CS4501: Introduction to Computer Vision

Light and Image Processing

Various slides from previous courses by:
D.A. Forsyth (Berkeley / UIUC), I. Kokkinos (Ecole Centrale / UCL). S. Lazebnik (UNC / UIUC), S. Seitz (MSR / Facebook), J. Hays (Brown / Georgia Tech), A. Berg (Stony Brook / UNC), D. Samaras (Stony Brook) . J. M. Frahm (UNC), V. Ordonez (UVA).

Last Class

- Practical Advice on Photography
- Camera Parameters
- Brief Introduction to Projective Geometry (Computer Graphics)
- Intro to Light (BRDF)

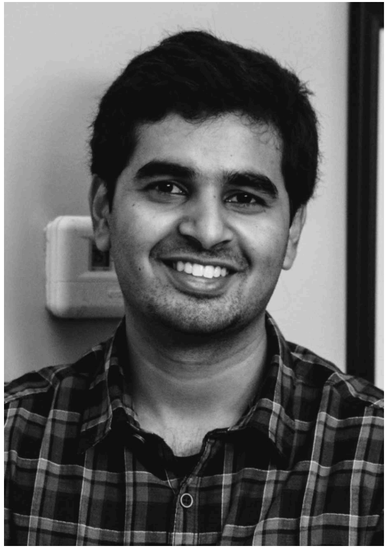
About the Course

CS4501-008: Introduction to Computer Vision

- Instructor: Vicente Ordonez
- Email: vicente@virginia.edu
- Website: http://vicenteordonez.com/vision/
- Class Location: **Thornton Hall E316**
- Class Times: **Monday-Wednesday 2pm - 3:15pm**
- Piazza: http://piazza.com/virginia/spring2018/cs4501008/home

Teaching Assistants + Office Hours

Fengyang Zhang
Tuesday 3pm to 4pm (Rice 340)
Thursday 3pm to 4pm (Rice 340)



Gautam Somappa
Monday 4pm to 5pm (Rice 436)
Tuesday 2pm to 3pm (Rice 436)

Siva Sivaraman
Wednesday 3:30 to 4:30pm (Rice 436)
Thursday 2pm to 3pm (Rice 340)

Things to Remember for Quiz

- Pinhole camera model
- Focal length in the pinhole camera model
- Shutter Time / Aperture / ISO
- Homogeneous Coordinates
- Extrinsic Camera Properties and Intrinsic Camera Properties
- Describe mathematically (and intuitively) the conversion process from World Coordinates to Image Coordinates (Should be easy after completing the first programming assignment)

Light

- What determines the color of a pixel?

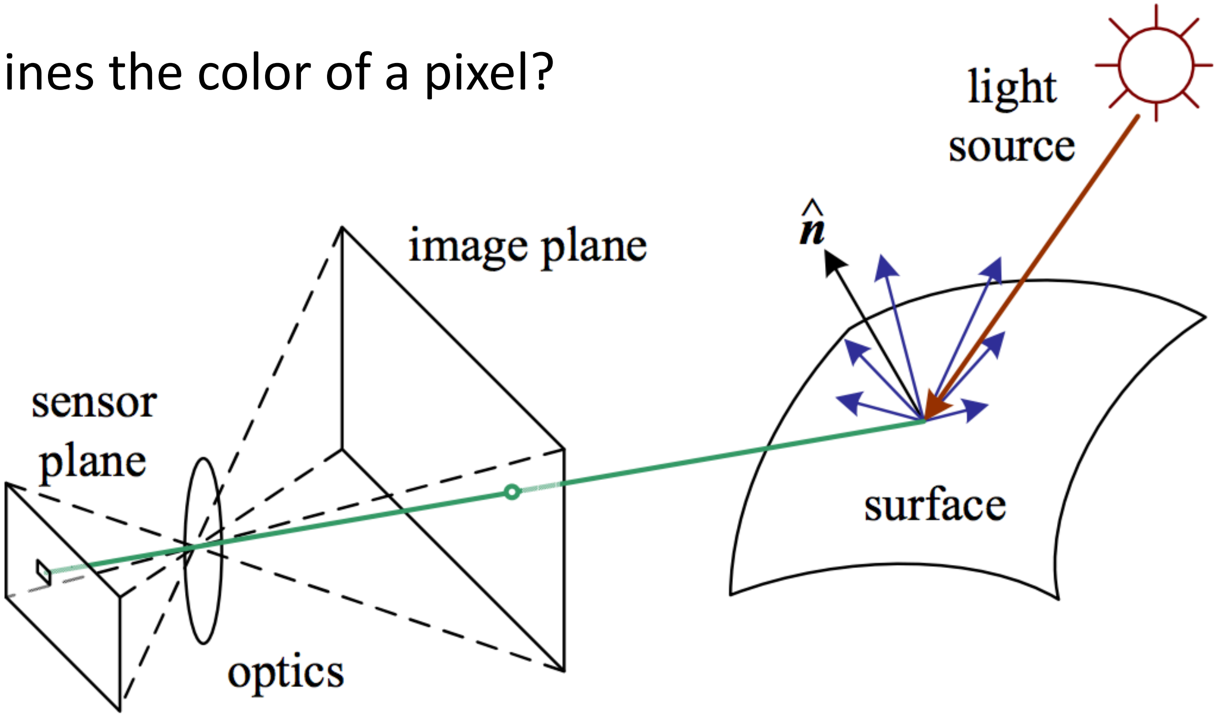
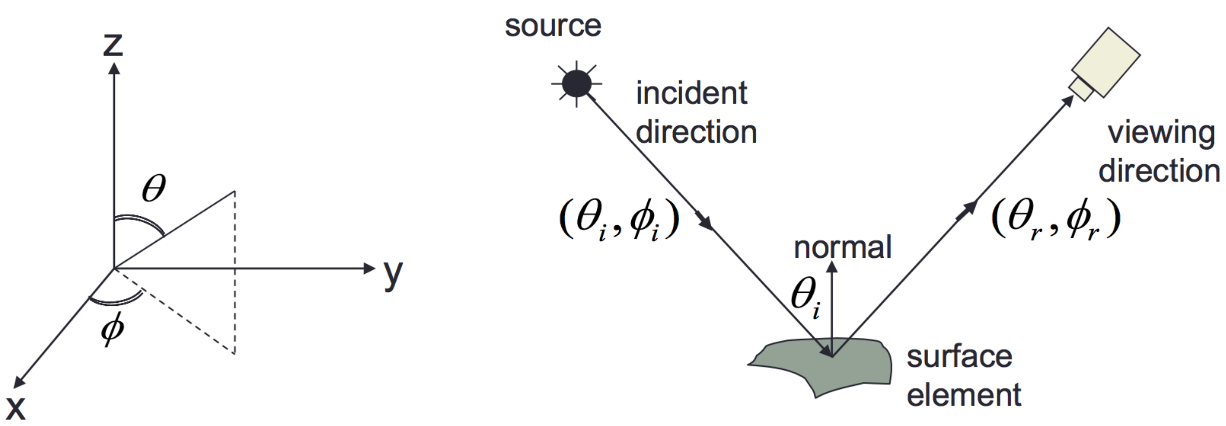


Figure from Szeliski

BRDF (Bidirectional reflectance distribution function)



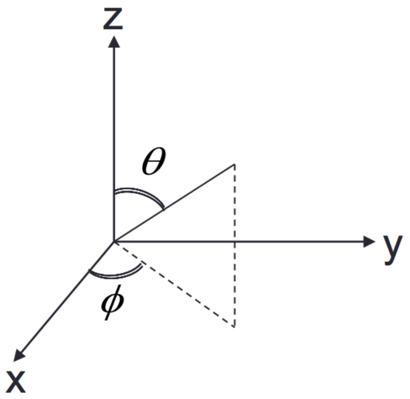
$E^{surface}(\theta_i, \phi_i)$ Irradiance at Surface in direction (θ_i, ϕ_i)

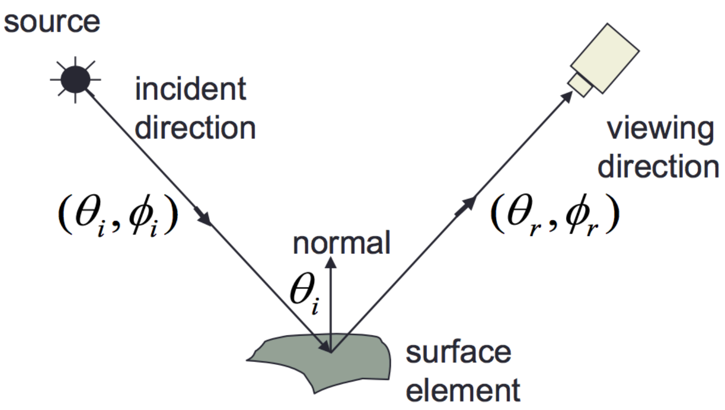
$L^{surface}(\theta_r, \phi_r)$ Radiance of Surface in direction (θ_r, ϕ_r)

$$E^{surface}(\theta_i, \phi_i) \sim \cos\theta_i \, L^{surface}(\theta_i, \phi_i)$$

Slide by Aaron Bobick

BRDF (Bidirectional reflectance distribution function)





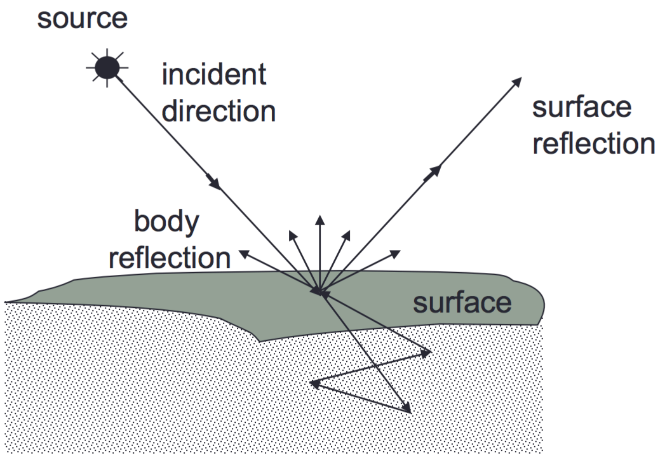
$E^{surface}(\theta_i, \phi_i)$ Irradiance at Surface in direction (θ_i, ϕ_i)

$L^{surface}(\theta_r, \phi_r)$ Radiance of Surface in direction (θ_r, ϕ_r)

$$\text{BRDF : } f(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{L^{surface}(\theta_r, \phi_r)}{E^{surface}(\theta_i, \phi_i)}$$

Slide by Aaron Bobick

Reflection



- Body Reflection:

  Diffuse Reflection
  Matte Appearance
  Non-Homogeneous Medium
  Clay, paper, etc

- Surface Reflection:

  Specular Reflection
  Glossy Appearance
  Highlights
  Dominant for Metals

Image Intensity = Body Reflection + Surface Reflection

Slide by Aaron Bobick

Reflection

Body Reflection:

Diffuse Reflection
Matte Appearance
Non-Homogeneous Medium
Clay, paper, etc

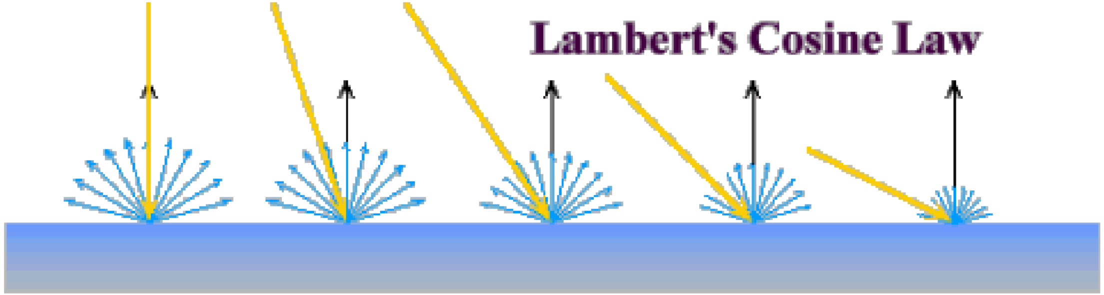
Surface Reflection:

Specular Reflection
Glossy Appearance
Highlights
Dominant for Metals

Many materials exhibit **both Reflections**:

Slide by Aaron Bobick

Diffuse Reflection – Lambertian Surface / BRDF



- Only body reflection, and no specular reflection
- BRDF is independent of outgoing direction
- BRDF depends on indent direction (foreshortening)

- Light intensity does not depend on the outgoing direction. Only incoming.
- It is independent of where the viewer stands.
- Smooth surface, not glossy. Can think of any examples?

