CS4501: Introduction to Computer Vision

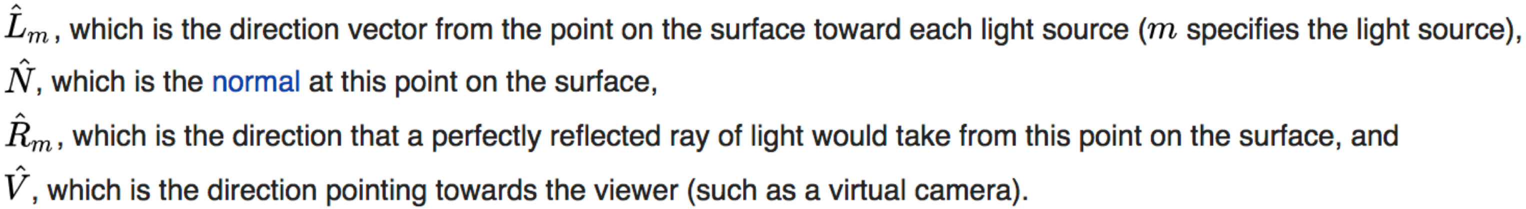
Human Vision and Image Processing

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)
- Introduction to Light (BRDF)

Phong Reflection Model

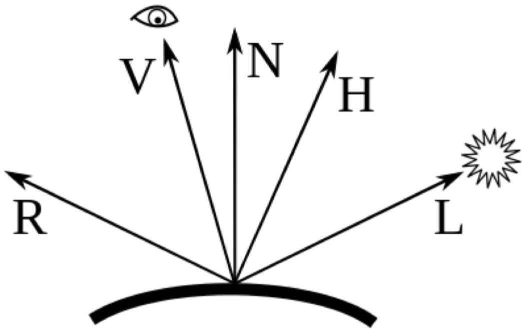


$\hat{L}_m$, which is the direction vector from the point on the surface toward each light source (m specifies the light source),

$\hat{N}$, 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).

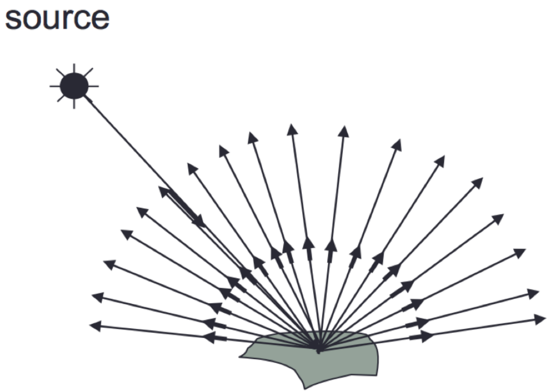


$$I_\text{p} = k_\text{a} i_\text{a} + \sum_{m \in \text{lights}} \left(k_\text{d} (\hat{L}_m \cdot \hat{N}) i_{m,\text{d}} + k_\text{s} (\hat{R}_m \cdot \hat{V})^\alpha i_{m,\text{s}} \right).$$

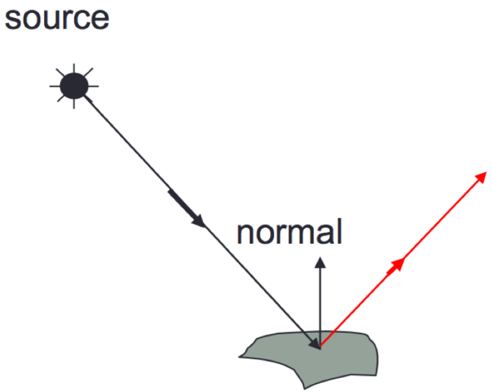
https://en.wikipedia.org/wiki/Phong_reflection_model

