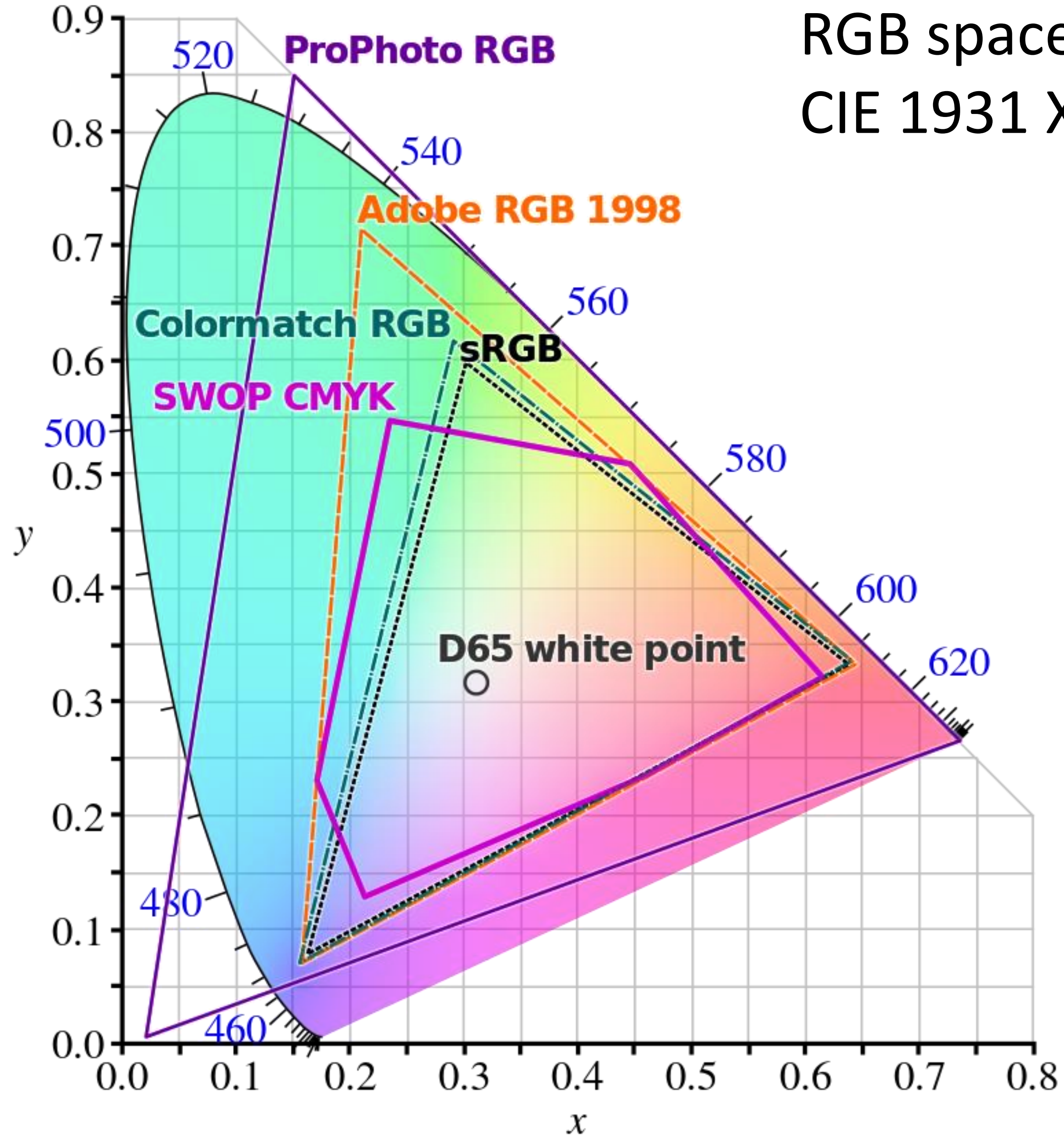


https://csci1230.graphics/demos/metamers/index.html

Colour representation

- Common colour spaces:
- RGB (red, green, blue)
  - Most common spaces for digital images
- HSL/HSV (hue, saturation, lightness/value)
  - Attempt to match human understanding of colour
- CIE 1931 XYZ
  - Based on human cone sensitivity, basis for other spaces
- LAB (luminance, a*=red/green, b*=blue/yellow)
  - Approximately perceptually uniform space

RGB spaces, plotted in CIE 1931 XYZ space



Colour transforms

- Converting between colour spaces is straightforward:
  - Linearize R, G, B values
  - Linear transform, e.g. $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} R \\ G \\ B \end{bmatrix}$
  - Values of M can be looked up for various colour spaces and white points (= the value defined to be "white" for a given colour space)
- Built-in functions in OpenCV, scikit-image

Example: colour swap

Swap R,G channels in RGB

Invert red-green axis in LAB

Summary

- Colour is not just three values, but human eye (and standard camera) depends on just three sensors
- Many trichromatic colour spaces
- RGB most common for image storage, other spaces may be more useful for colour manipulations

Shading and surfaces

Goal of vision

$$I_D(x) = I_L R \mathbf{N}(x) \cdot \mathbf{L}$$

Intensity of reflected light

Surface's normal vector

Reflectance (Colour)