Slide by Aaron Bobick

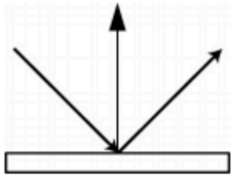
CAN’T perceive the shape of the snow covered terrain!

CAN perceive shape in regions lit by the street lamp!!

WHY?

Slide by Aaron Bobick

The other extreme – Only Specular Reflection



How about a mirror?

Reflection **ONLY** at mirror angle

https://wonderopolis.org/wonder/how-do-mirrors-work

Slide by Aaron Bobick

Problem in Computer Vision:
Intrinsic Image Decomposition

Given this

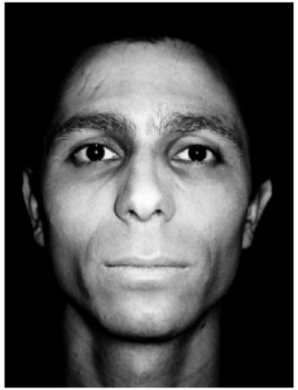
Extract this

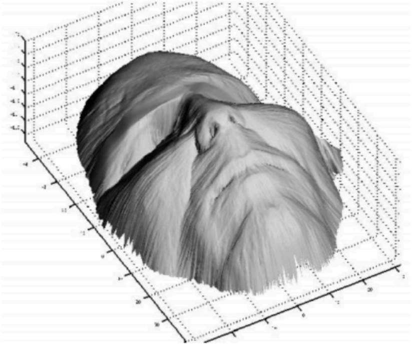
Images by Marc Serra

Problem in Computer Vision:
Shape from Shading

Given this

Extract this





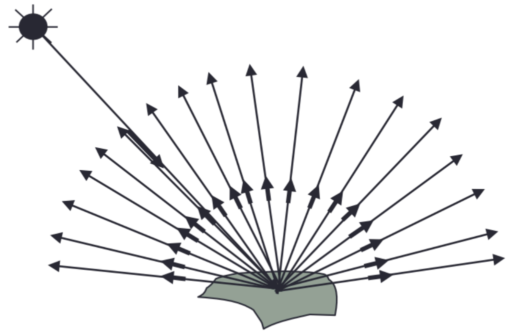
Images by Aaron Bobick

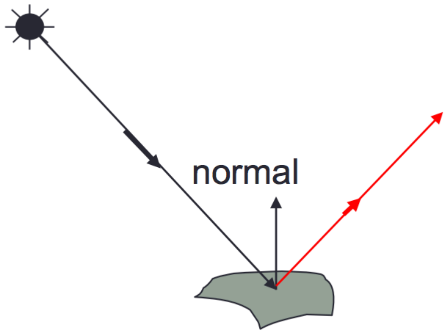
Same ideas used in Computer Graphics

- Ray Tracing
- Radiosity
- Photon Mapping

Phong Reflection Model

- The BRDF of many surfaces can be approximated by
  The Lambertian + Specular Model





Slide by Aaron Bobick

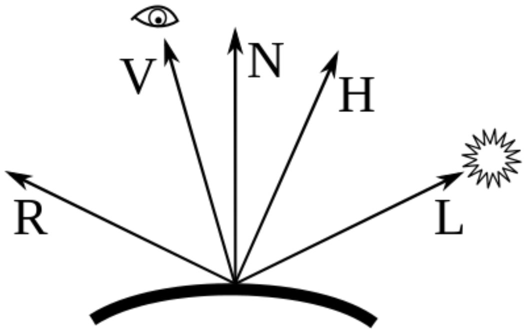
Phong Reflection Model

$\hat{L}_m$, which is the direction vector from the point on the surface toward each light source (m specifies the light source),

$\hat{N}$, which is the normal at this point on the surface,

$\hat{R}_m$, which is the direction that a perfectly reflected ray of light would take from this point on the surface, and

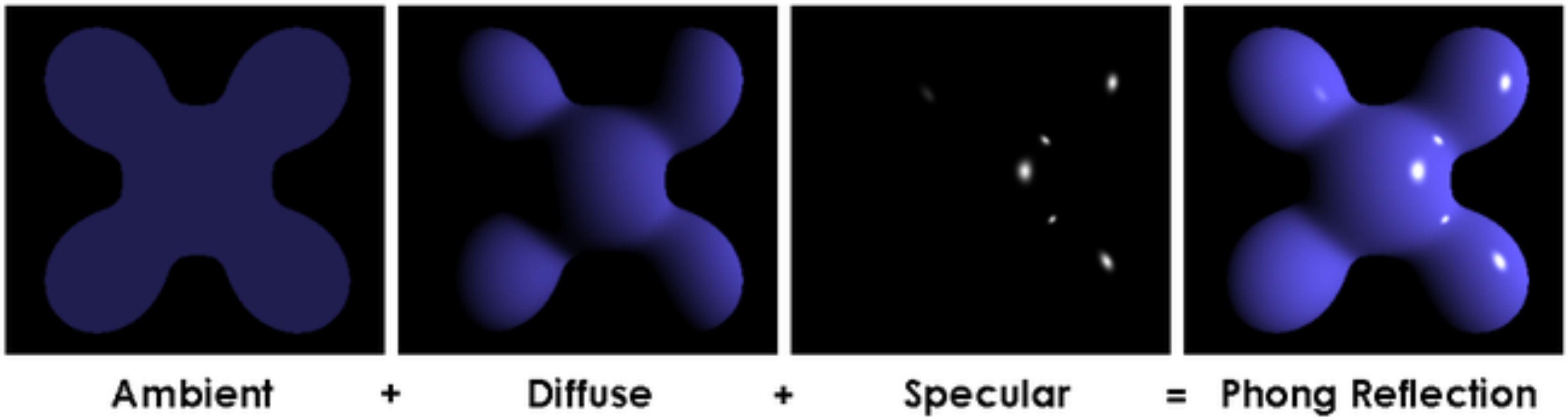
$\hat{V}$, which is the direction pointing towards the viewer (such as a virtual camera).



$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,s}).$$

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

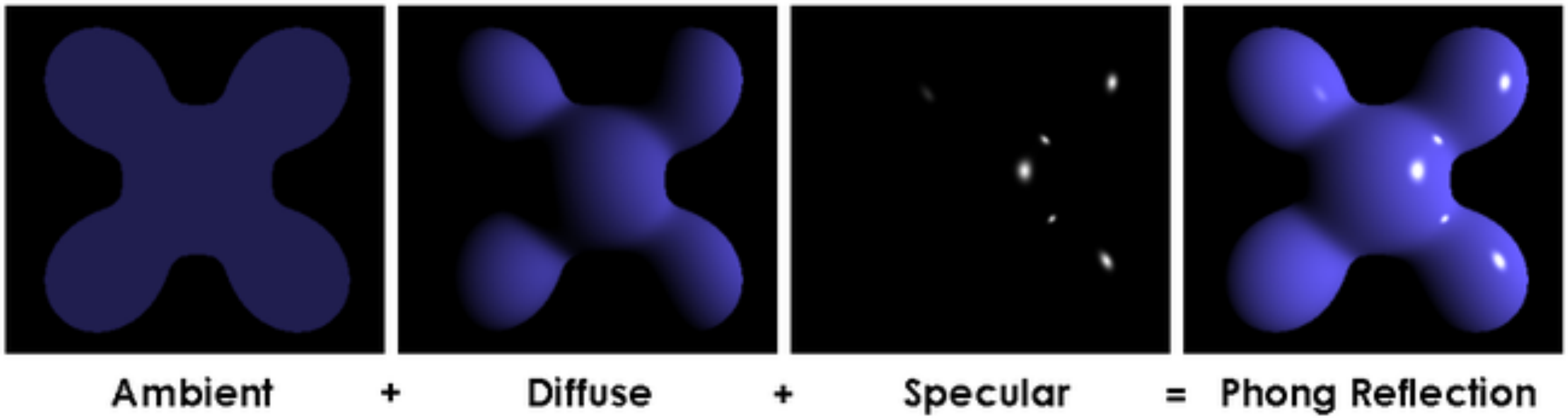
Phong Reflection Model - Recap



$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,s}).$$

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

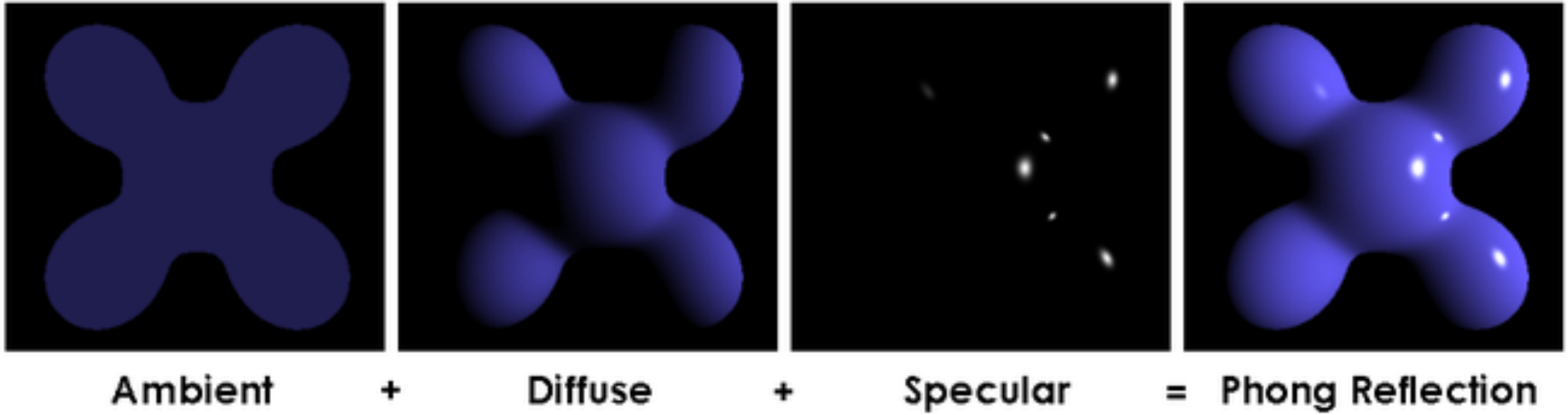
Phong Reflection Model - Recap



$$I_p = \boxed{k_a i_a} + \sum_{m \,\in\, \text{lights}} \left(k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,s}\right).$$

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

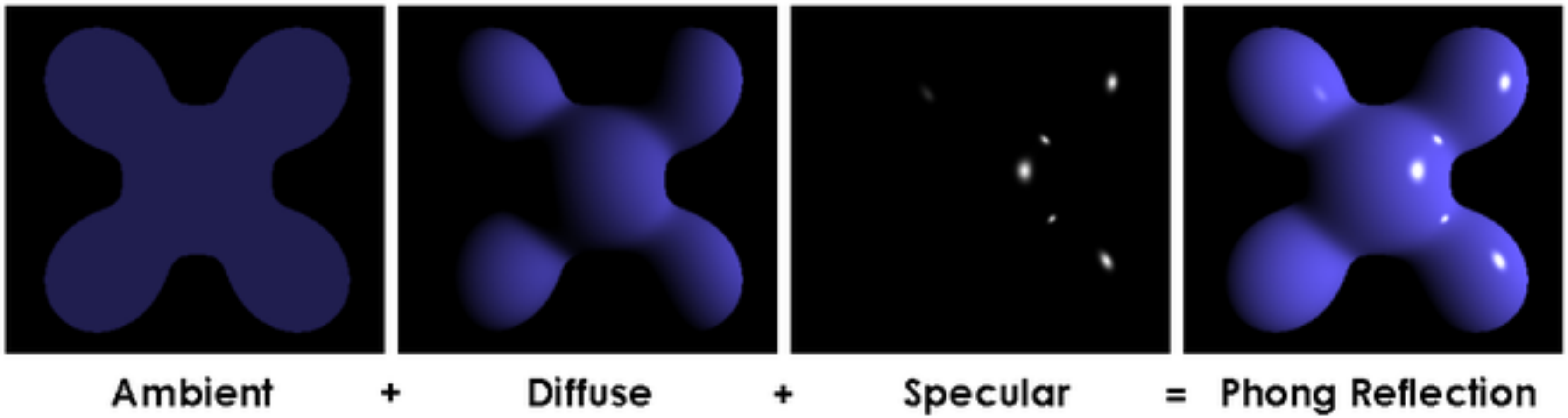
Phong Reflection Model - Recap



$$I_p = k_a i_a + \sum_{m \in \text{lights}} \left(k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,s} \right).$$

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

Phong Reflection Model - Recap



$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,s}).$$

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

Phong’s Shading / Illumination Model

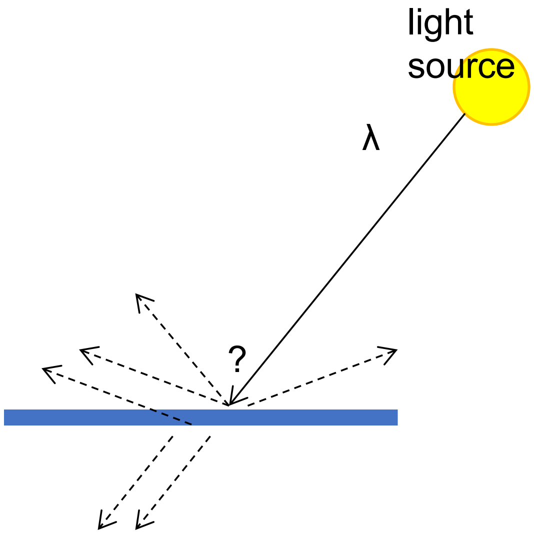
- Originally from Vietnam / PhD from Utah, Professor at Utah, and later Stanford.