Phong Reflection Model

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



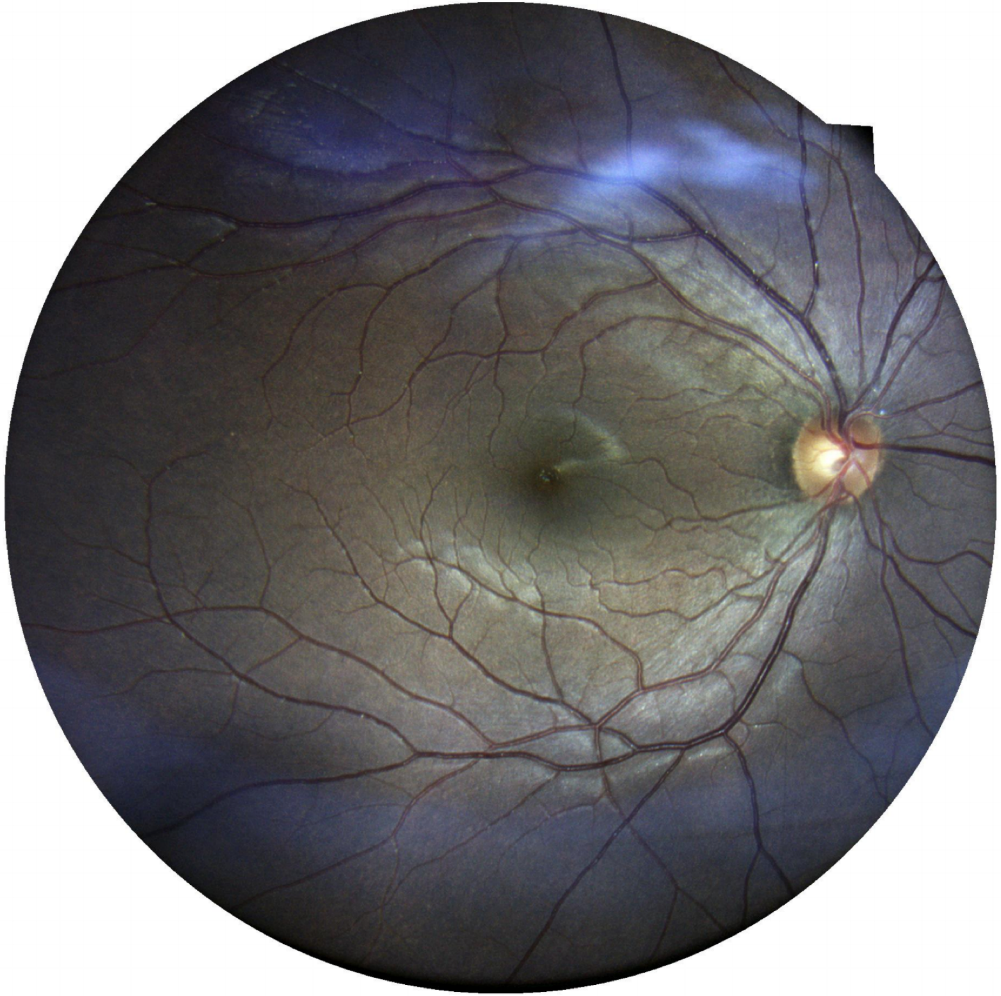
Lambertian



Specular

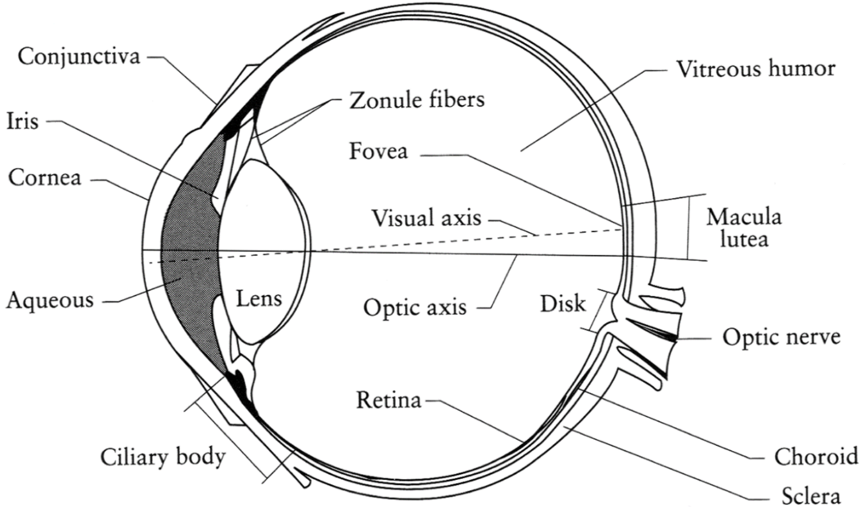
Slide by Aaron Bobick

Our own Camera as a species:
The Human Eye



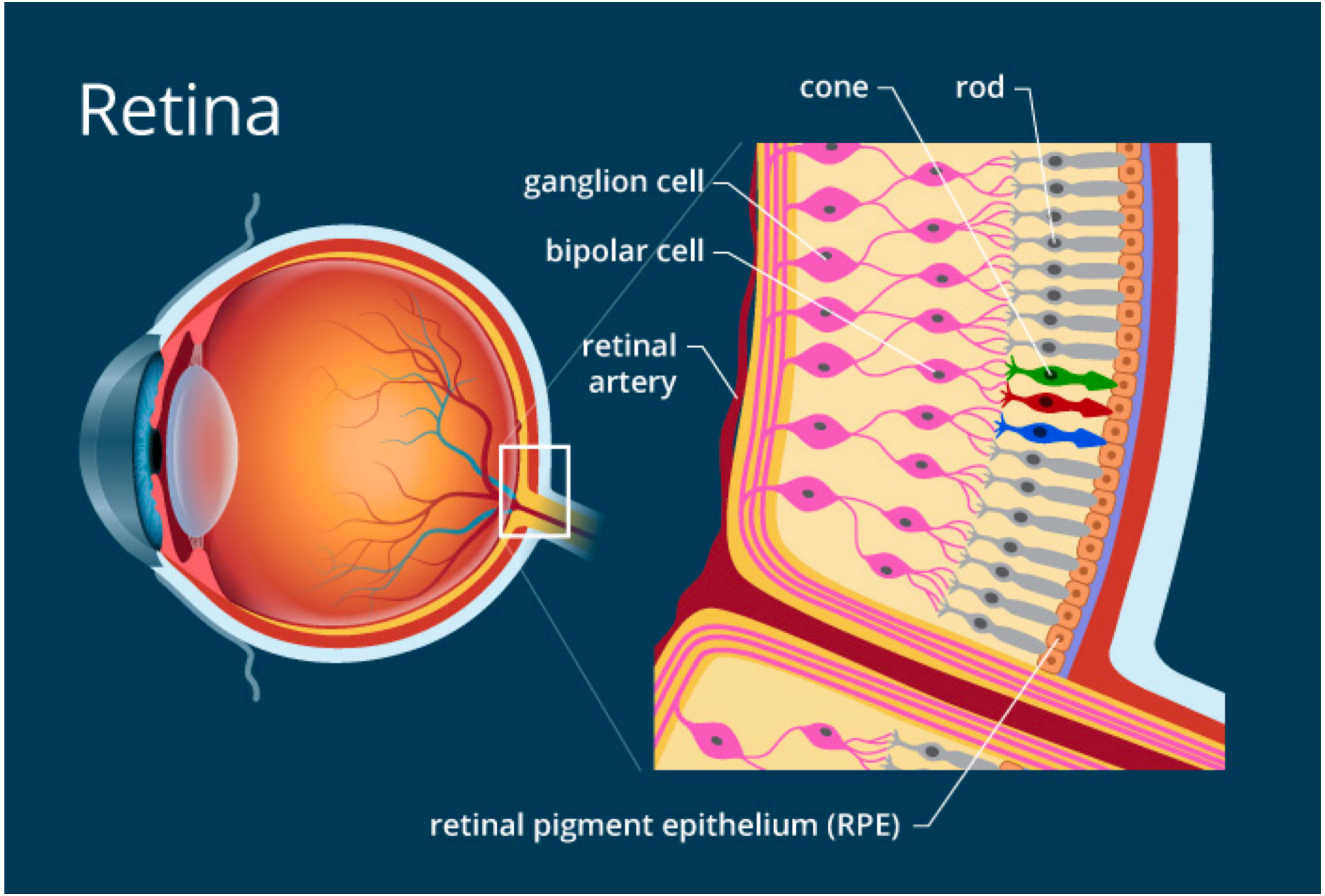
Vicente’s eye

The Eye



- The human eye is sort of 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



https://www.findlight.net/blog/2018/03/16/artificial-photoreceptors/

More about the eye

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

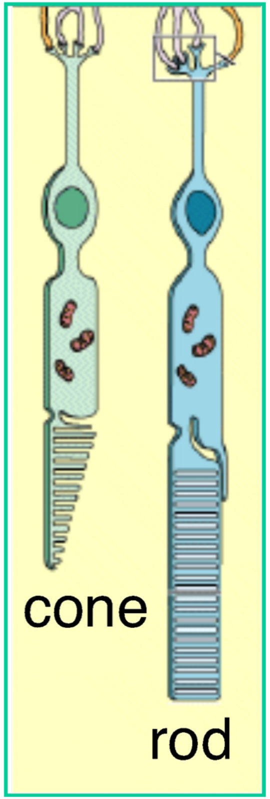
Two types of light-sensitive receptors

Cones

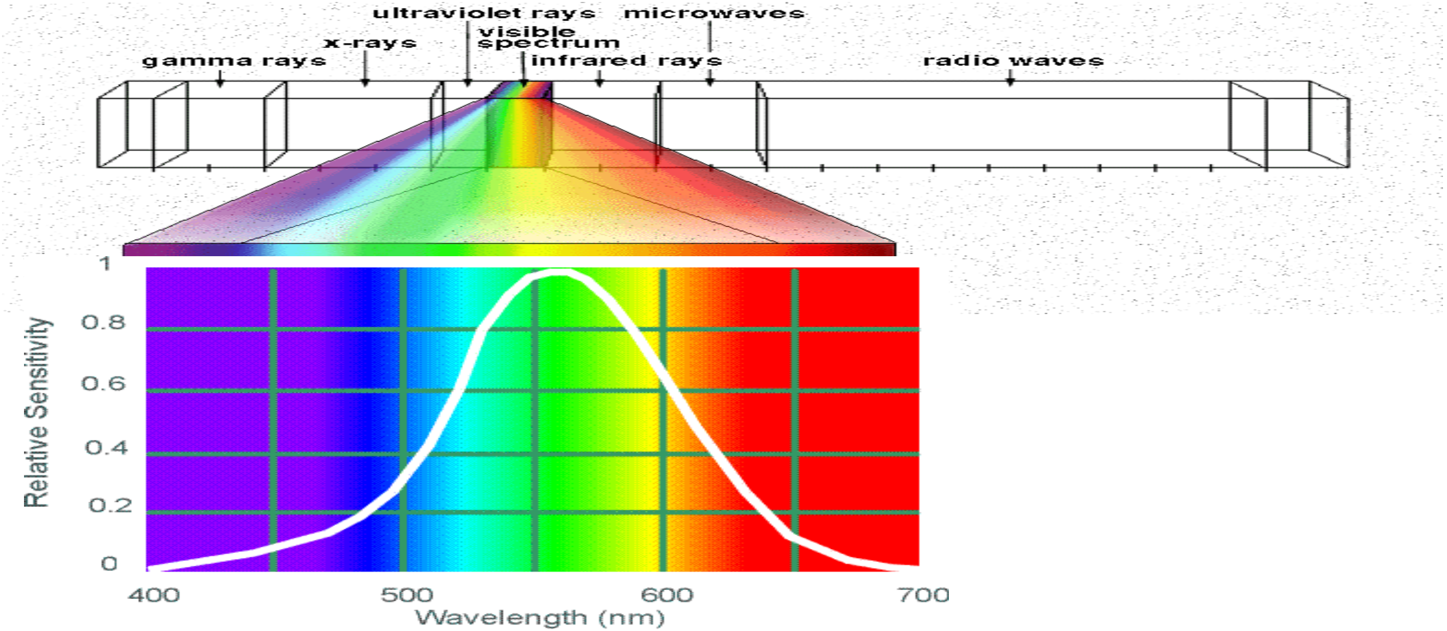
cone-shaped
less sensitive
operate in high light
color vision