Recover surface colour and normal from reflected light

Recovering surface properties

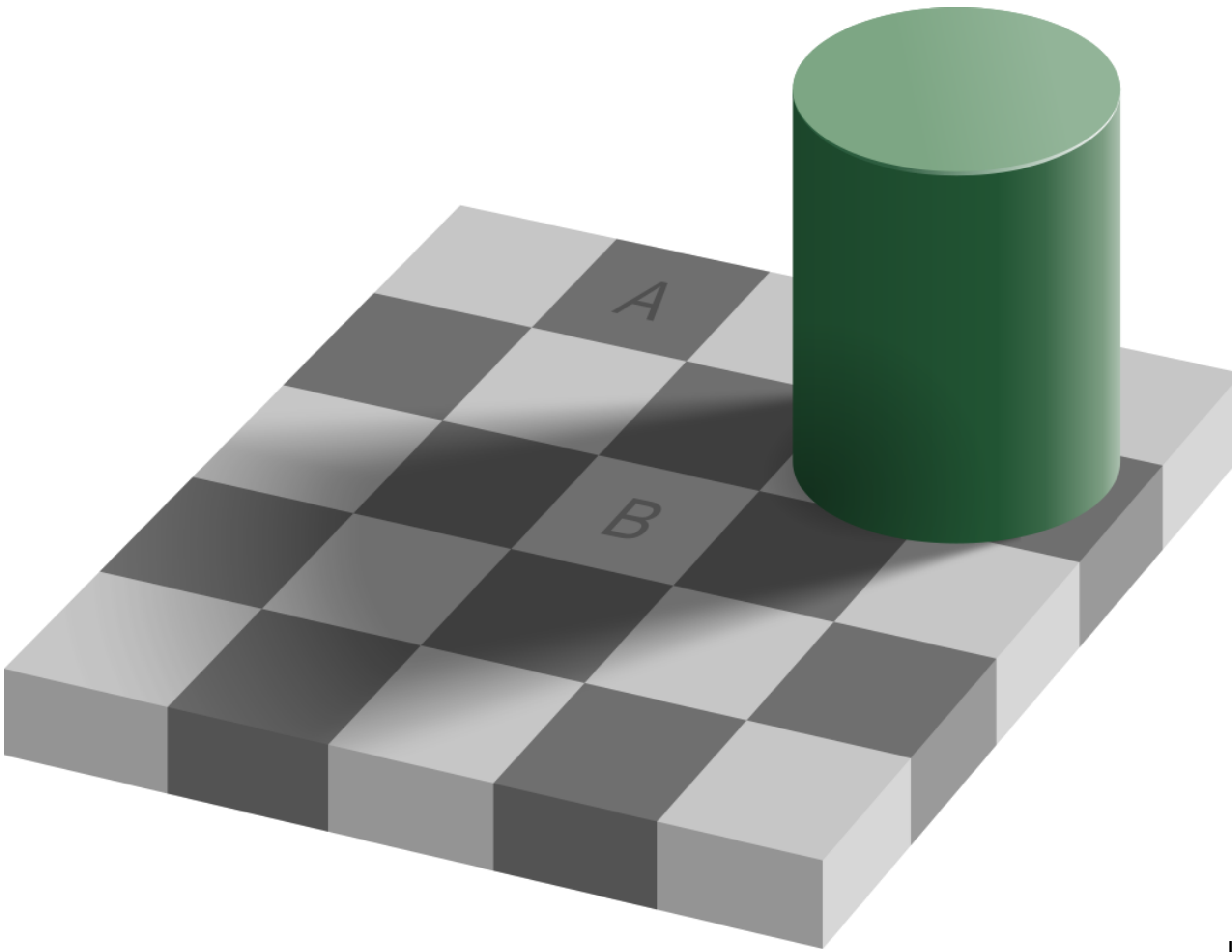


Image: E. Adelson

Recovering surface properties

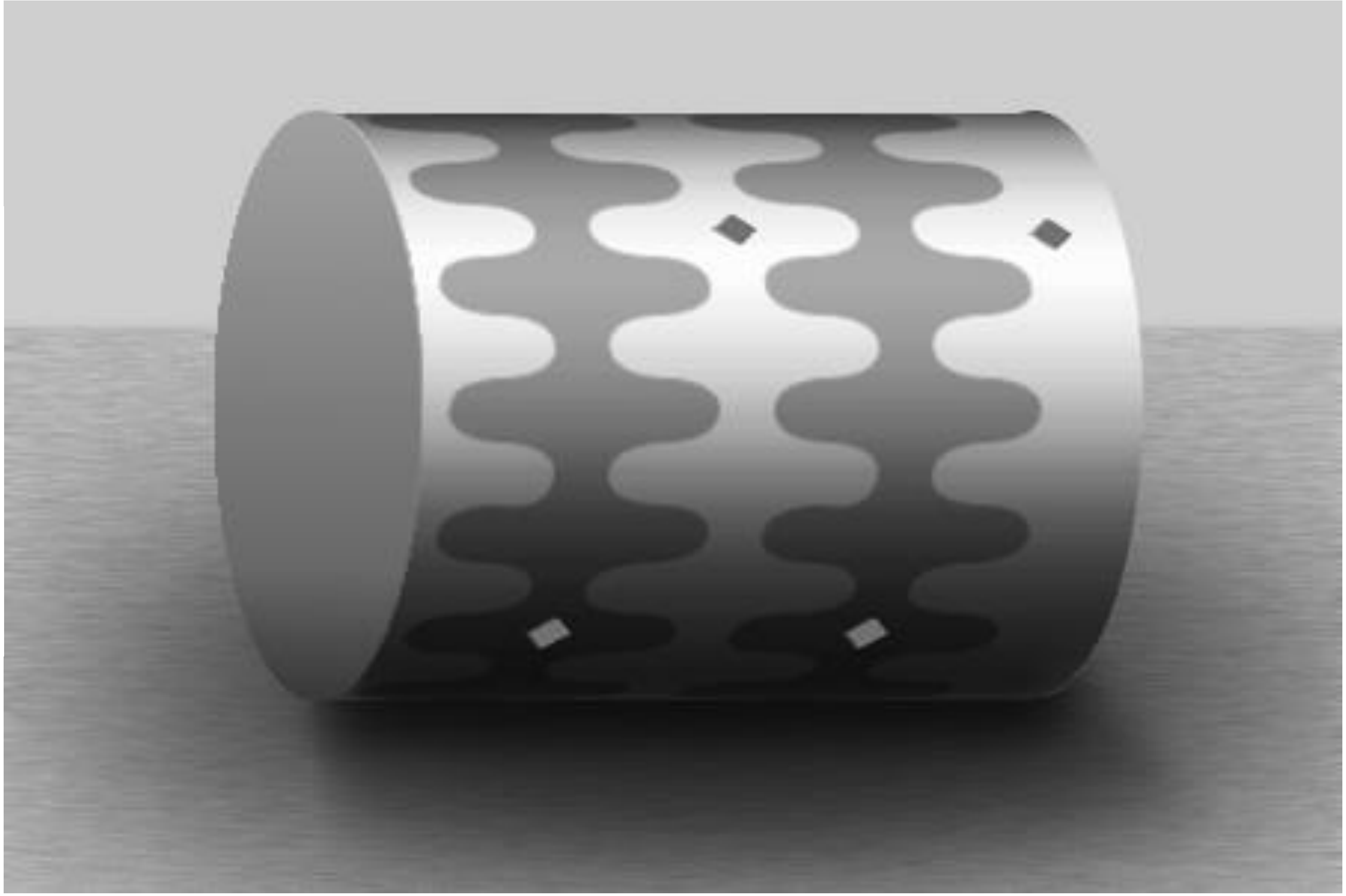
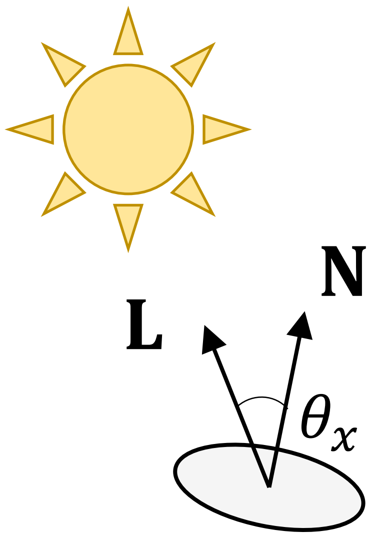