- Died at age 32 from leukemia

- Phong’s professor Ivan Sutherland went on to win the Turing Award (Nobel Prize in CS) for lifelong contributions to Computer Graphics

A photon's life choices

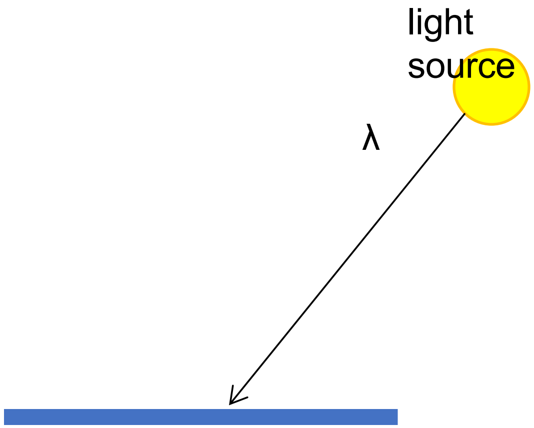
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

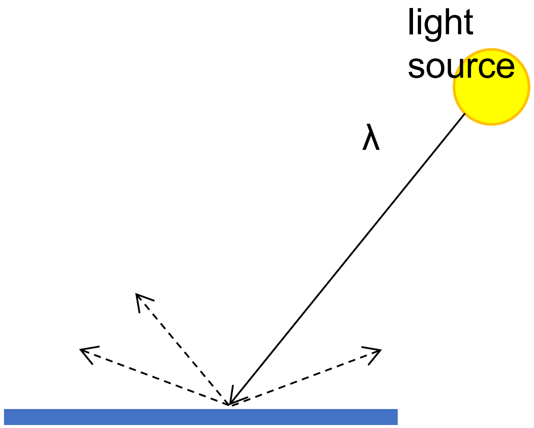
- **Absorption**
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

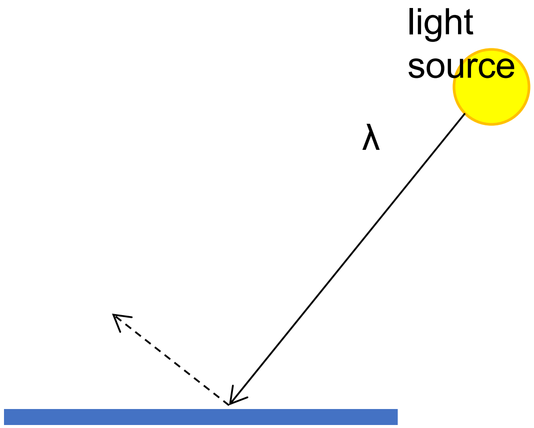
- Absorption
- **Diffuse Reflection**
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon’s life choices

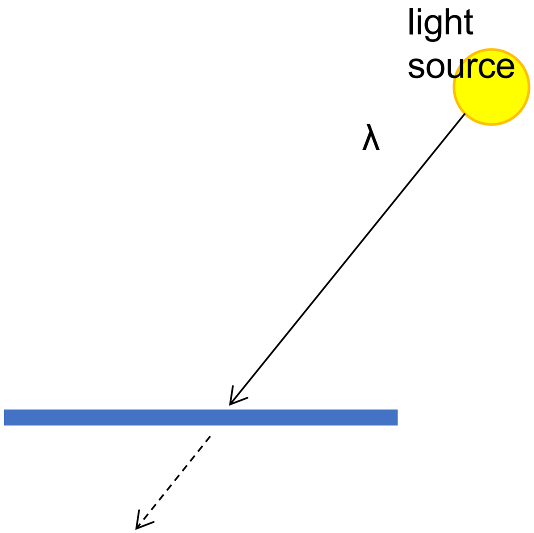
- Absorption
- Diffusion
- **Specular Reflection**
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

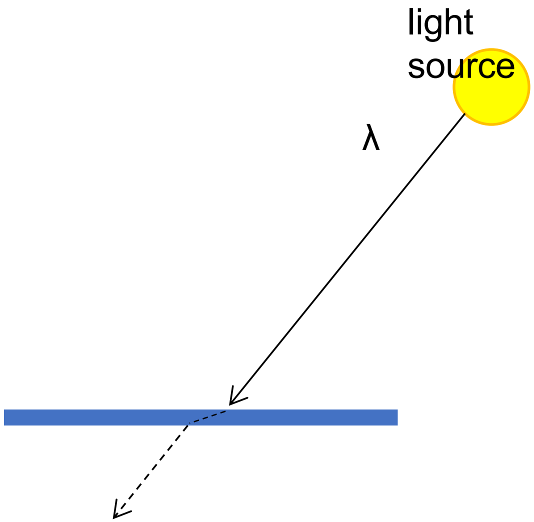
- Absorption
- Diffusion
- Reflection
- **Transparency**
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

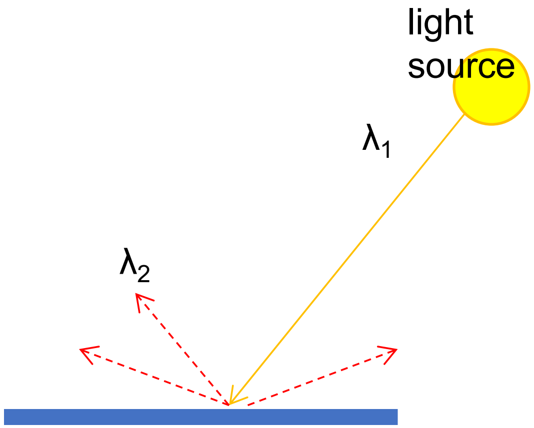
- Absorption
- Diffusion
- Reflection
- Transparency
- **Refraction**
- Fluorescence
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

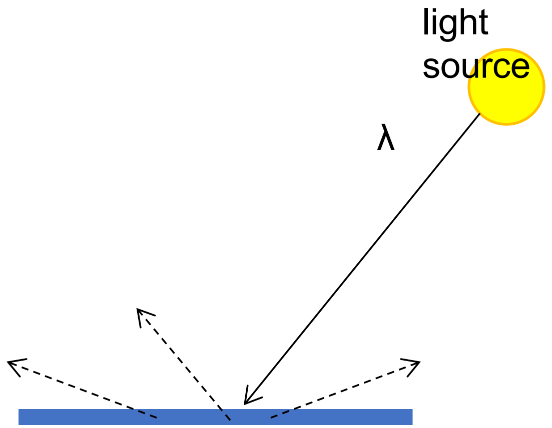
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- **Fluorescence**
- Subsurface scattering
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

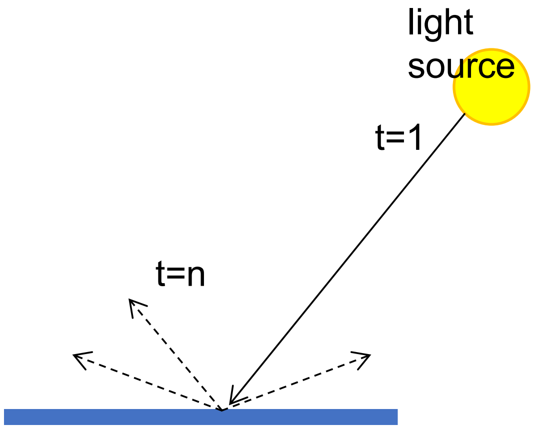
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- **Subsurface scattering**
- Phosphorescence
- Interreflection



Slide by James Hays

A photon's life choices

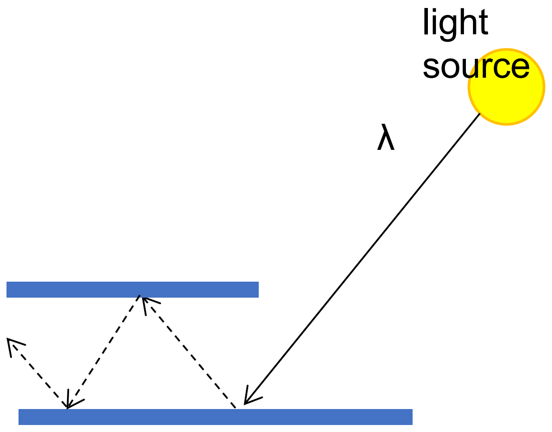
- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- **Phosphorescence**
- Interreflection



Slide by James Hays

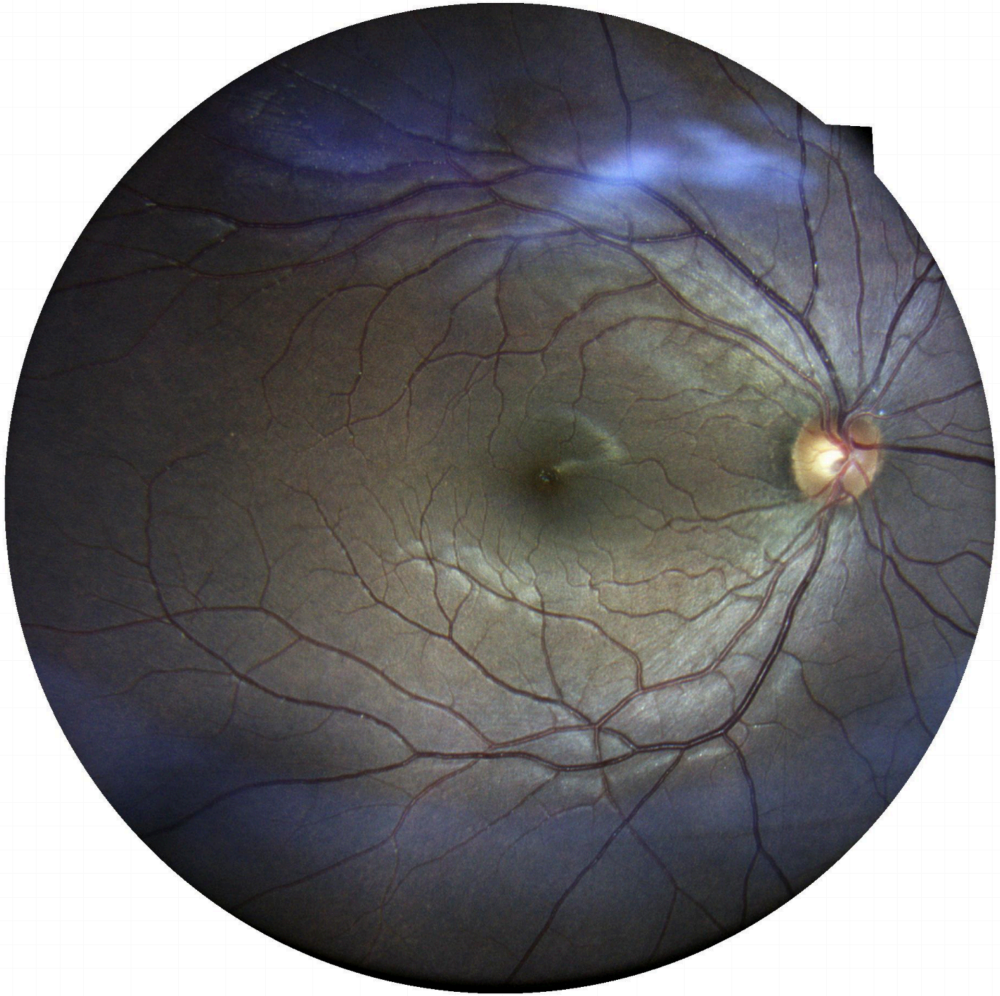
A photon’s life choices

- Absorption
- Diffusion
- Reflection
- Transparency
- Refraction
- Fluorescence
- Subsurface scattering
- Phosphorescence
- **Interreflection**