Rods

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



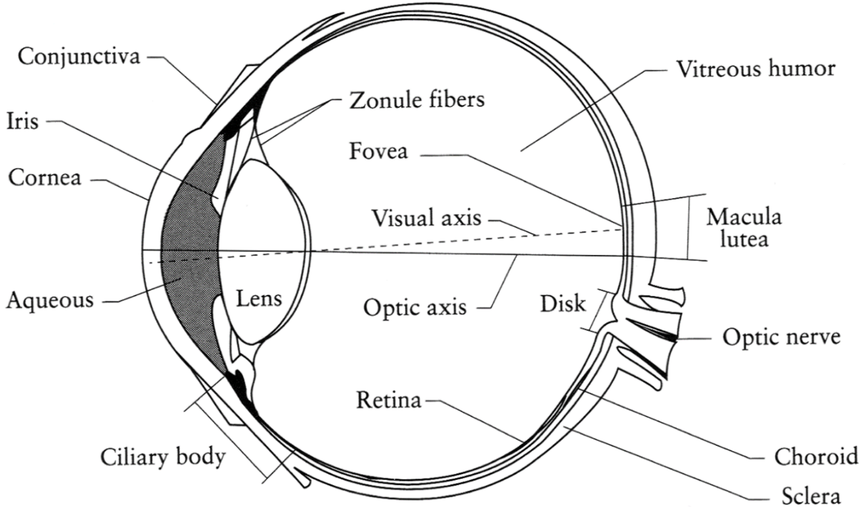
Electromagnetic Spectrum



Human Luminance Sensitivity Function

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

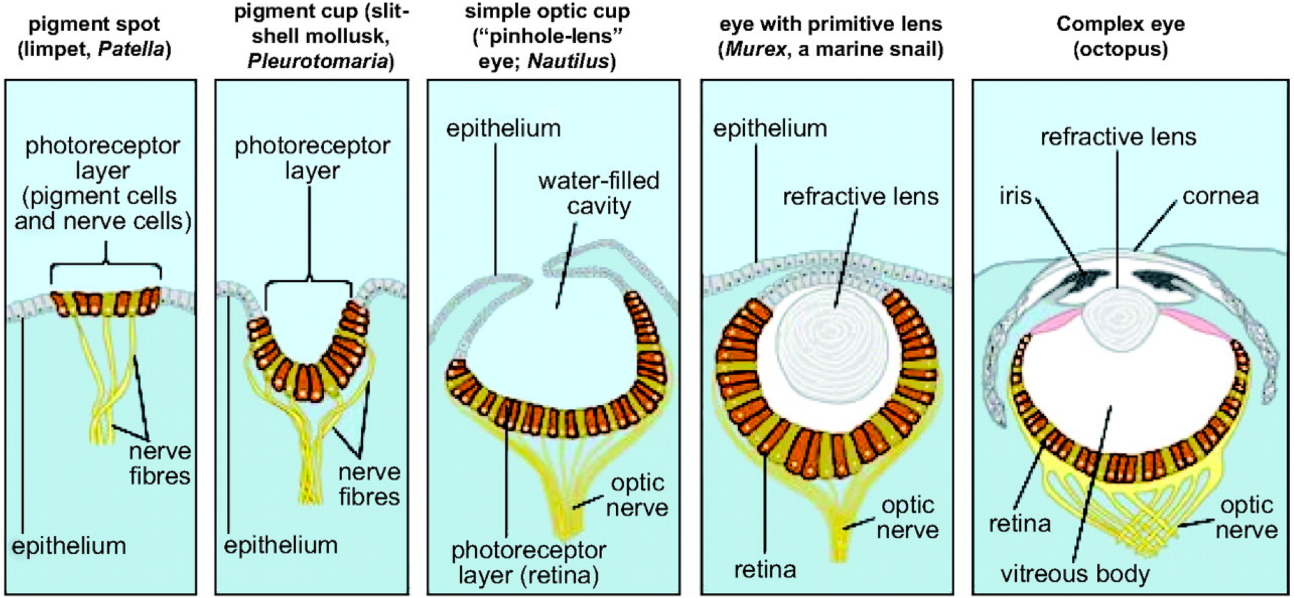
The Eye



- The human eye is sort of 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

Eye Evolution

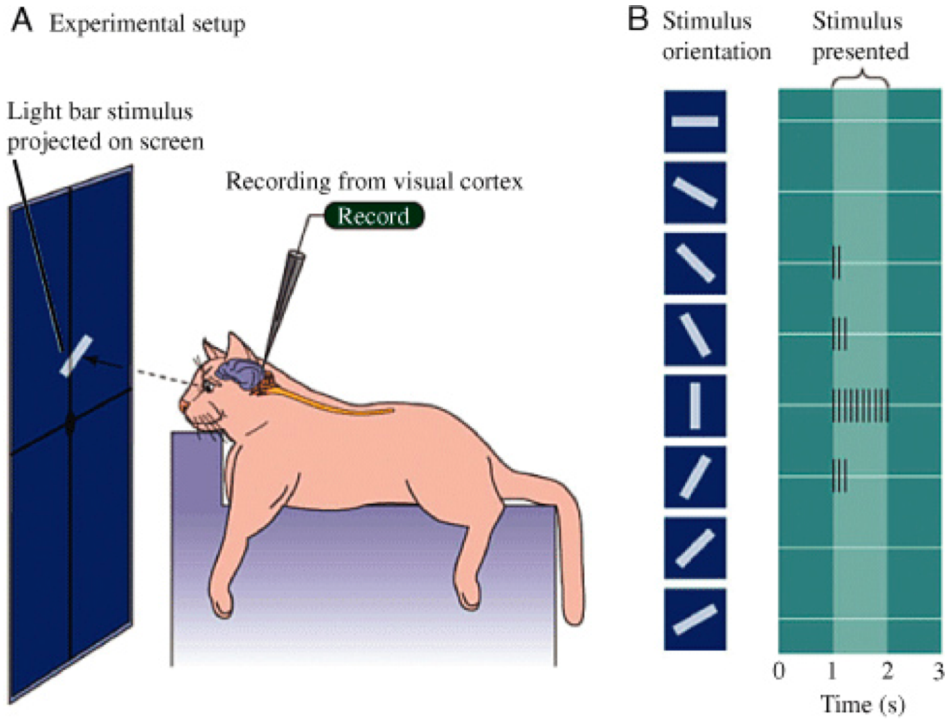

pigment spot (limpet, *Patella*)
photoreceptor layer (pigment cells and nerve cells)
nerve fibres
epithelium
pigment cup (slit-shell mollusk, *Pleurotomaria*)
photoreceptor layer
nerve fibres
epithelium
simple optic cup ("pinhole-lens" eye; *Nautilus*)
epithelium
water-filled cavity
optic nerve
photoreceptor layer (retina)
eye with primitive lens (*Murex*, a marine snail)
epithelium
refractive lens
optic nerve
retina
Complex eye (octopus)
refractive lens
iris
cornea
retina
optic nerve
vitreous body

Ayala 2007. Darwin's greatest discovery: Design without designer. Proceedings of the National Academy of Sciences. USA.

What the Frog's Eye Tells the Frog's Brain*

J. Y. LETTVIN†, H. R. MATURANA‡, W. S. McCULLOCH||, SENIOR MEMBER, IRE,
AND W. H. PITTS||

Hubel and Wiesel



https://www.youtube.com/watch?v=IOHayh06LJ4

http://www.informit.com/articles/article.aspx?p=1431818

Image Processing & Image Filtering

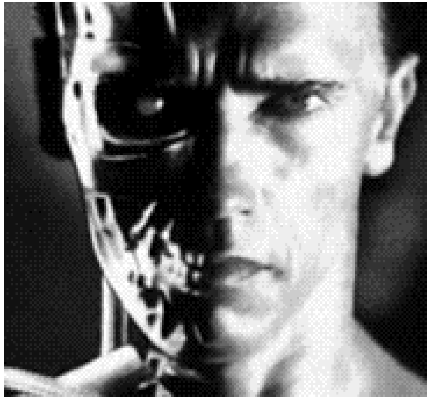
Reminder of what is an image for a computer.

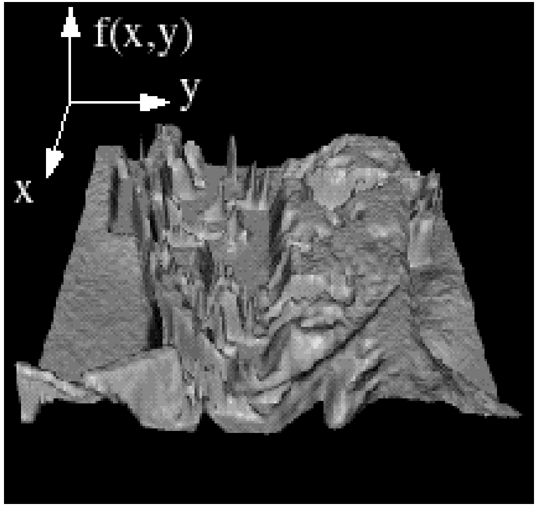
La Gare Montparnasse, 1895

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

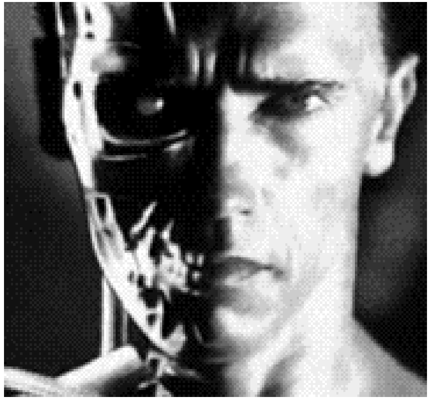
$$z = f(x, y)$$

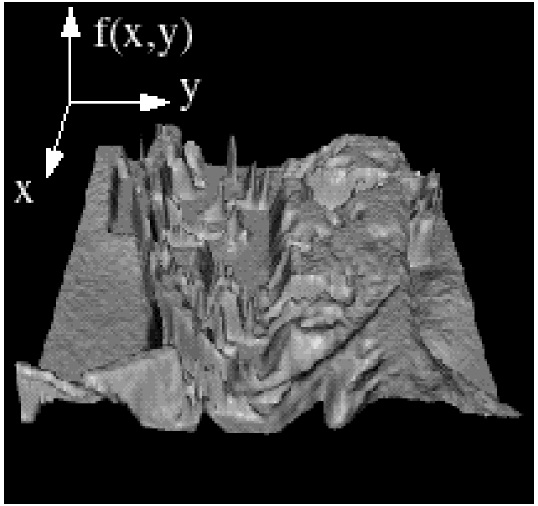




Images as Functions

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



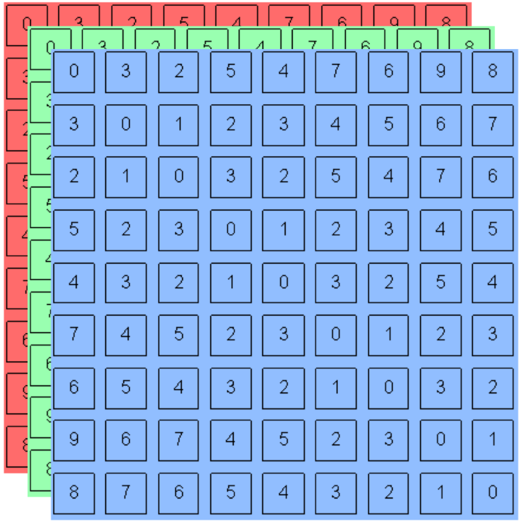


- The domain of x and y is [0, img-width) and [0 and img-height)
- x, and y are discretized into integer values.