Image: E. Adelson

Recovering surface normal

- Assume no changes in surface colour/reflectance (constant albedo)
- Can you recover surface normal from image?
- $I_D(x) = \mathbf{N}(x) \cdot \mathbf{L} = \cos\theta_x$



Recovering surface normal

- Can recover *angle* between surface normal and light source, but not normal
- However, can add additional assumptions:
  - Normals along boundary of object are known
  - Neighbouring normal are similar

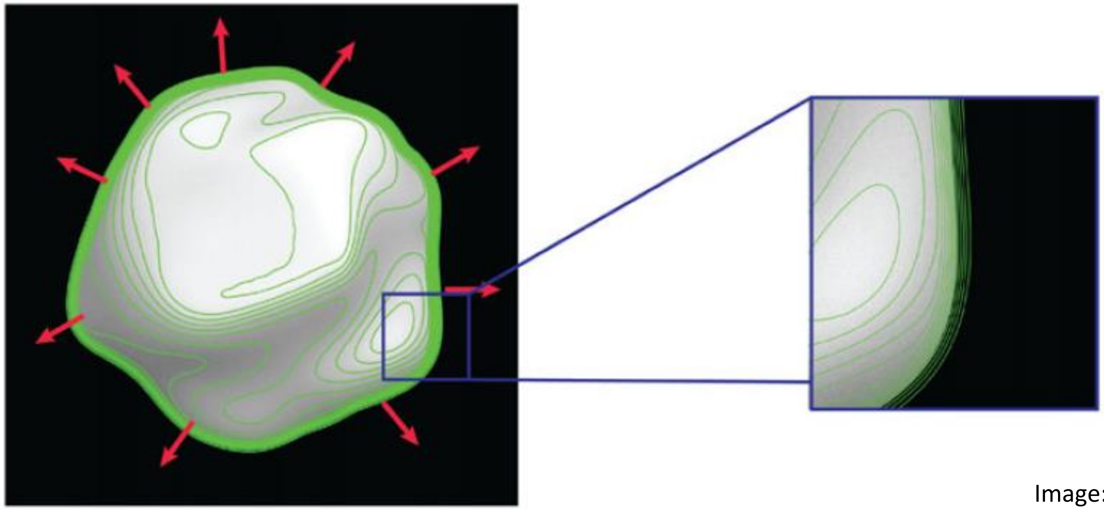


Image: Kunsberg, et al. (2018)

“Shape from shading”