(Specular Interreflection)

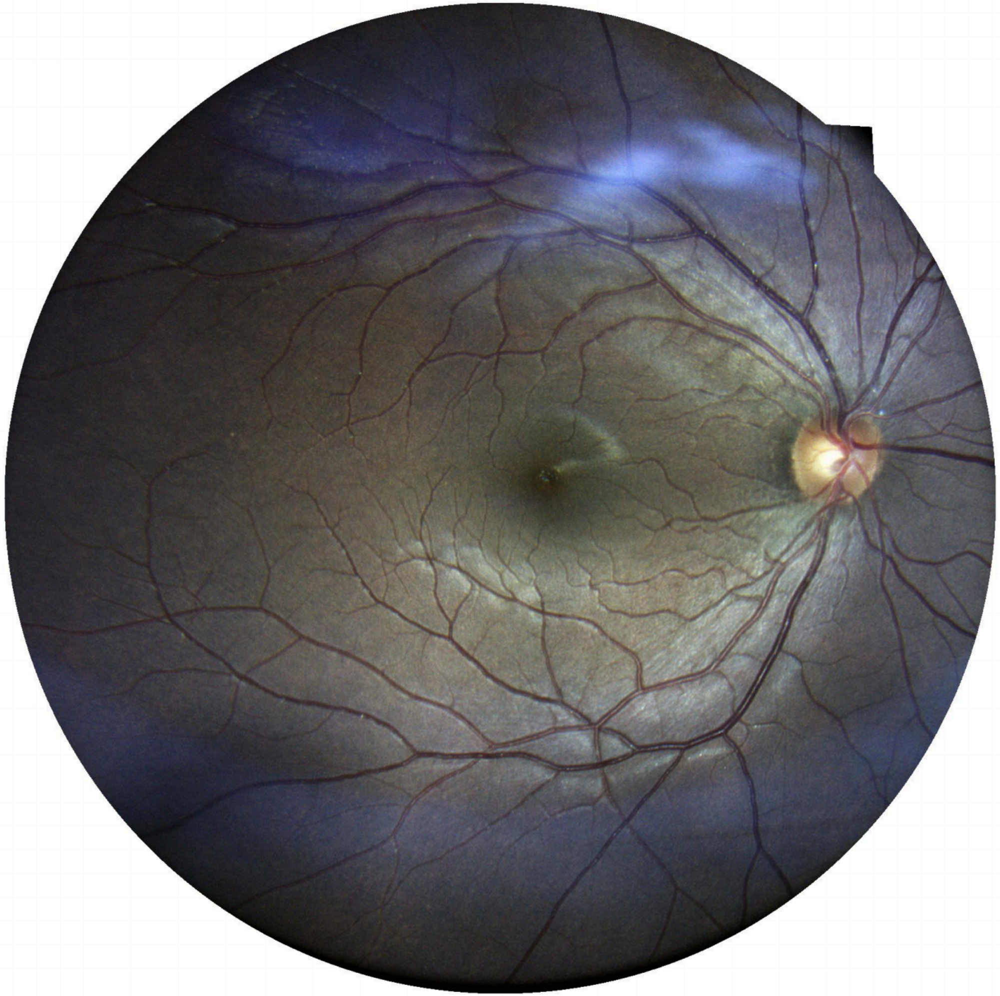
Slide by James Hays



Vicente’s eye

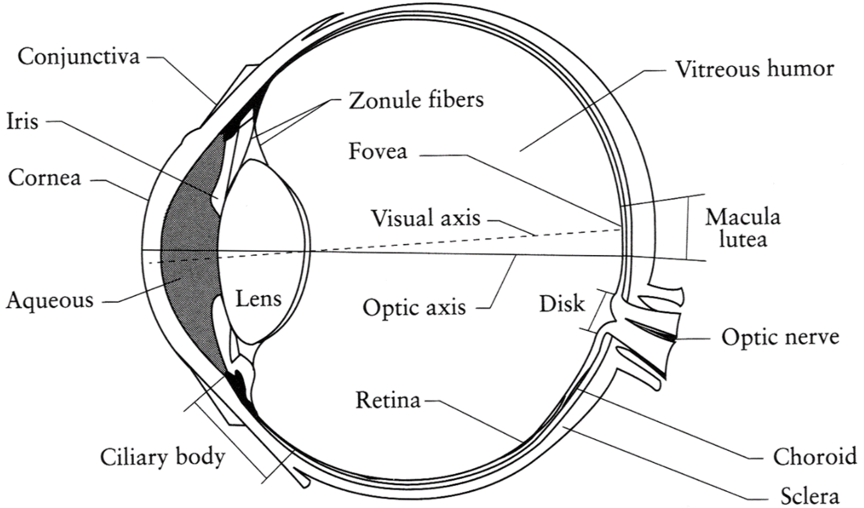
Our own Camera as a species:

The Human Eye



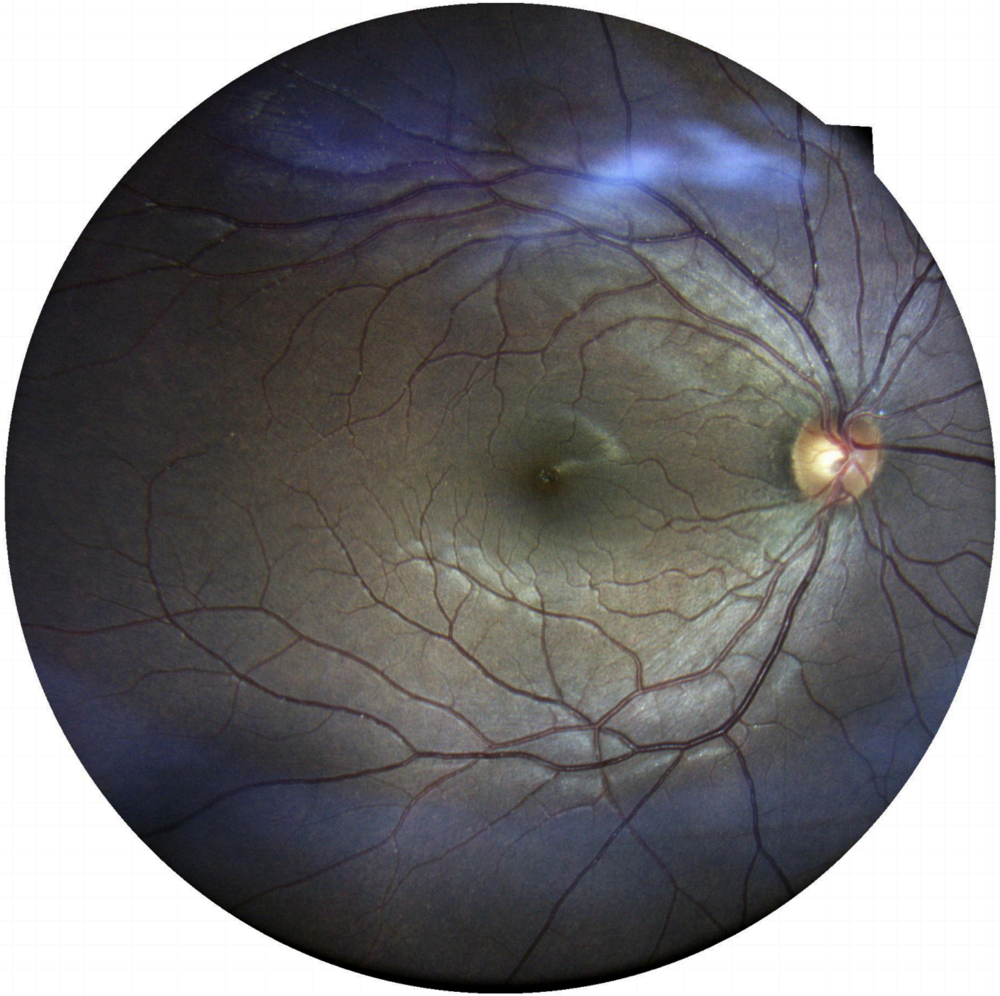
Vicente’s eye

The Eye



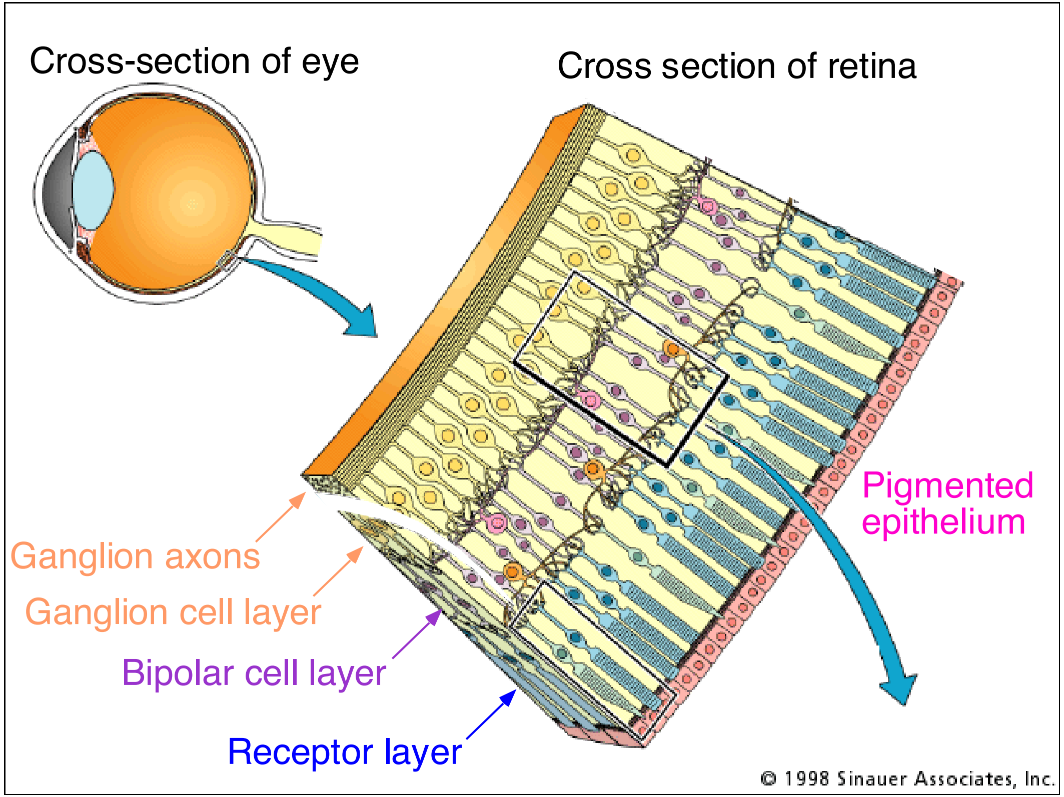
- The human eye is a camera!
  - **Iris** - colored annulus with radial muscles
  - **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What's the "film"?
    - photoreceptor cells (rods and cones) in the **retina**

Slide by Steve Seitz



Vicente’s eye

The Retina

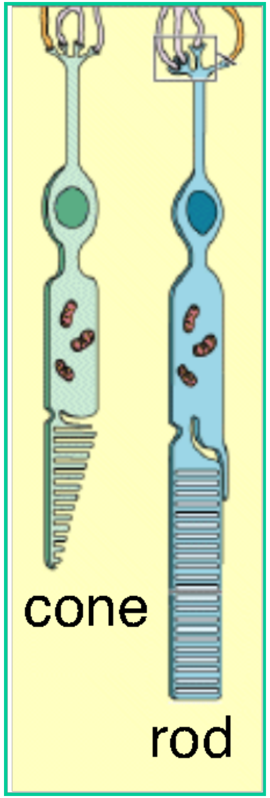


Cross-section of eye
Cross section of retina
Ganglion axons
Ganglion cell layer
Bipolar cell layer
Receptor layer
Pigmented epithelium
© 1998 Sinauer Associates, Inc.