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

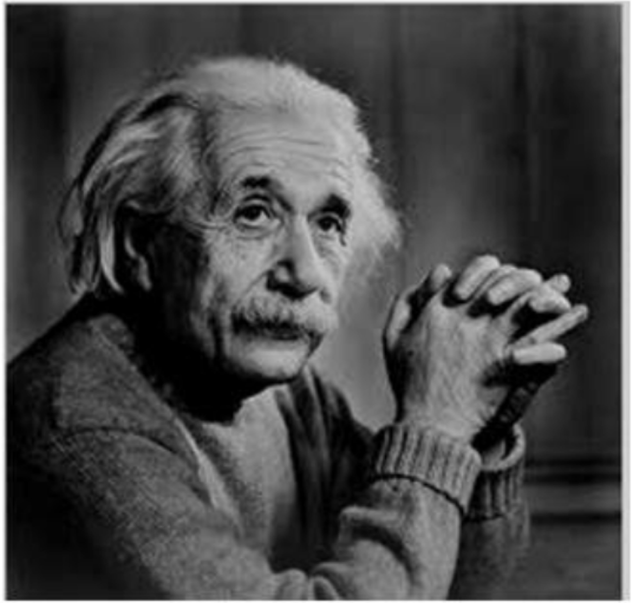


0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

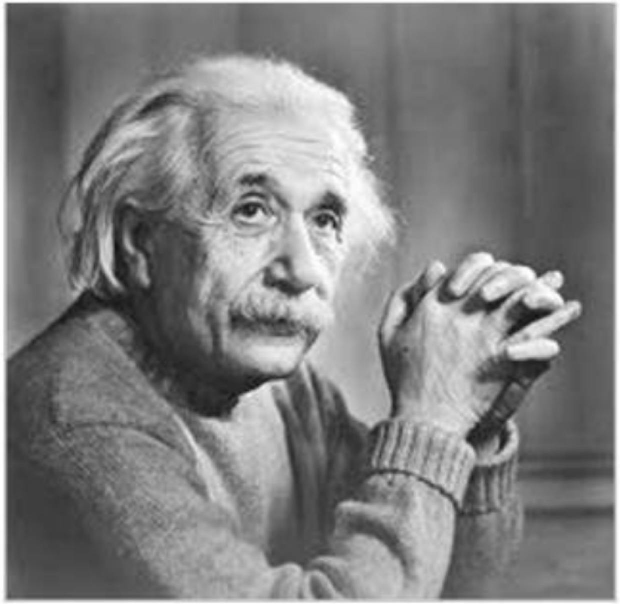
channel x height x width

Basic Image Processing

I



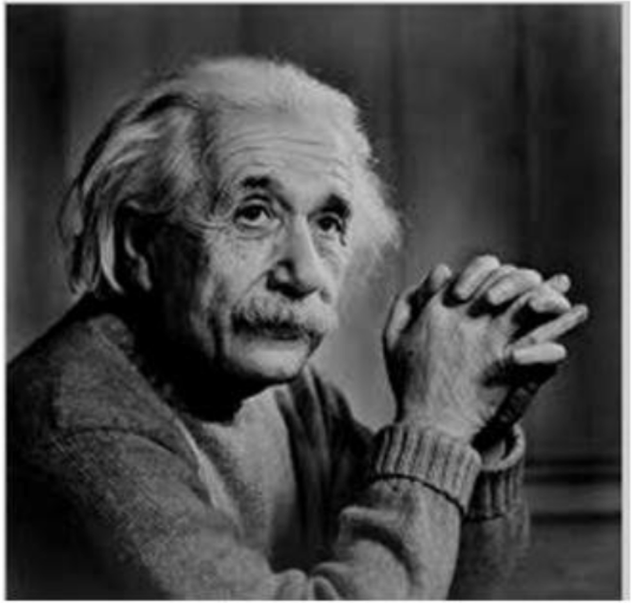
αI



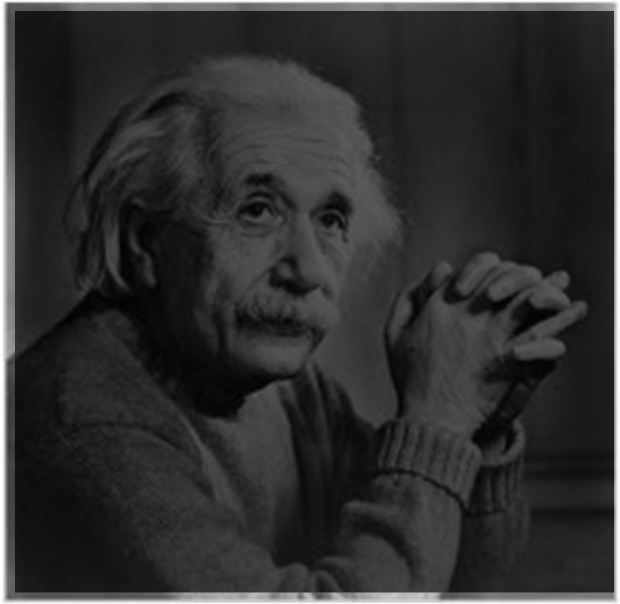
$\alpha > 1$

Basic Image Processing

I

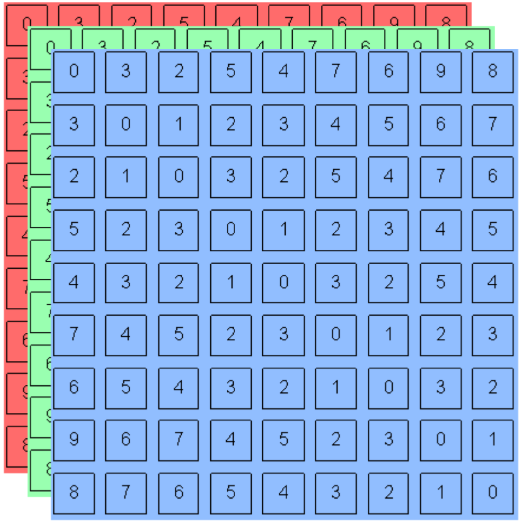


αI



$0 < \alpha < 1$

Color Images as Tensors



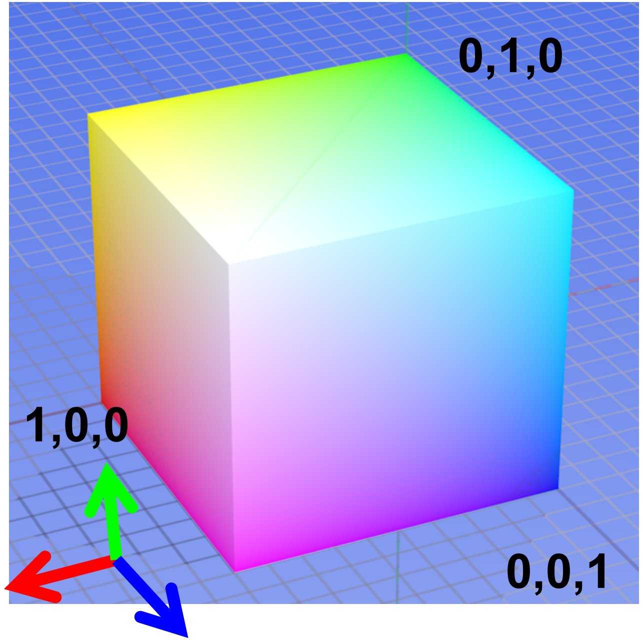
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

$channel\ x\ height\ x\ width$

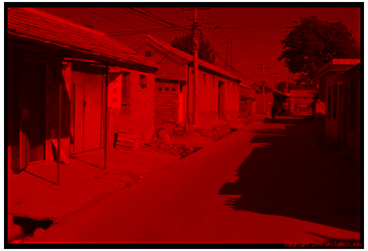
Channels are usually RGB: Red, Green, and Blue

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

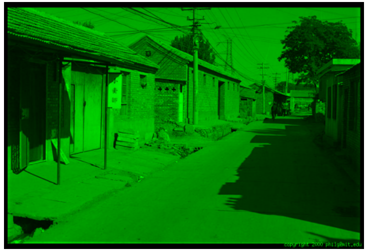
Color spaces: RGB



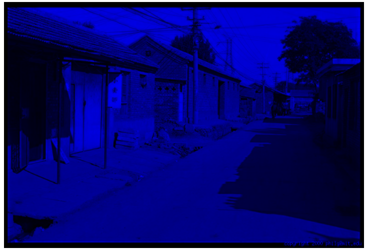
Some drawbacks
- Strongly correlated channels
- Non-perceptual



R
(G=0,B=0)



G
(R=0,B=0)



B
(R=0,G=0)