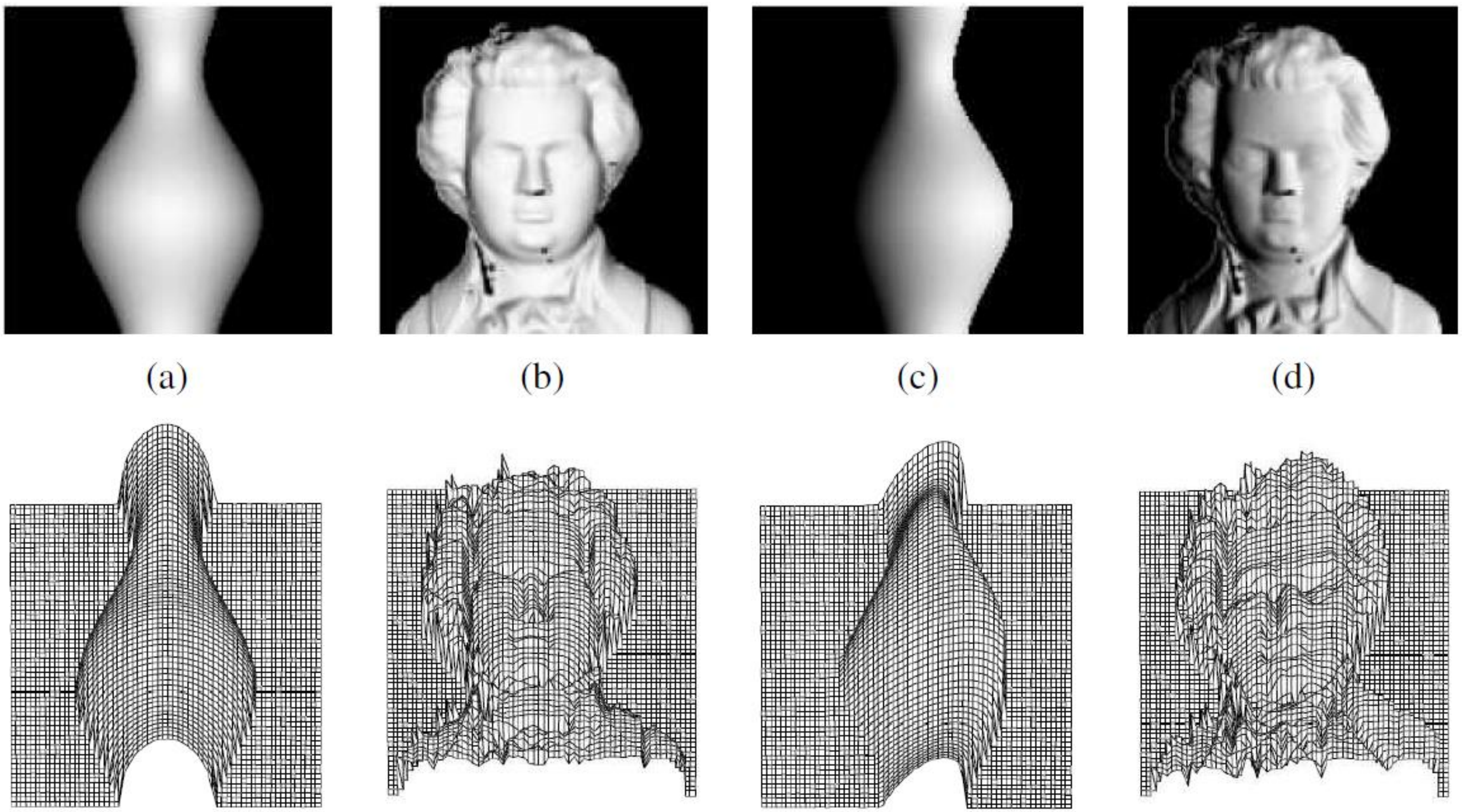


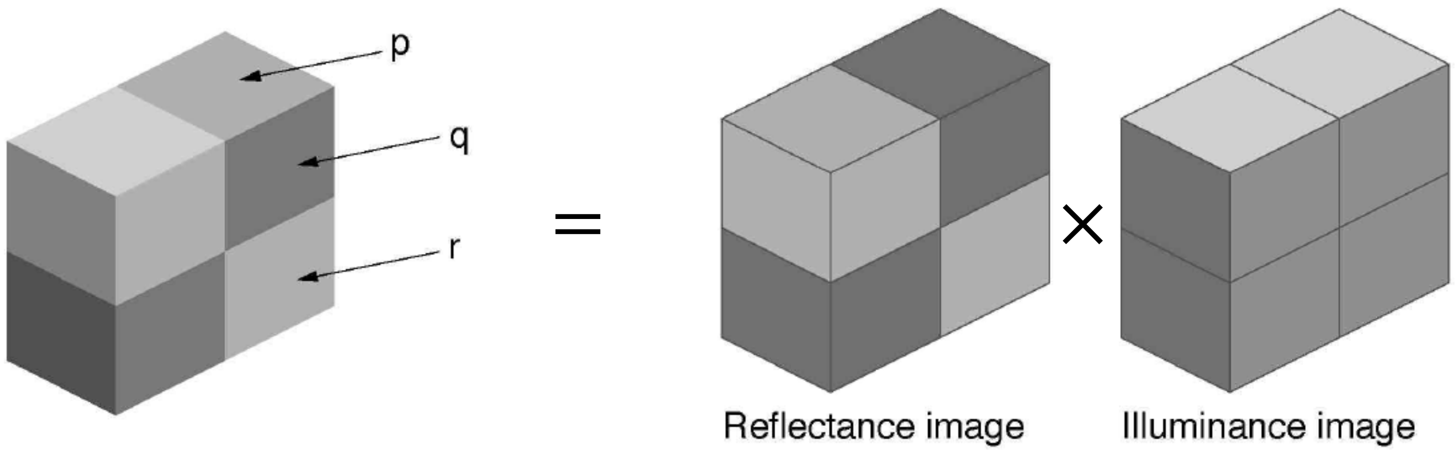
Image: R. Szeliski, *Computer Vision*, Figure 13.2

“Shape from shading”

- Recover 3D shape from 2D image based only on surface brightness (shading)
- Requires additional assumptions, no algorithm works for all cases

- What if surface isn’t constant albedo?

Recovering surface reflectance



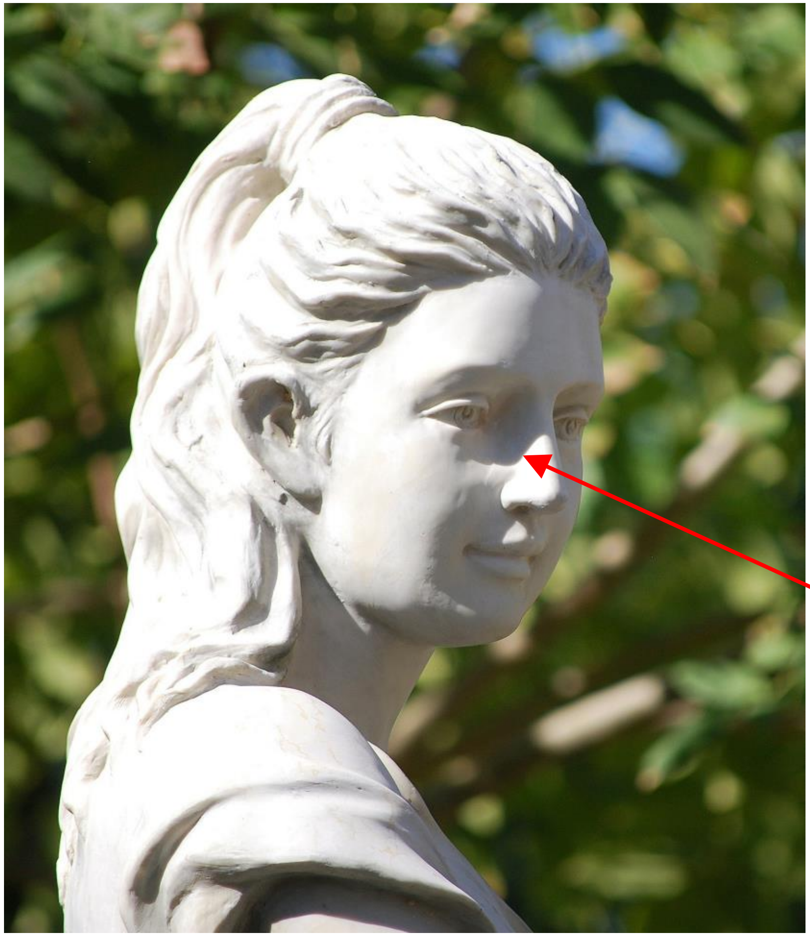
$$Luminance = Reflectance \times Illumination$$

Image: E. Adelson (2000)