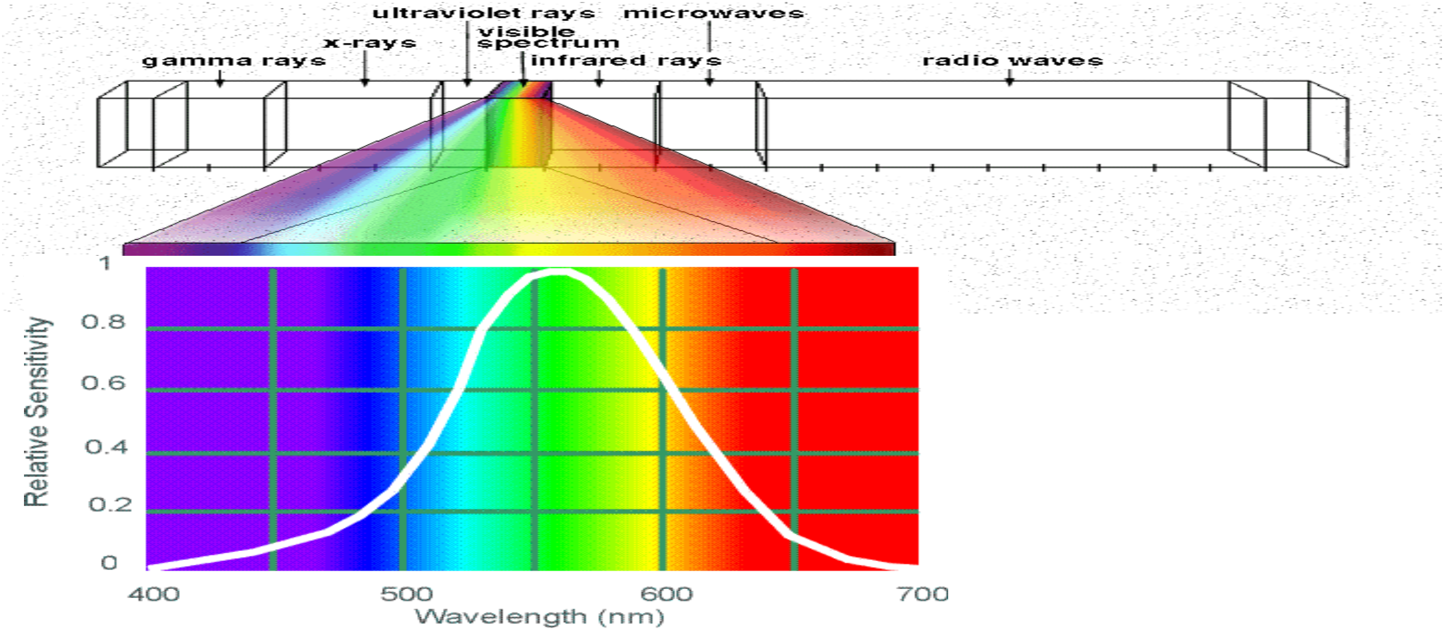
Two types of light-sensitive receptors

Cones
- cone-shaped
- less sensitive
- operate in high light
- color vision

Rods
- rod-shaped
- highly sensitive
- operate at night
- gray-scale vision

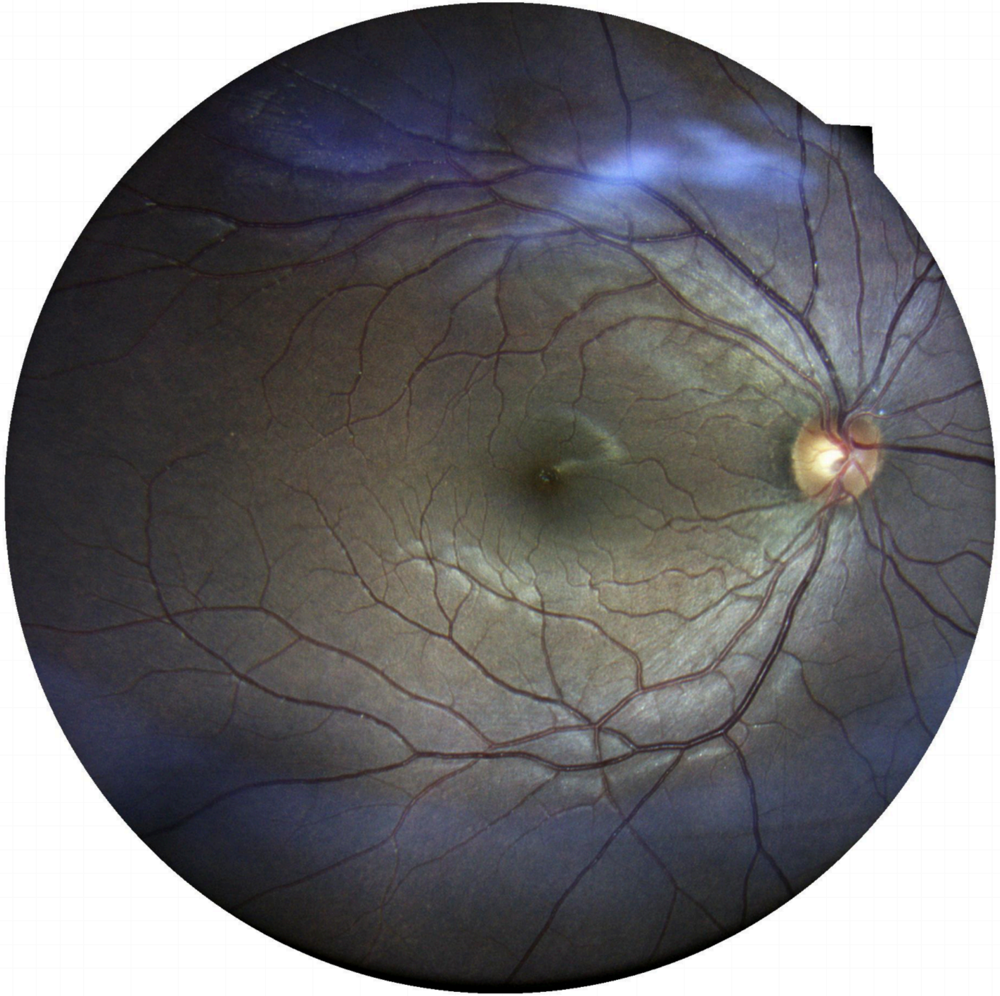


Electromagnetic Spectrum



Human Luminance Sensitivity Function

http://www.yorku.ca/eye/photopik.htm



Vicente’s eye

More about the eye

https://www.youtube.com/watch?v=L_W-IXqoxHA

What you need to know for a Quiz

- Describe the various factors that modify object's lighting
  - Camera position (viewer position)
  - Light positions
  - Object shape (surface normals) and material properties (BRDF)
- Understanding the effect of the following:
  - Ambient Light
  - Diffuse Light
  - Specular Light

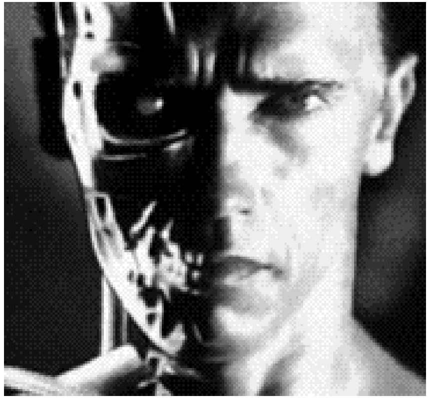
Image Processing & Image Filtering

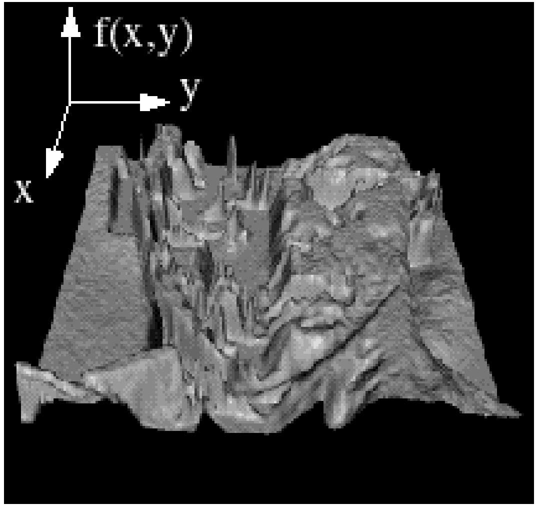
Reminder of what is an image for a computer.

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Images as Functions

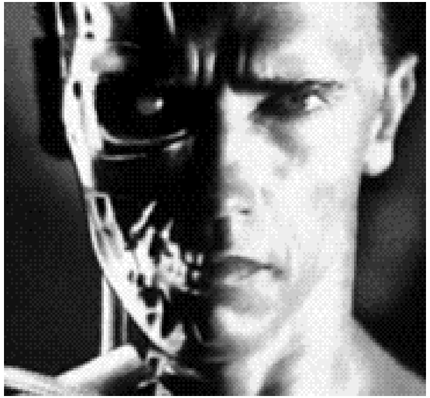
$$z = f(x, y)$$

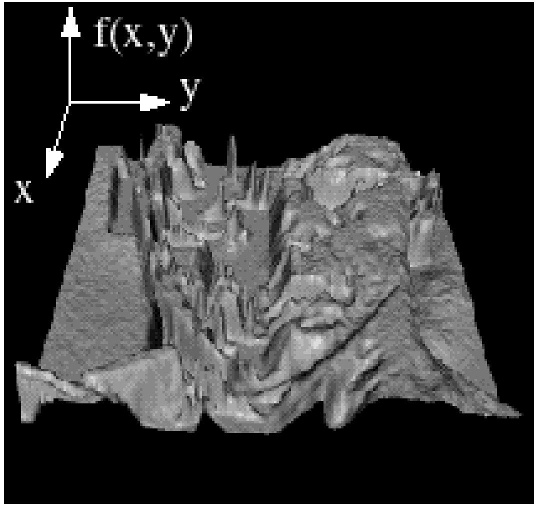




Images as Functions

$$z = f(x, y)$$



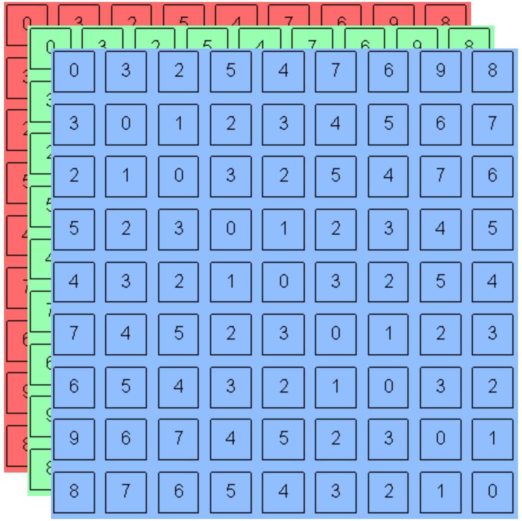


- The domain of x and y is [0, img-width) and [0 and img-height)
- x, and y are discretized into integer values.