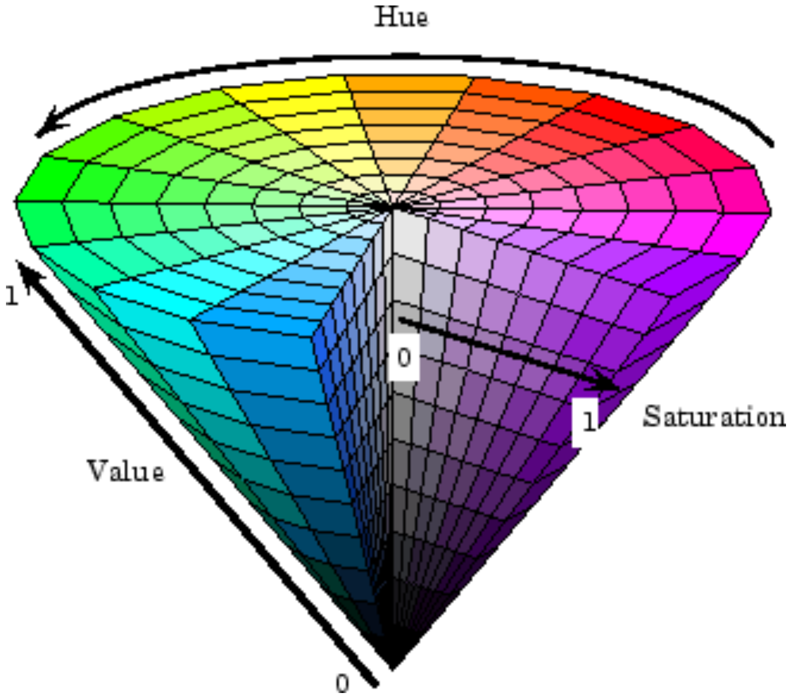
Default color space

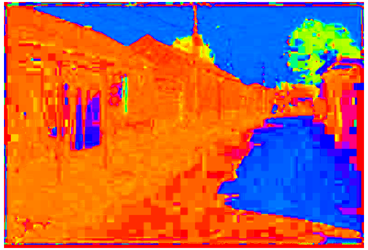
Slide by James Hays

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

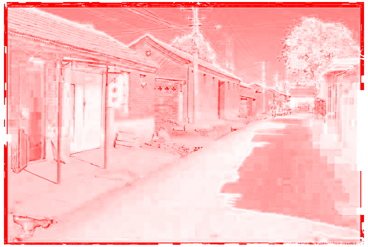
Color spaces: HSV

Intuitive color space

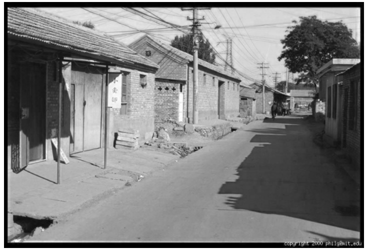




H
(S=1,V=1)



S
(H=1,V=1)

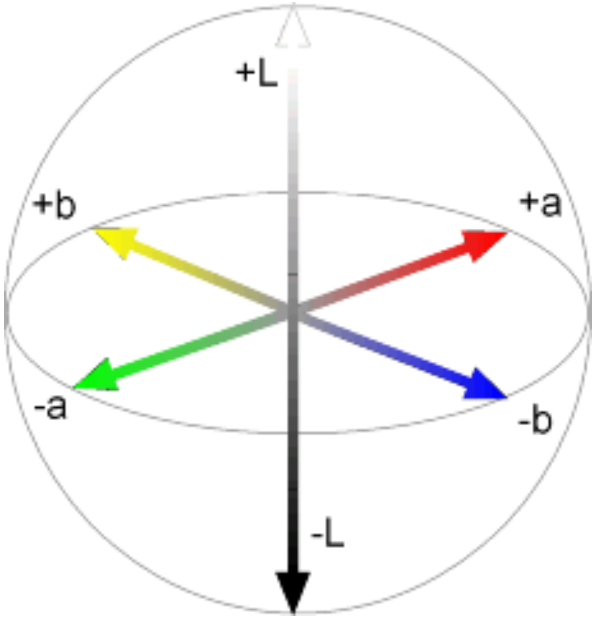


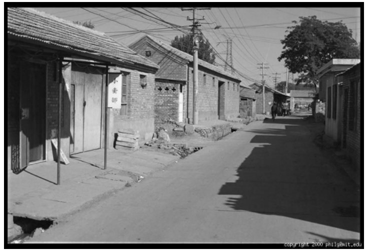
V
(H=1,S=0)

Slide by James Hays

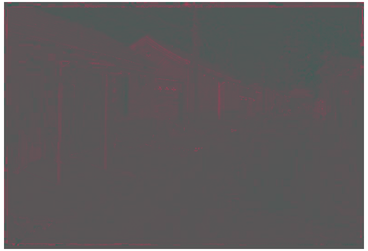
Color spaces: L*a*b*

“Perceptually uniform”* color space

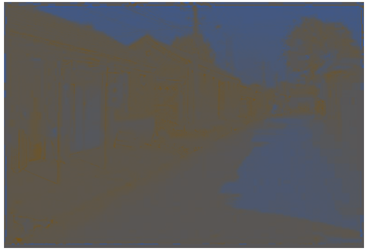




L
(a=0,b=0)



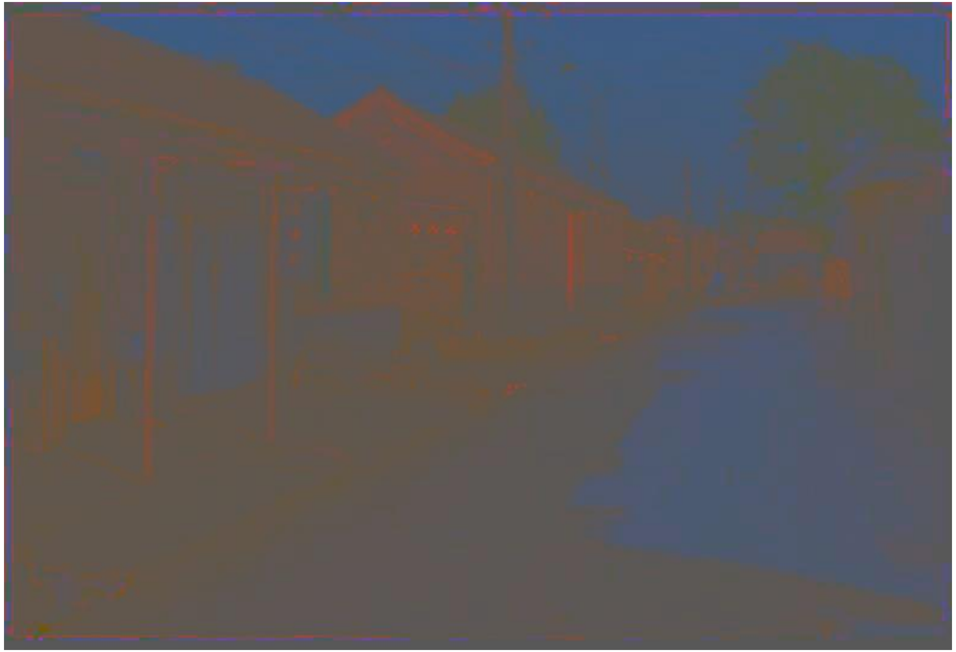
a
(L=65,b=0)



b
(L=65,a=0)