Recovering surface properties

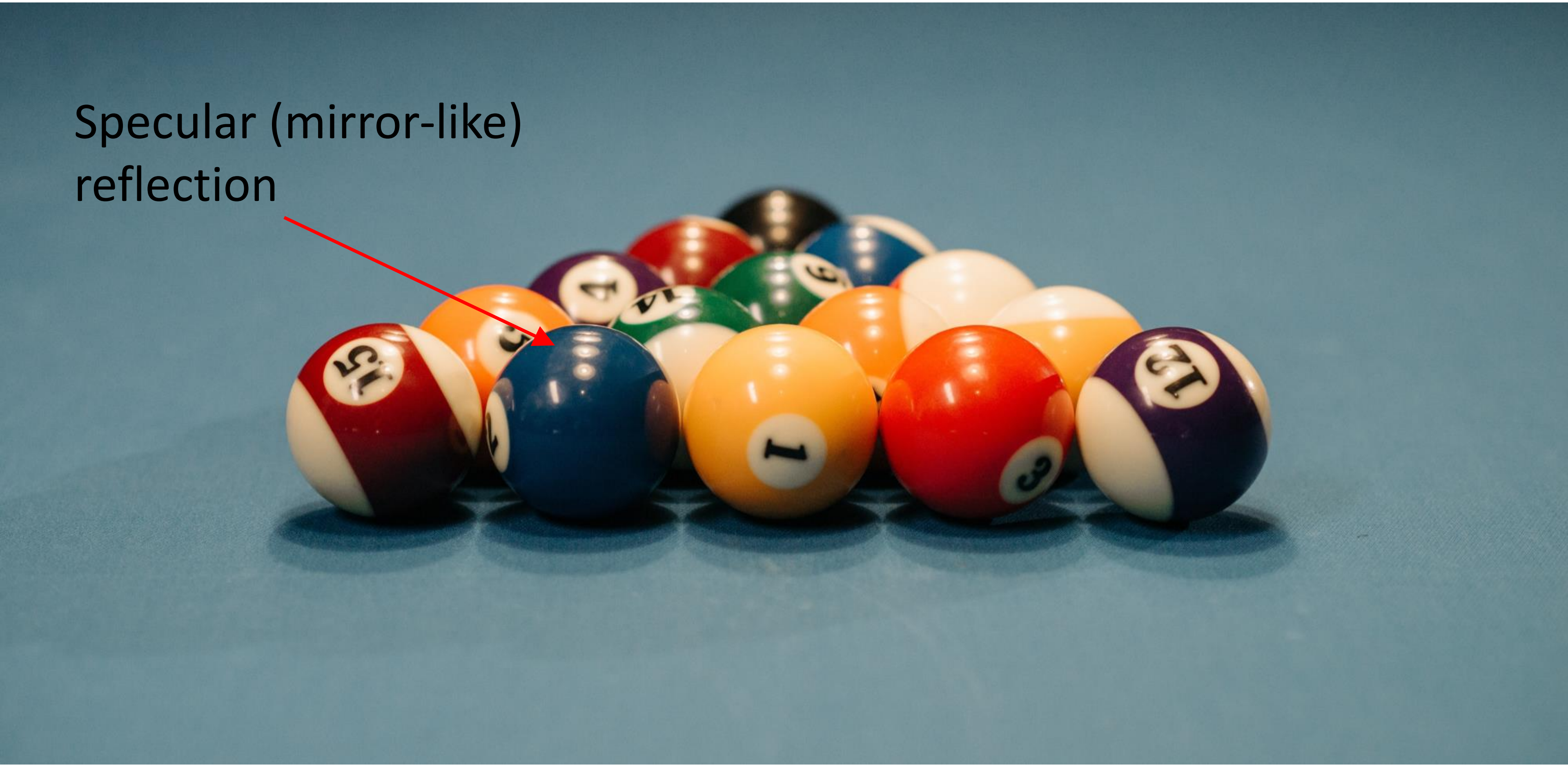
- Simple approach: assume lighting is blurry/smooth and hard edges are always due to reflectance
  - Some reflectance edges are smooth
  - Some lighting edges are not smooth (textures, corners)
- Even more complicated in practice!
  - Lighting usually isn’t uniform
  - Most surfaces aren’t matte/Lambertian

Cast shadows



Cast shadow – change in illumination, not change in surface

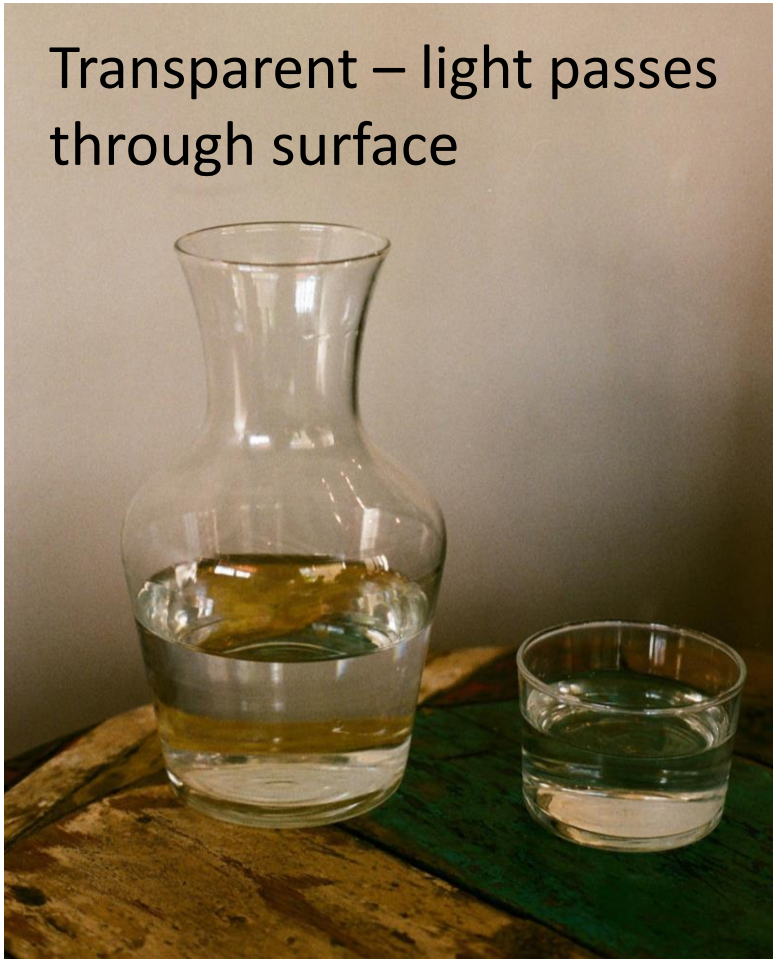
Specularity



Anisotropy

Transparency / translucency

Transparent – light passes through surface



Translucent – light passes through but is scattered

Summary

- Goal of vision: recover object properties (shape, reflectance) from the image
- Problem is underconstrained
- Some solutions, but rely on additional assumptions (e.g., Lambertian surface, smoothness constraint)