Images as Matrices

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

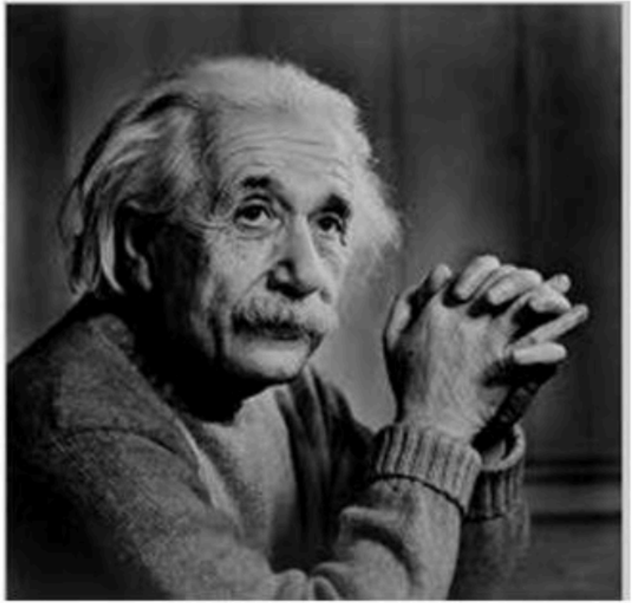
Color Images as Tensors



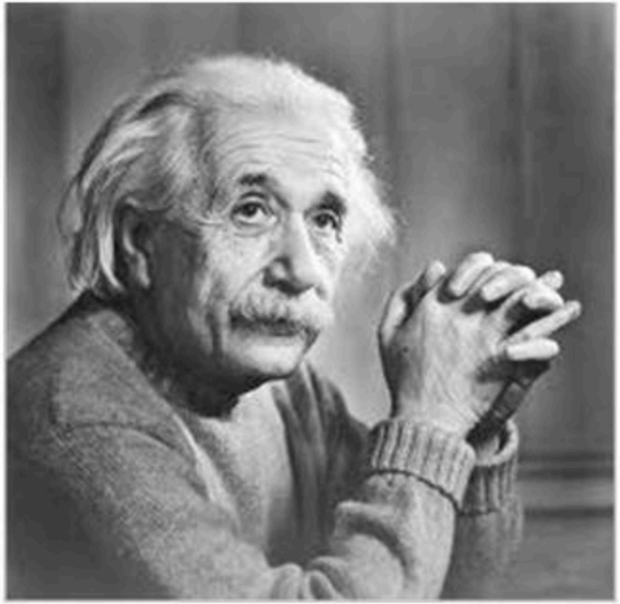
channel x height x width

Basic Image Processing

I



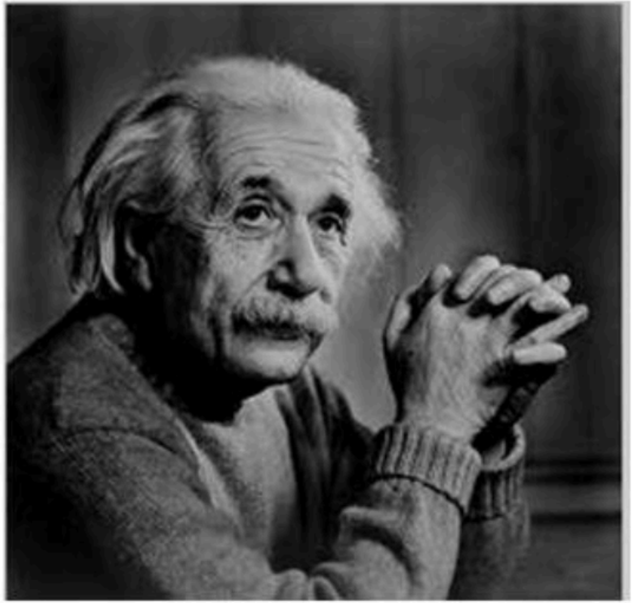
αI



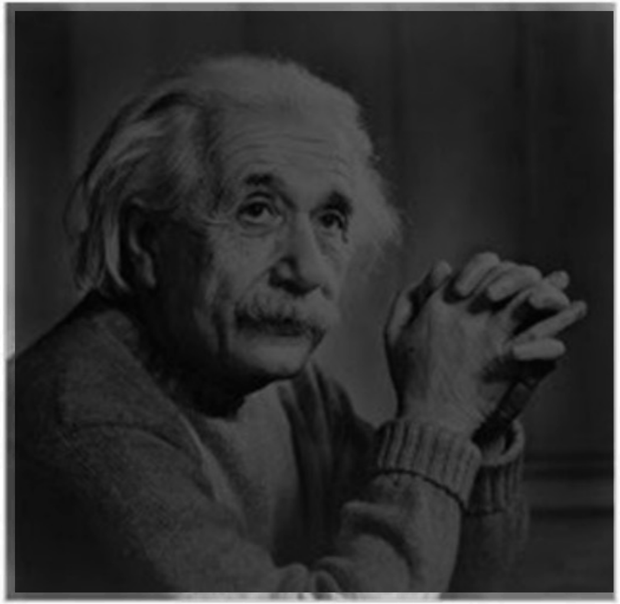
$\alpha > 1$

Basic Image Processing

I

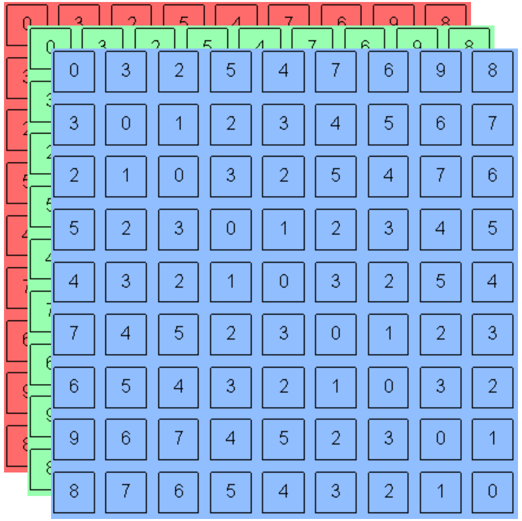


αI



$0 < \alpha < 1$

Color Images as Tensors

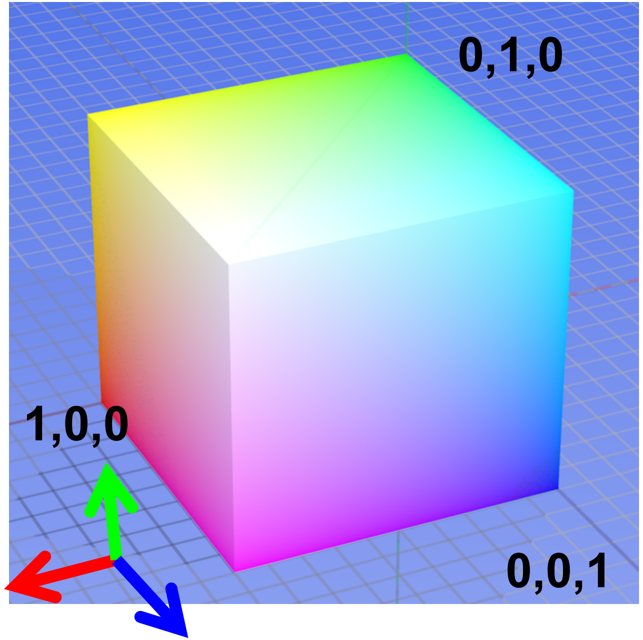


channel x height x width

Channels are usually RGB: Red, Green, and Blue

Other color spaces: HSV, HSL, LUV, XYZ, Lab, CMYK, etc

Color spaces: RGB



R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)

Some drawbacks
- Strongly correlated channels
- Non-perceptual

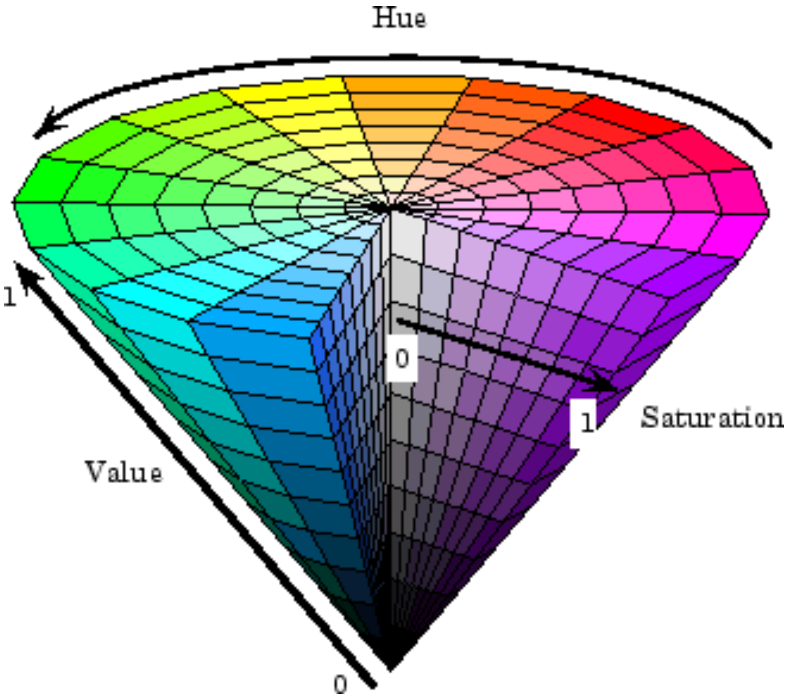
Default color space

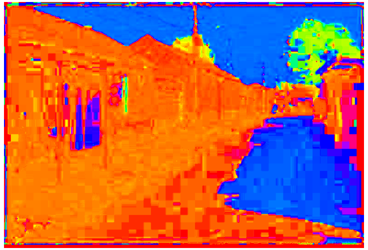
Slide by James Hays

Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Color spaces: HSV

Intuitive color space





H
(S=1,V=1)

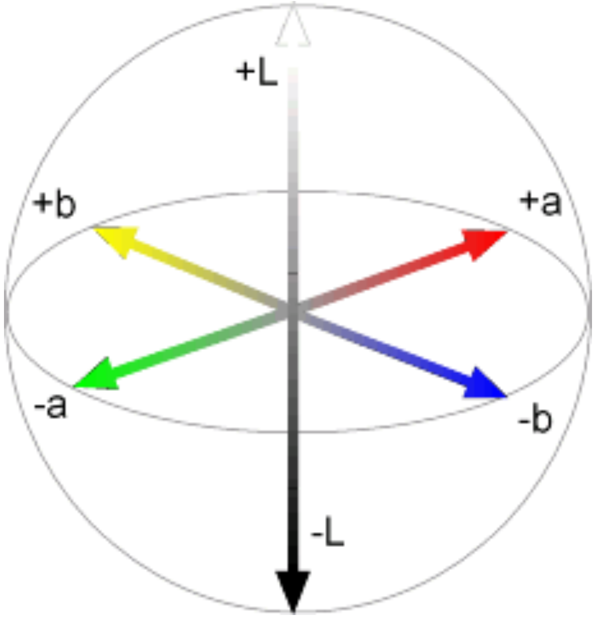
S
(H=1,V=1)

V
(H=1,S=0)