Slide by James Hays

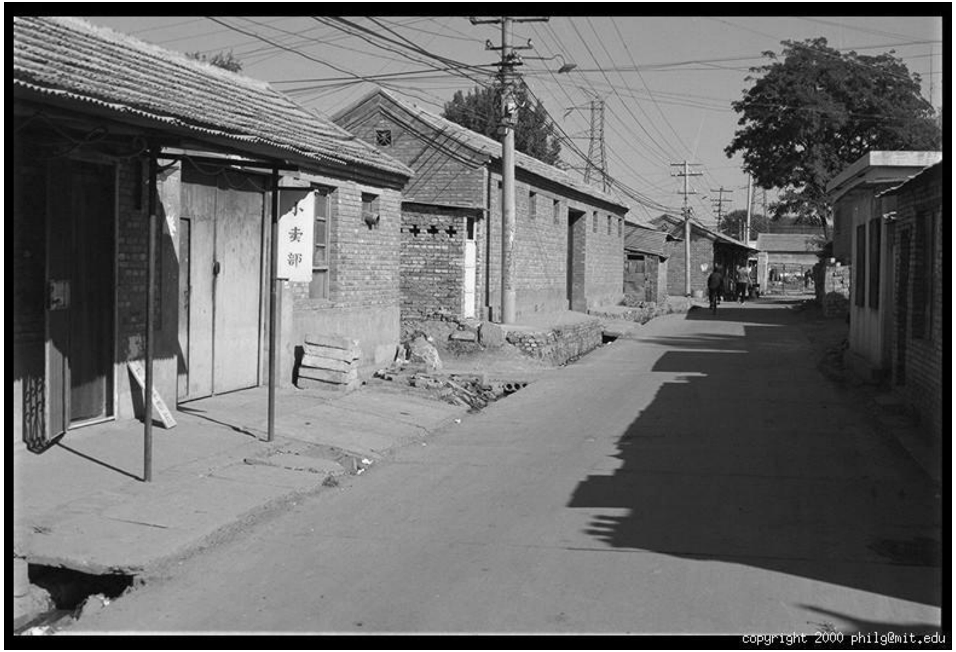
Most information in intensity



Only color shown – constant intensity

Slide by James Hays

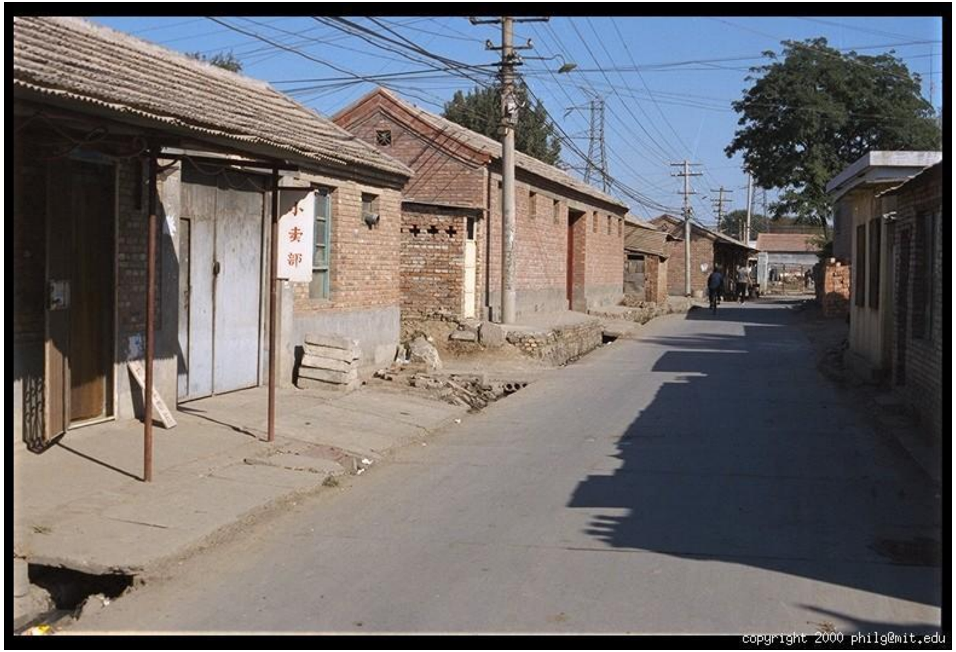
Most information in intensity



Only intensity shown – constant color

Slide by James Hays

Most information in intensity

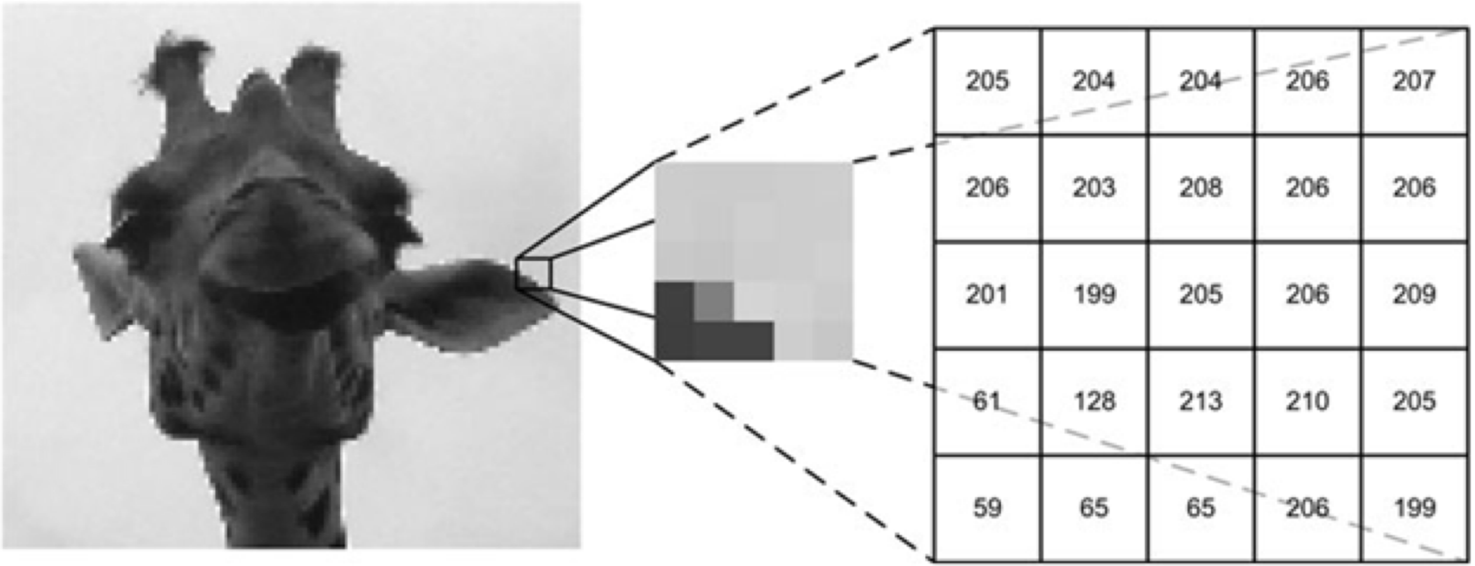


Original image

Slide by James Hays

Image filtering

Image filtering



205	204	204	206	207
206	203	208	206	206
201	199	205	206	209
61	128	213	210	205
59	65	65	206	199

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

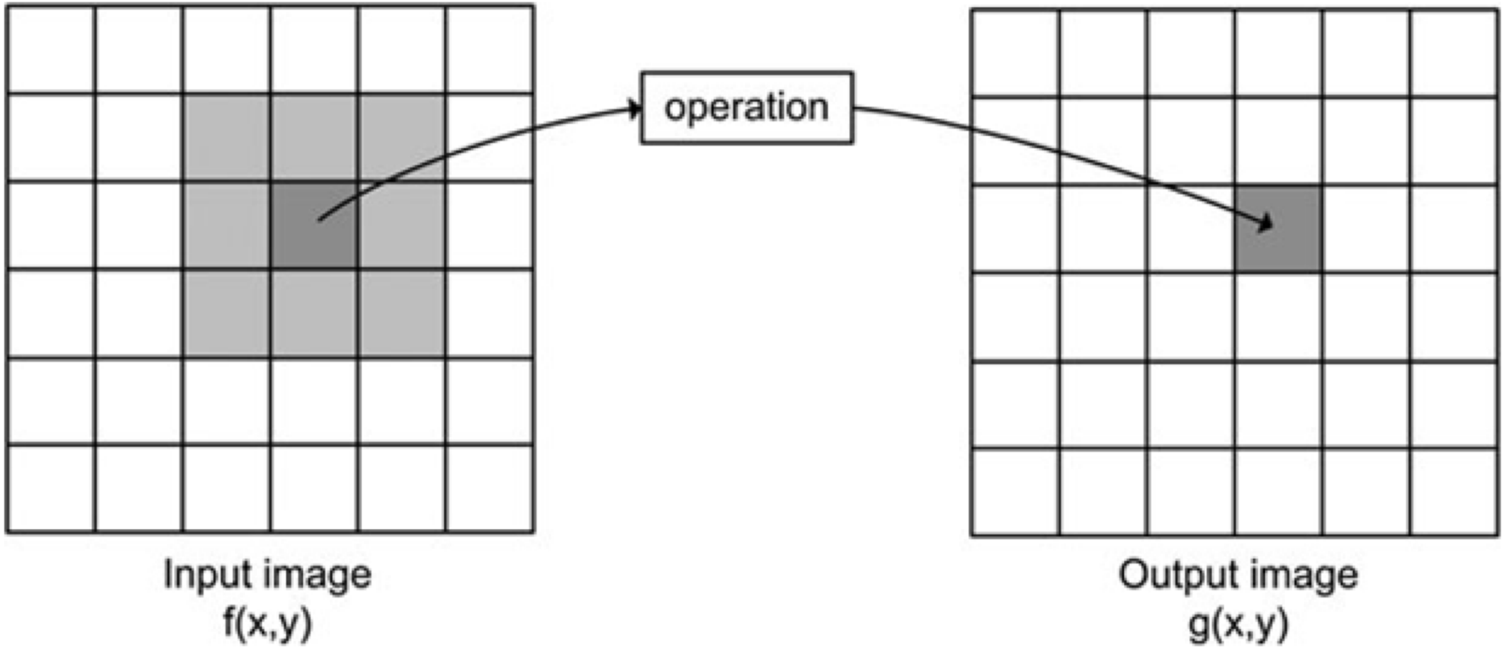
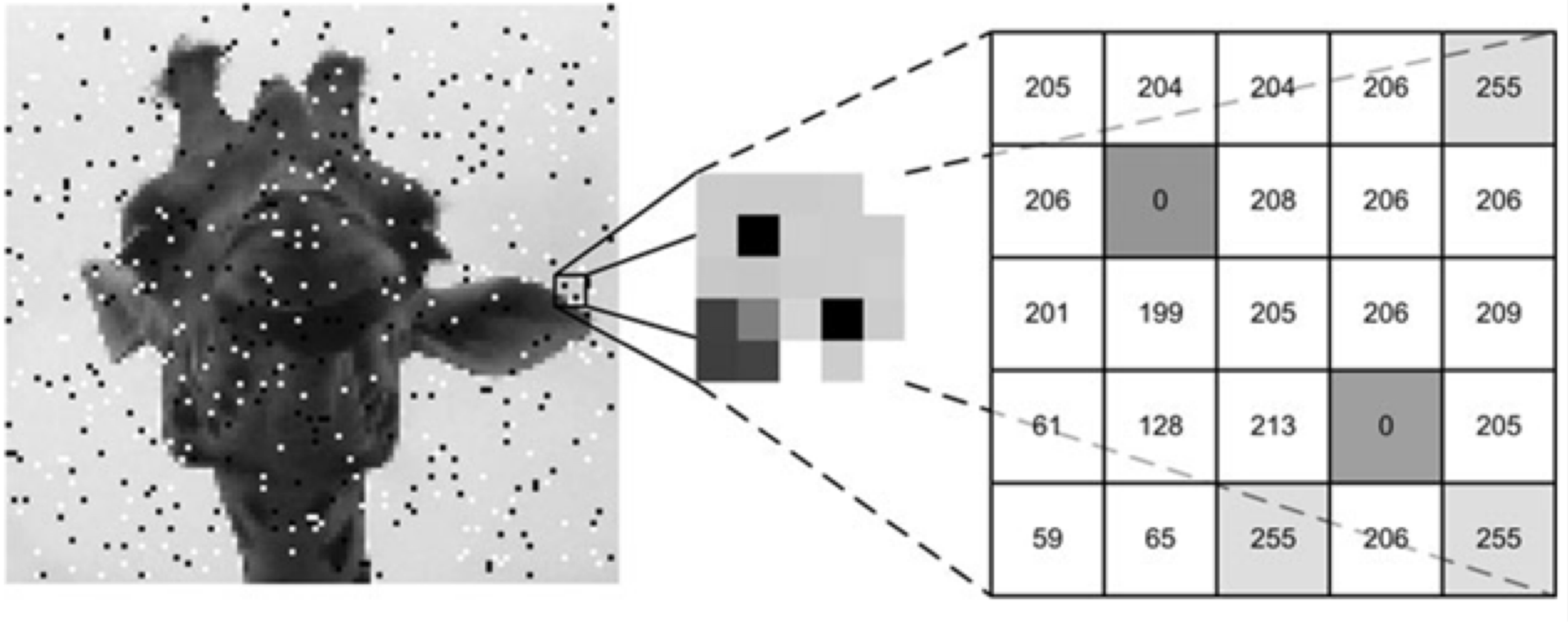