Feature invariance

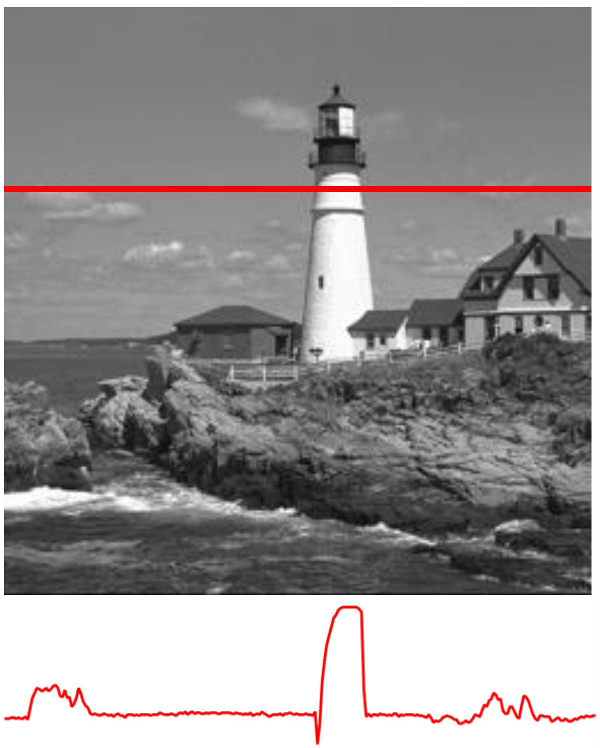
Goal of vision

$$I_D(x) = I_L R\mathbf{N}(x) \cdot \mathbf{L}$$

Not straightforward to recover reflectance (R) or normal (N)

However, **change** in I_D indicates a **change** in R or N (or I_L)

Intensity changes

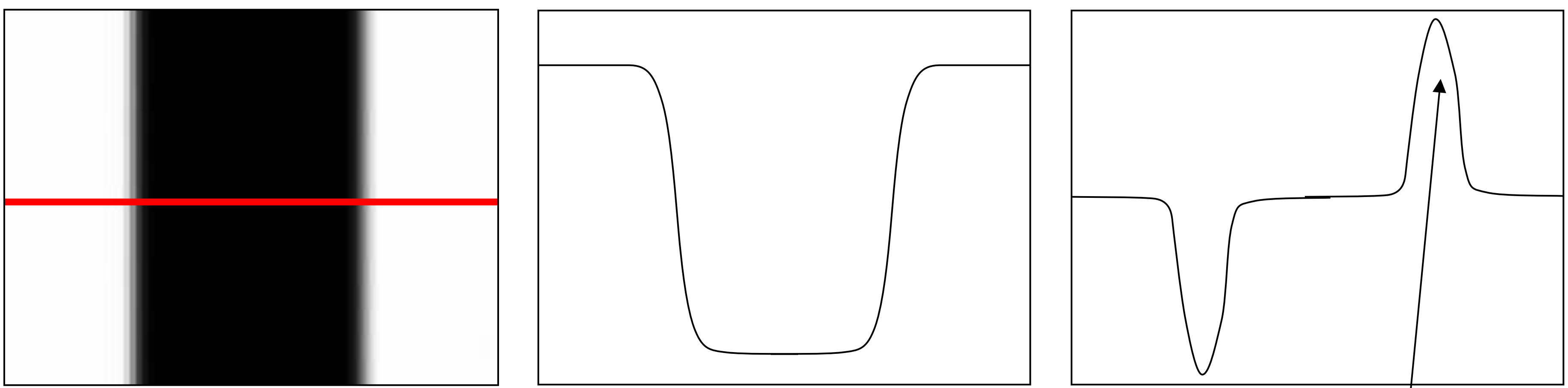


Intensity changes

image

intensity function
(along horizontal scanline)

first derivative



Edge = change in intensity

edges correspond to
extrema of derivative

Figure: D. Hoiem

Gradient

- Gradient of a function over x,y:
- $\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j}$

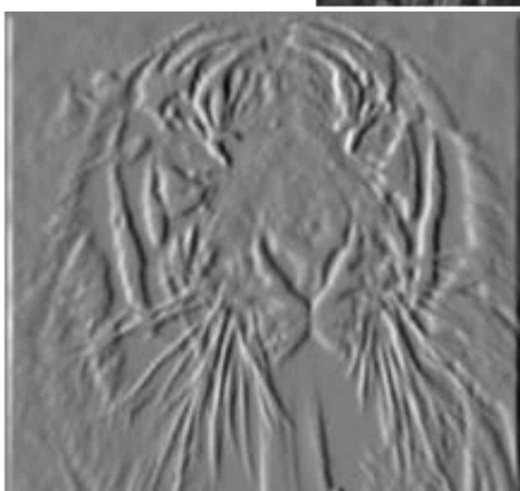
$\mathbf{i}$ = unit vector in the x direction
$\mathbf{j}$ = unit vector in the y direction

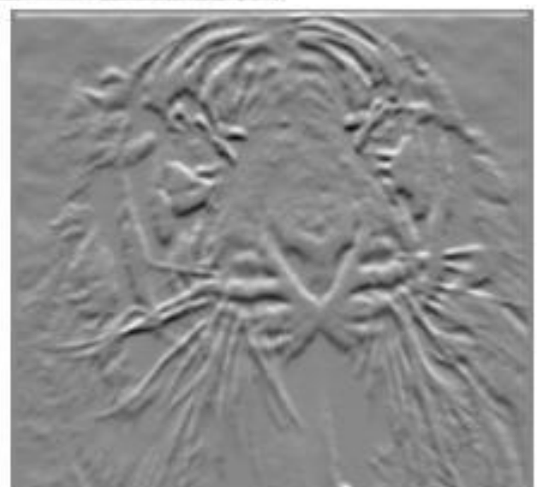
- Gradient at a single point (x,y) is a vector:
  - Direction is the direction of maximum slope:
  - $\theta = \tan^{-1}(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x})$
  - Length is the magnitude (steepness) of the slope
  - $\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$