Slide by James Hays

Color spaces: L*a*b*

"Perceptually uniform"* color space



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Slide by James Hays

Most information in intensity

Only color shown – constant intensity

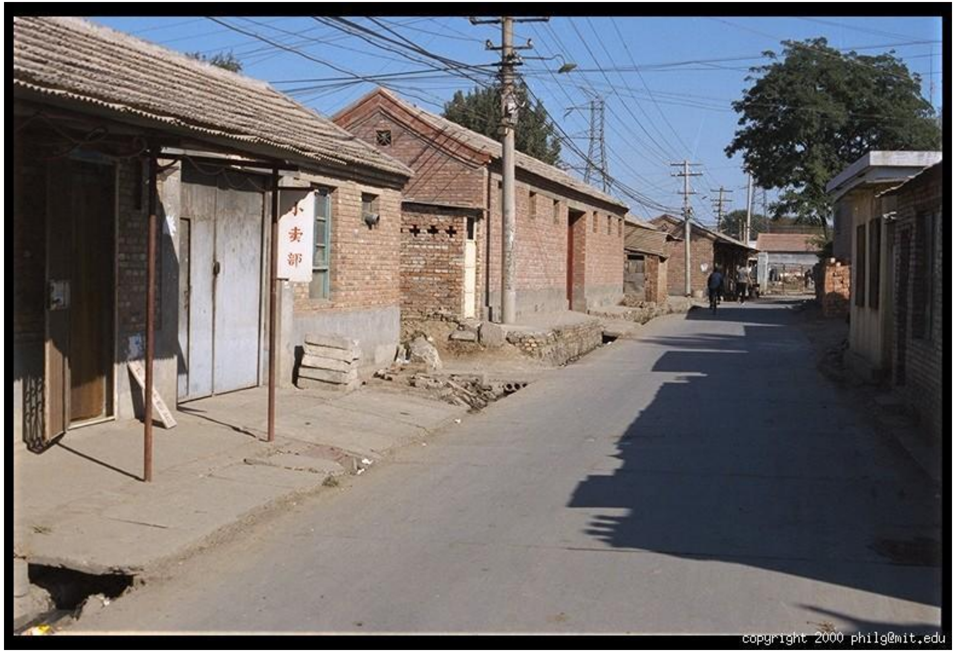
Slide by James Hays

Most information in intensity

Only intensity shown – constant color

Slide by James Hays

Most information in intensity



Original image

Image filtering

Image filtering

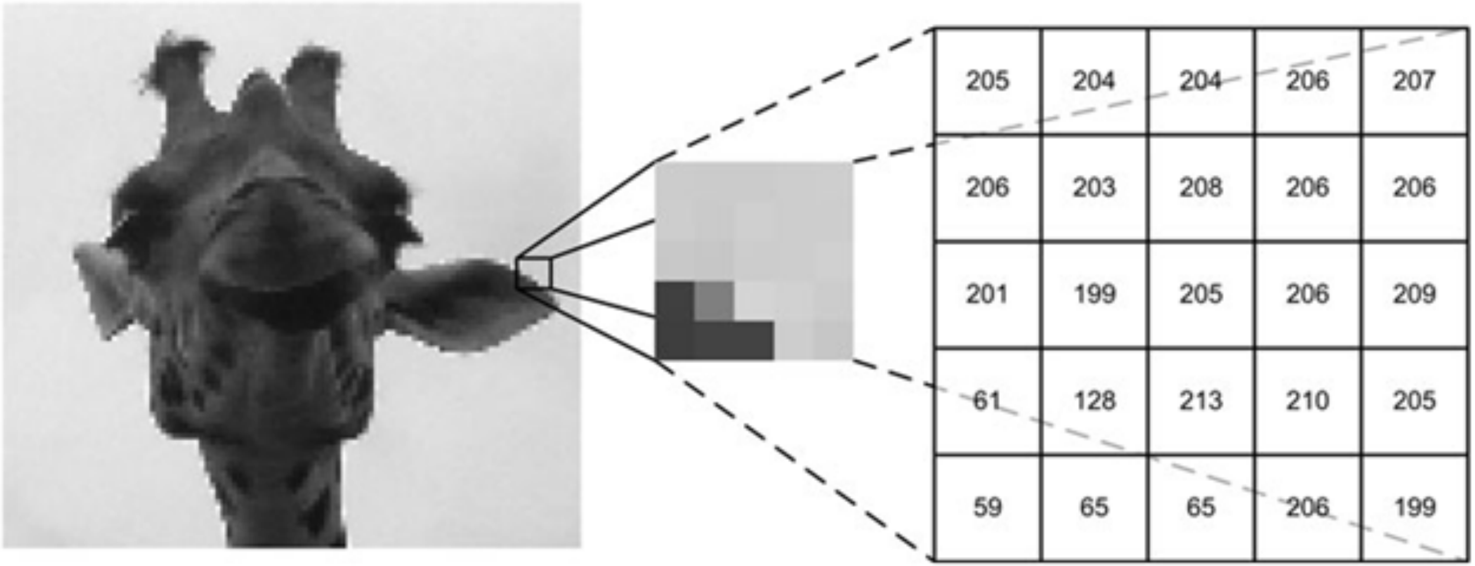


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering

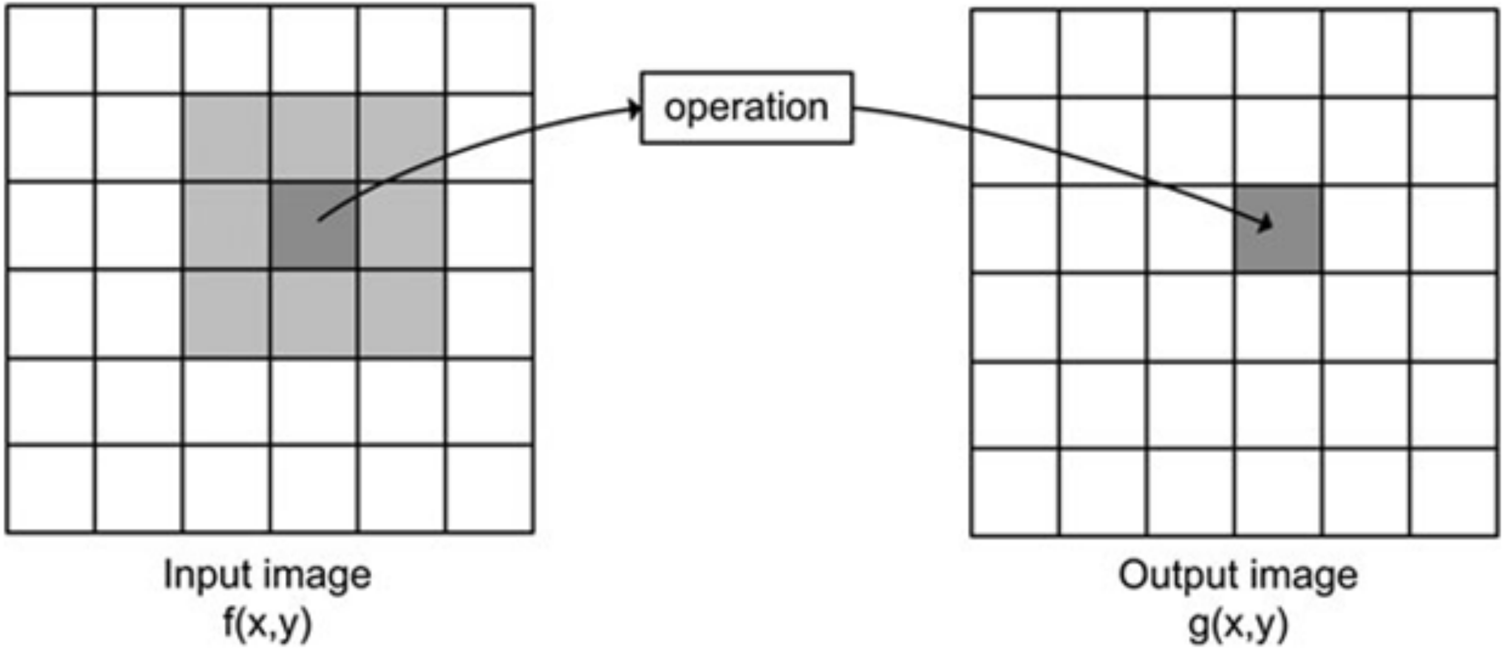
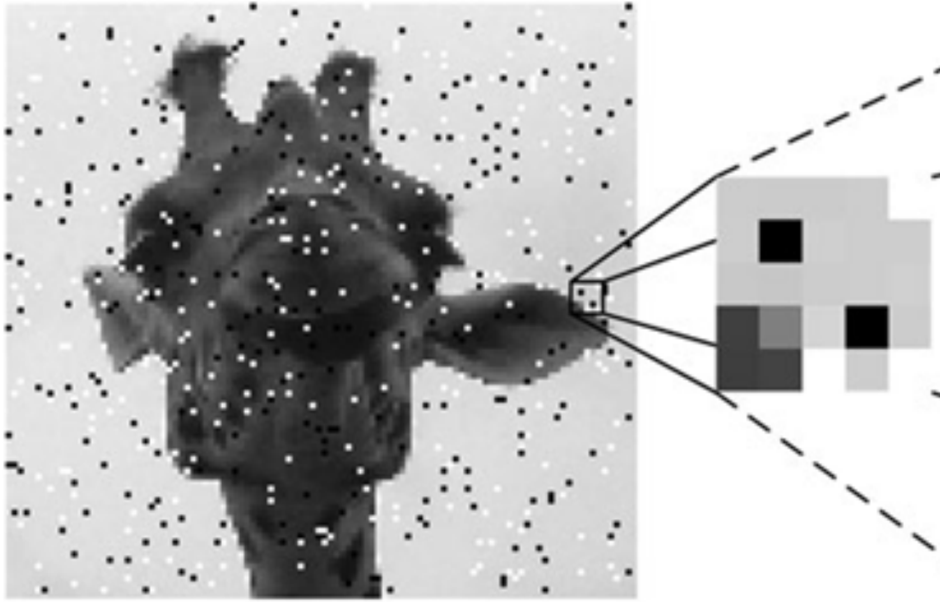


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: e.g. Mean Filter



205	204	204	206	255
206	0	208	206	206
201	199	205	206	209
61	128	213	0	205
59	65	255	206	255