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

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

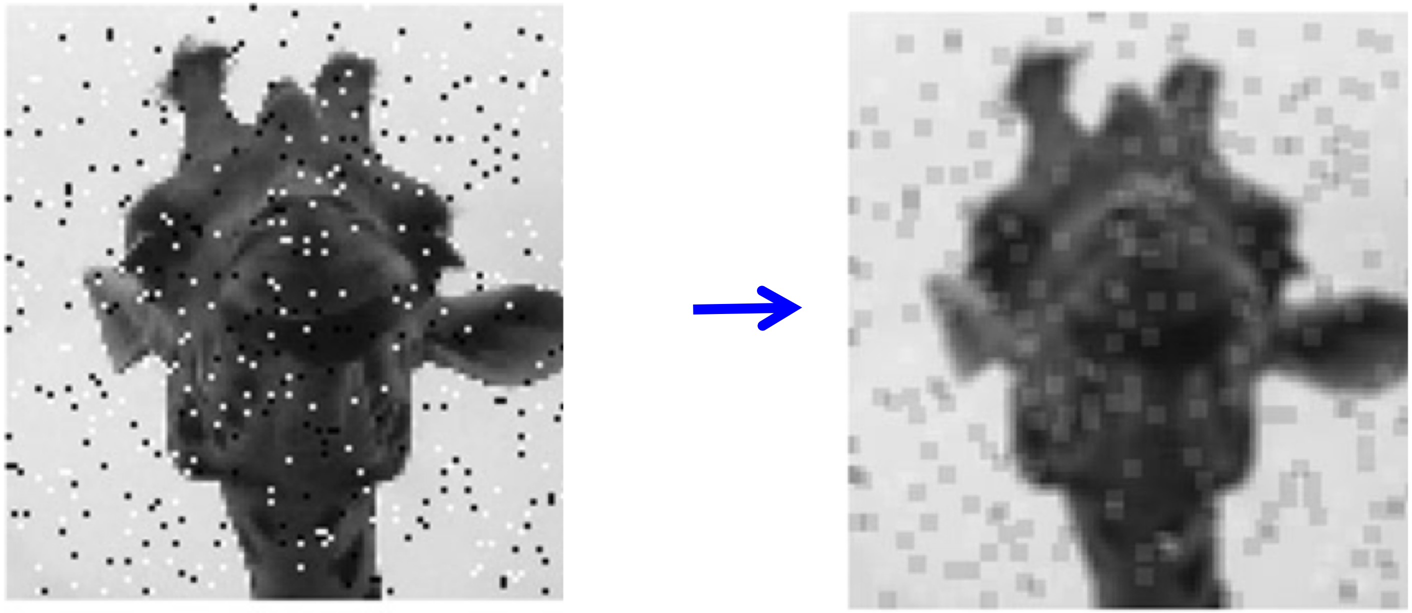


Image filtering: e.g. Median Filter

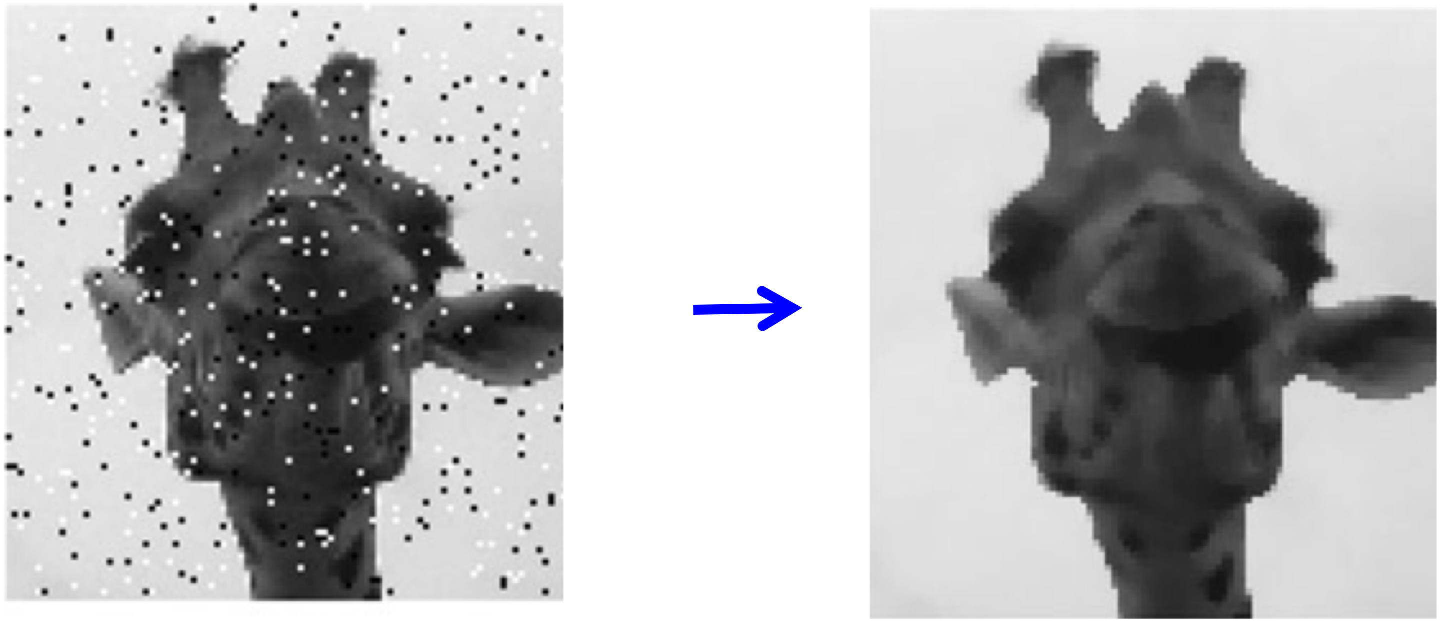
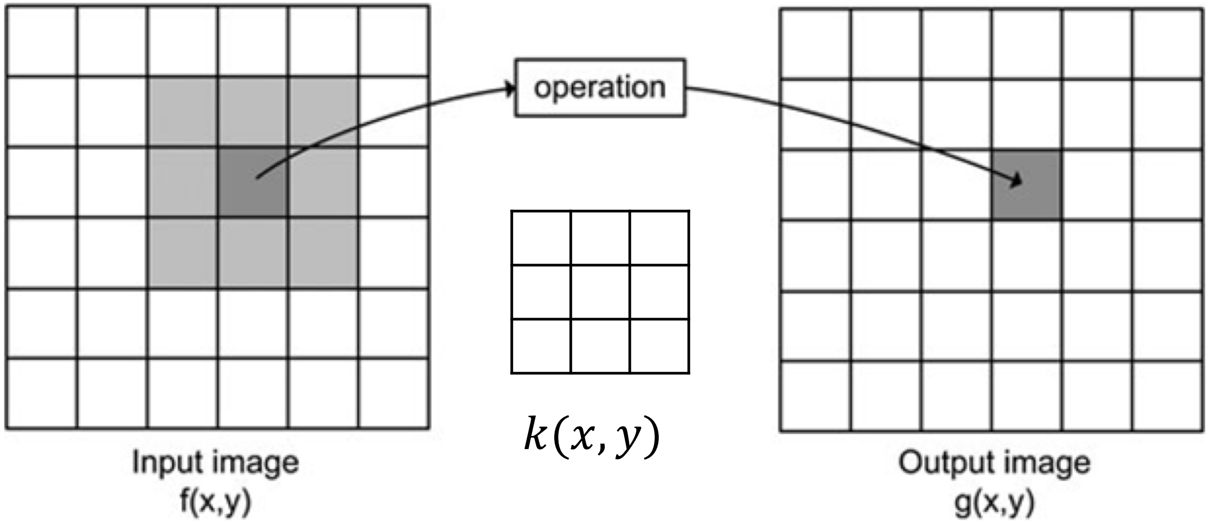