Partial derivatives in x, y

$$\frac{\partial f(x,y)}{\partial x}$$

$$\frac{\partial f(x,y)}{\partial y}$$





-1	1

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Figure: D. Hoiem

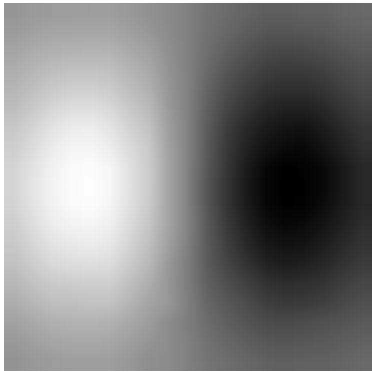
Edge filters

1	0	-1
2	0	-2
1	0	-1

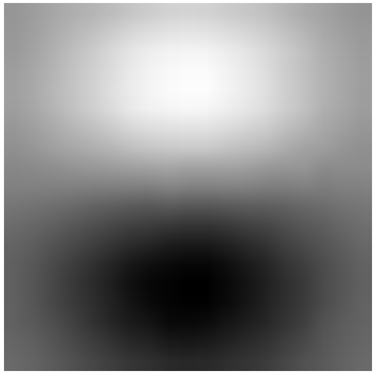
Sobel (x)

1	2	1
0	0	0
-1	-2	-1

Sobel (y)



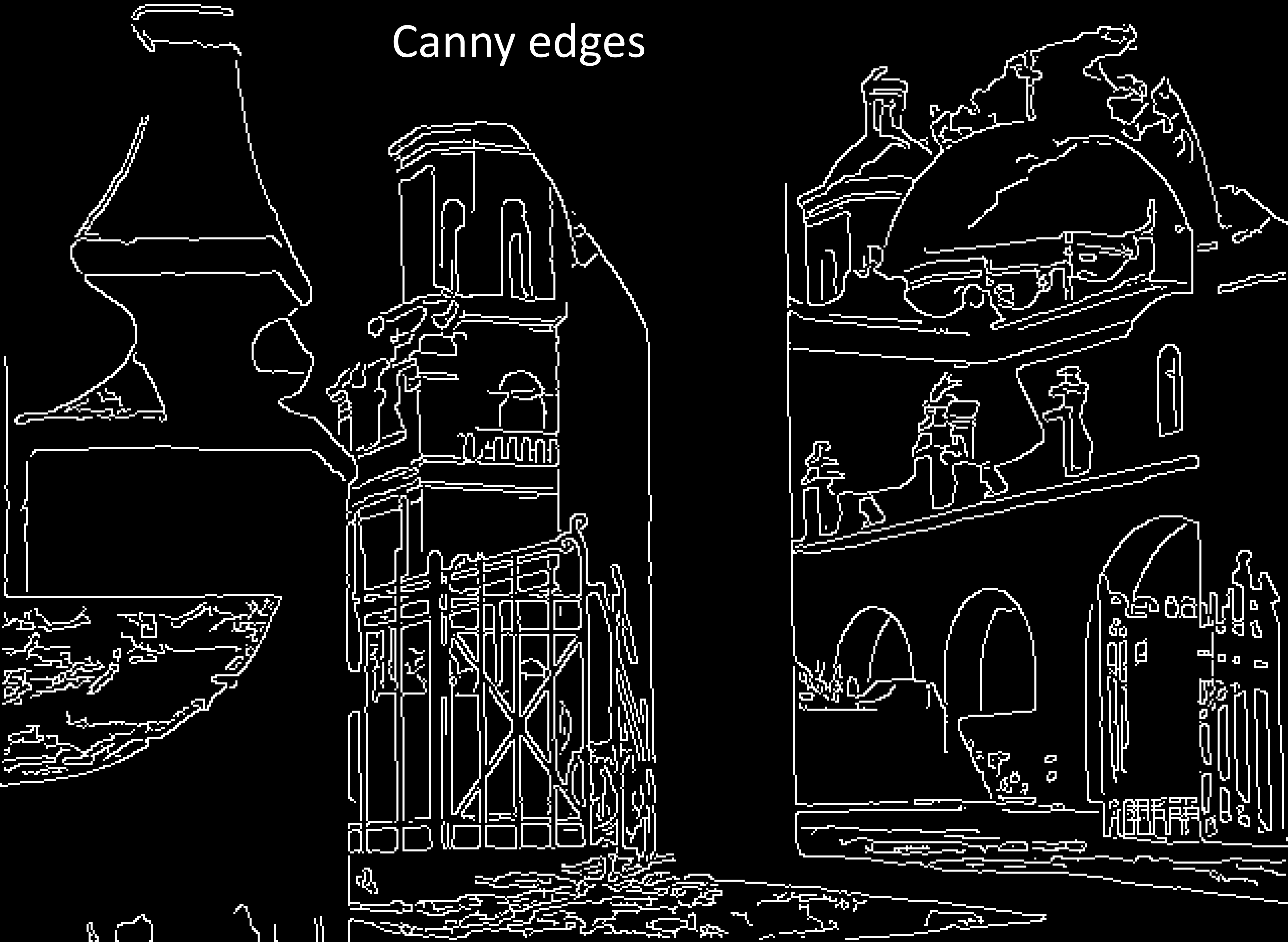
Derivative of Gaussian (x)



Derivative of Gaussian (y)

Photo: A. Adams (1968)