Image filtering: e.g. Mean Filter

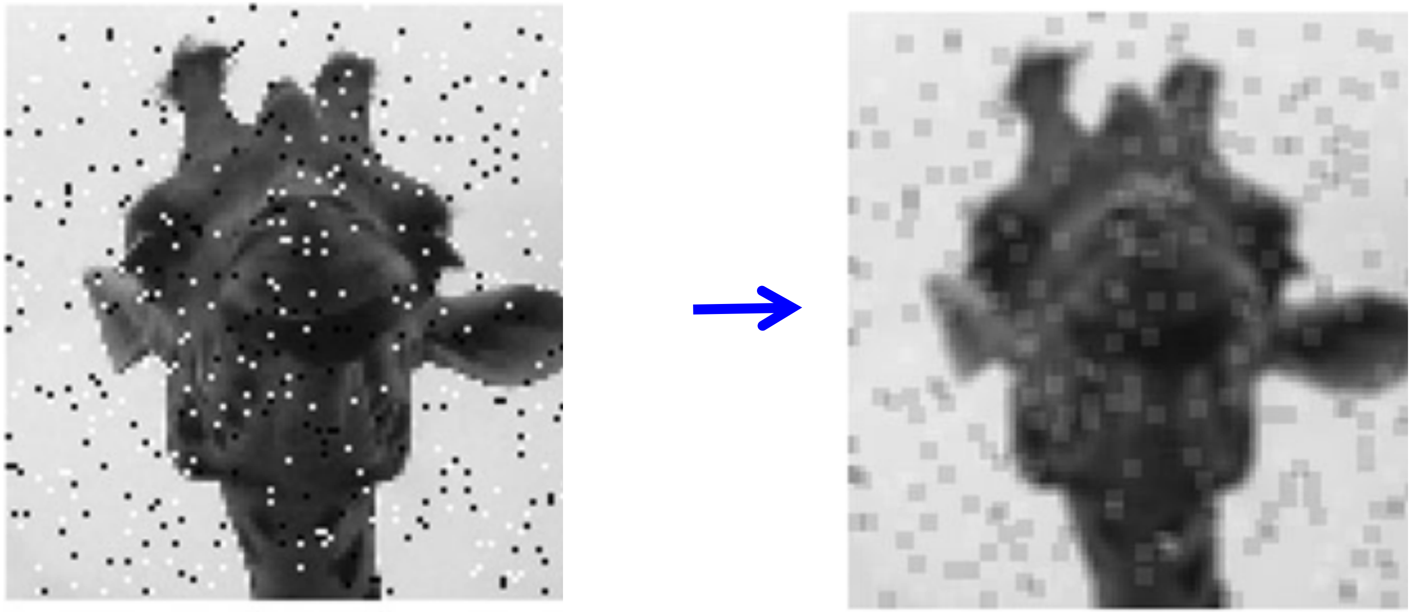


Image filtering: e.g. Median Filter

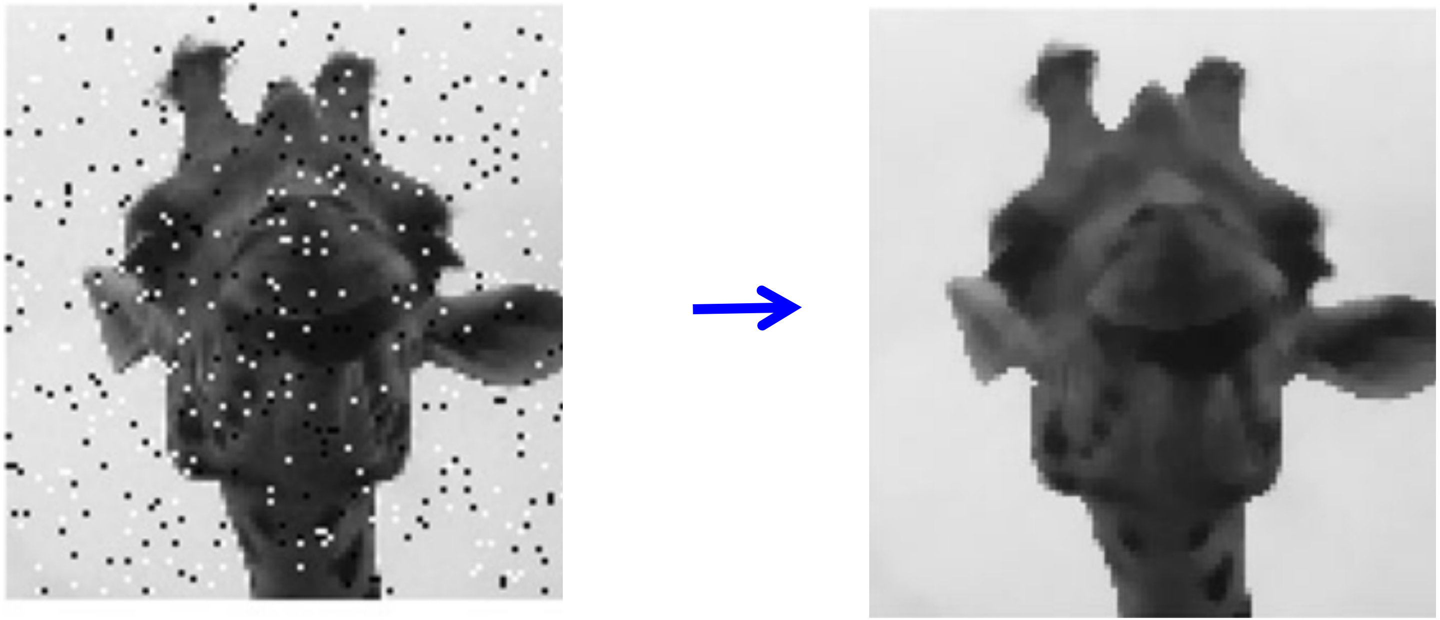
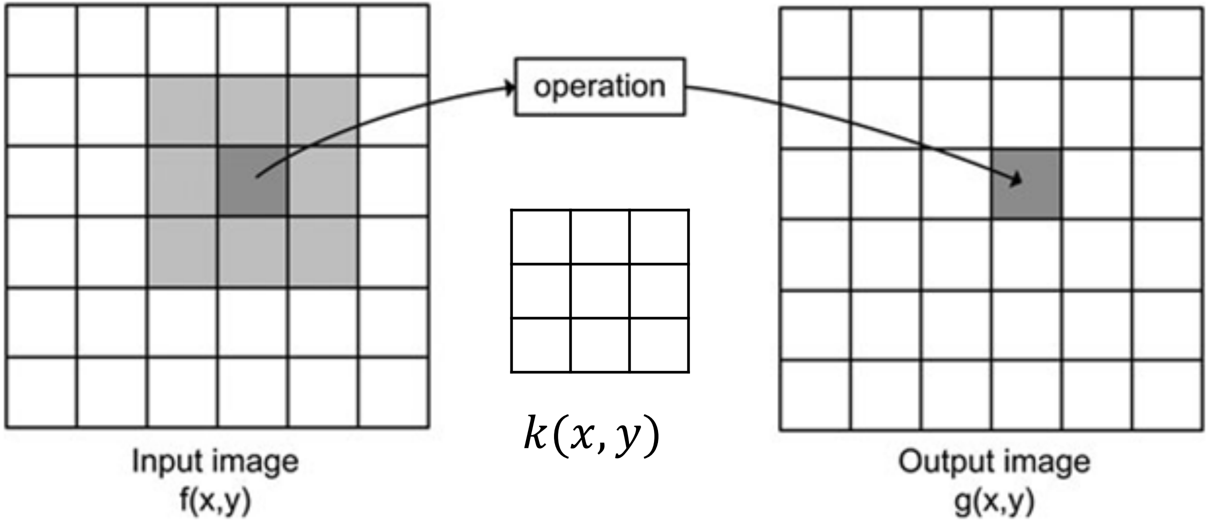


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator



$k(x, y)$

$$g(x, y) = \sum_{v} \sum_{u} k(u, v) f(x - u, y - v)$$

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

(filter, kernel)

Input image * Weights → Output image

Input image:

4	5	7	6	6
3	2	8	0	7
6	7	7	1	5
3	0	1	1	1
4	3	2	1	7

*

Weights:

0	0	0
1	0	1
0	0	0

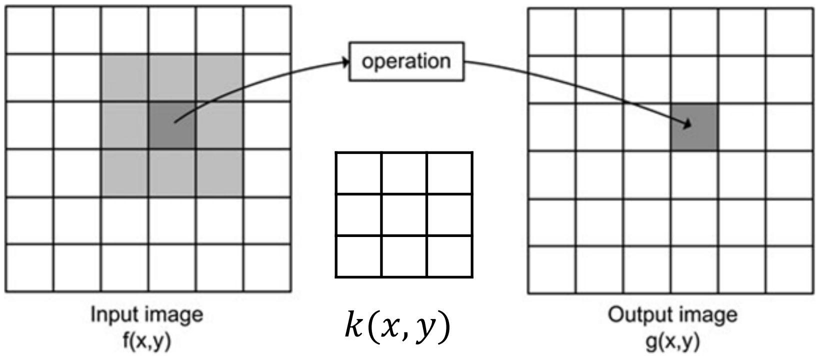
→

Output image:

	11	2	15	
	13	8	12	
	4			

http://www.cs.virginia.edu/~vicente/recognition/animation.gif

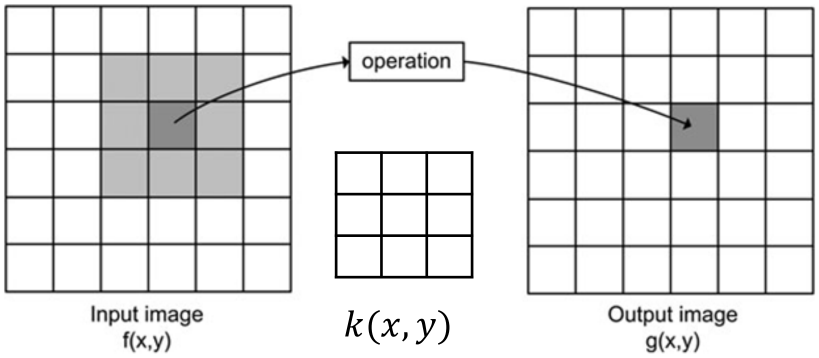
Image filtering: Convolution operator
e.g. mean filter



$$k(x,y) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator
e.g. mean filter



$k(x,y)$ =

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: e.g. Mean Filter

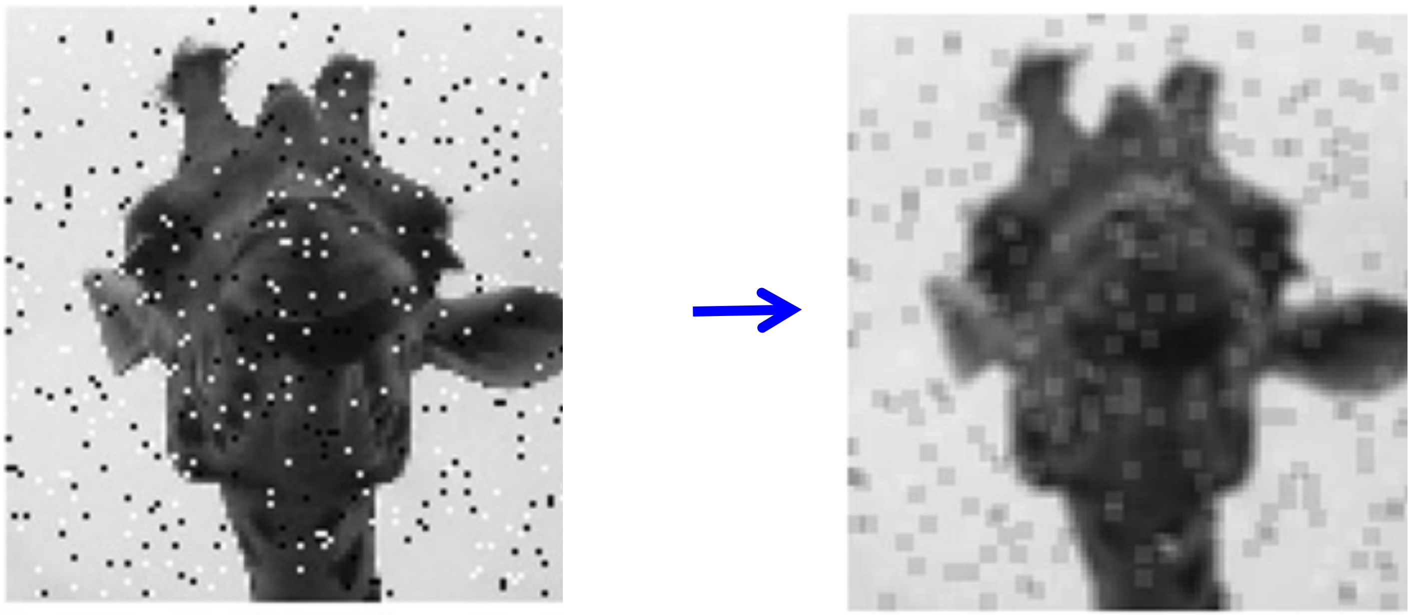
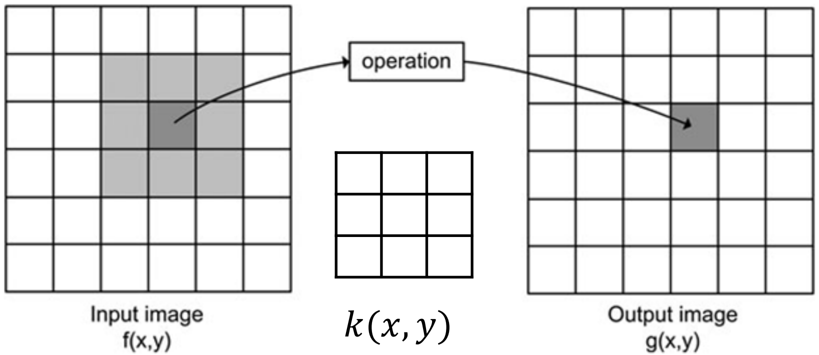


Image filtering: Convolution operator
e.g. gaussian filter (gaussian blur)



$k(x, y) =$

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

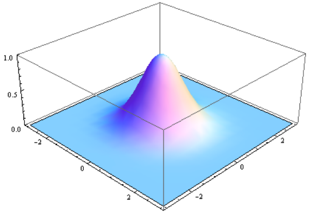
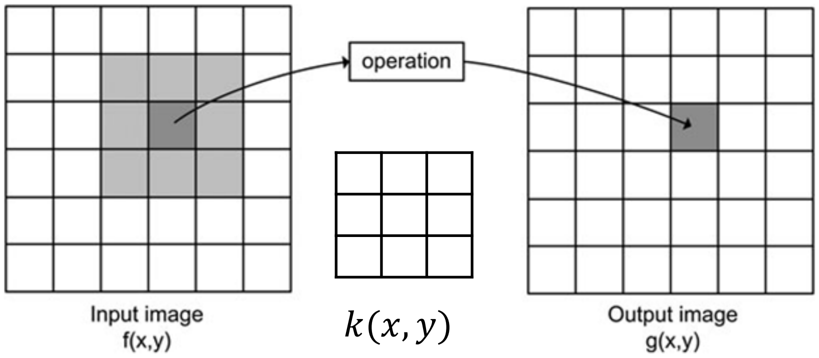


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator
e.g. gaussian filter (gaussian blur)

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

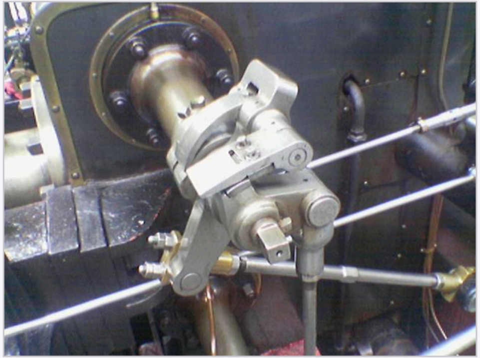
Image filtering: Convolution operator
e.g. sobel operator



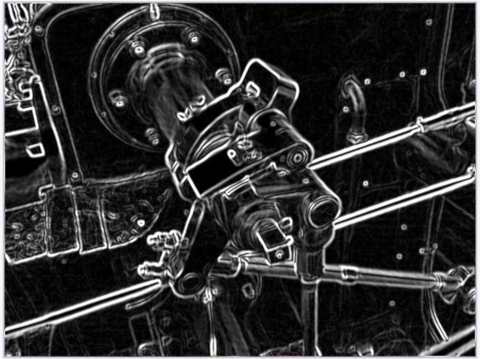
$k(x,y)$ =

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

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/



A color picture of a steam engine



The Sobel operator applied to that image

Next Class: More on Image Filters

Questions?