Canny edges

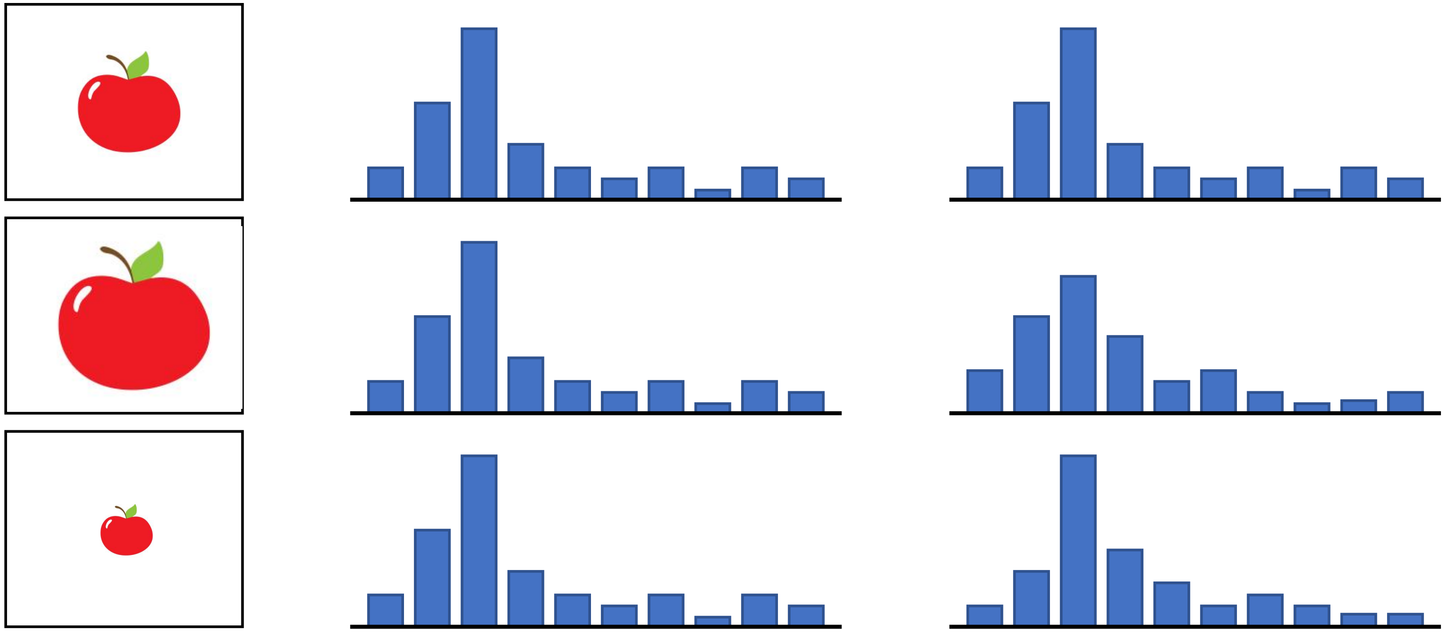


Edges and image recognition

- High gradient in image indicates change in the world:
  - Occlusion boundary
  - Change in surface normal (corner)
  - Change in colour
  - Change in light intensity (cast shadow)
- Edges are invariant or tolerant to many sources of image variation

Definition of invariance

- **Invariant** to X = response/representation does not vary with X, is insensitive to changes in X
- **Tolerant** to X = response is mostly insensitive to X



Invariant to light intensity?

- Image derivative is invariant to intensity shift ($I_{new} = I + b$)
- Tolerant to contrast change ($I_{new} = aI$), but depends on thresholds

Invariant to light direction?

Photo: Heshan Jayakody

Invariant to translation?

Invariant to rotation?

Invariant to scale?

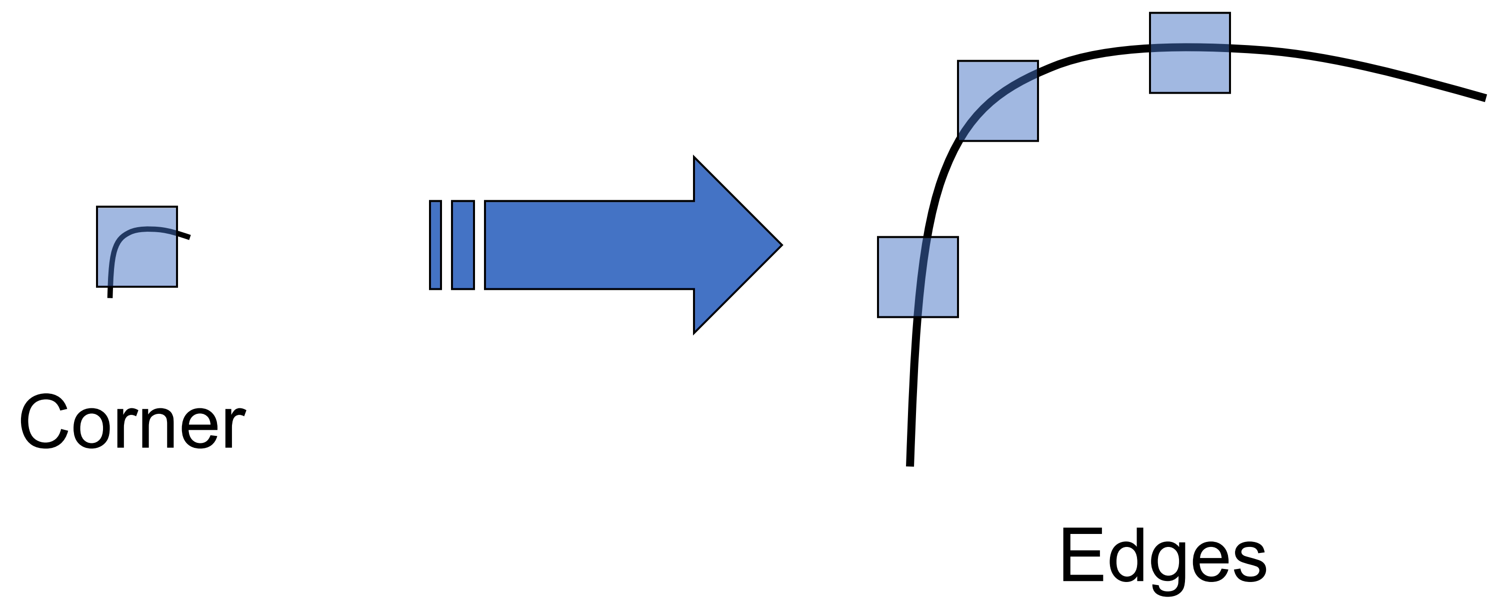


Image: K. Grauman

Invariant to 3D rotation / pose?

Summary

- Recovering world properties from a single image is an underconstrained problem
- Changes in image brightness (= image gradient, edges) indicate changes in world properties
- Most vision systems learn to detect edges because these are invariant features for object recognition