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator

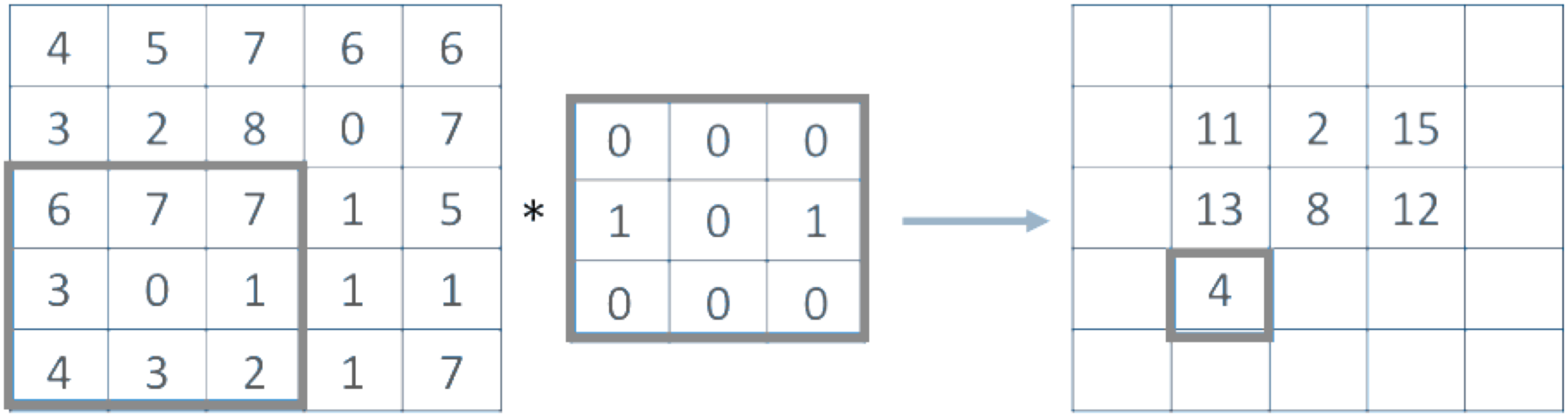


$$g(x,y) = \sum_{v}\sum_{u} k(u,v) f(x - u, y - v)$$

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



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

*

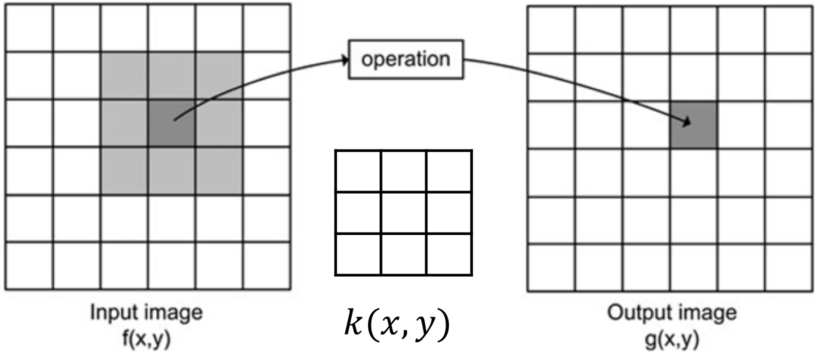
0	0	0
1	0	1
0	0	0

→

	11	2	15	
	13	8	12	
	4			

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

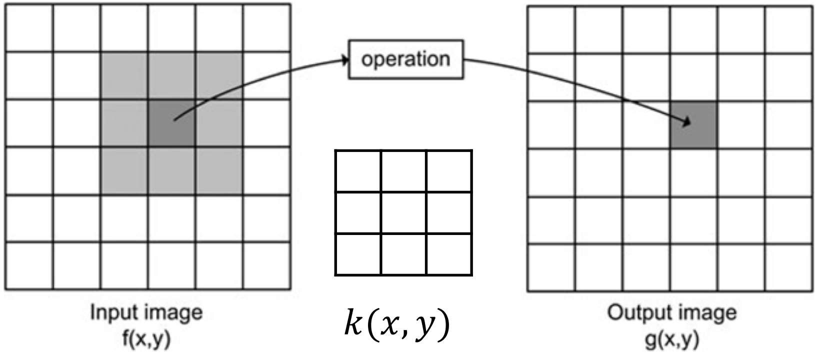
Image filtering: Convolution operator
e.g. mean filter



$$k(x,y) = \begin{array}{|c|c|c|}\hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline \end{array}$$

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



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

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: e.g. Mean Filter

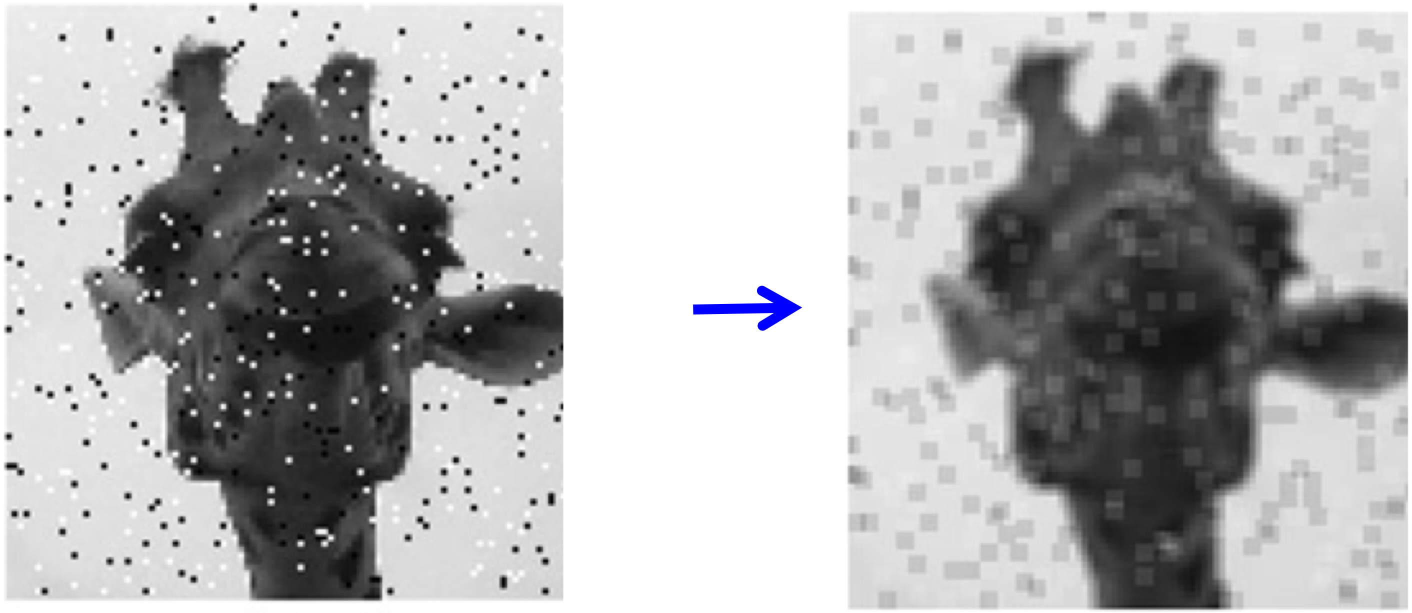
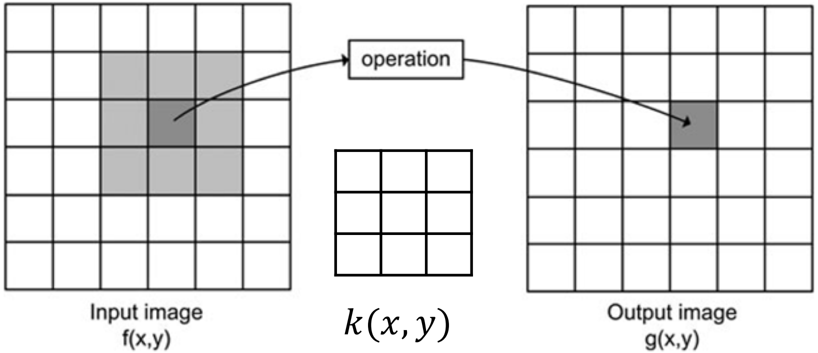


Image filtering: Convolution operator
e.g. gaussian filter (gaussian blur)



$$k(x,y) = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

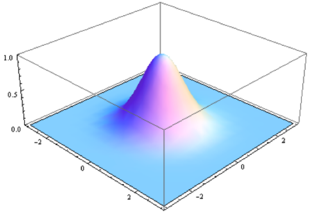
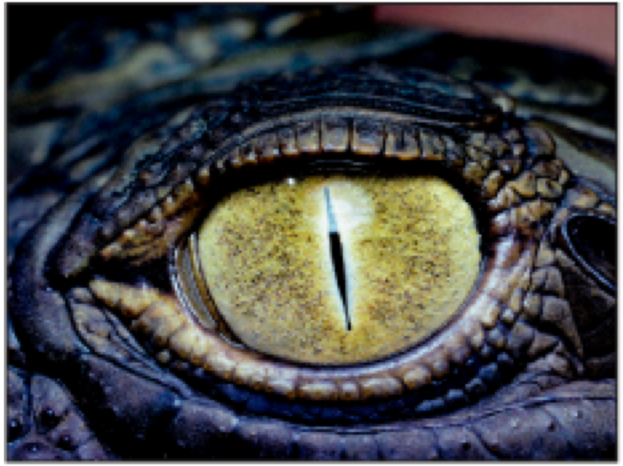


Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

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



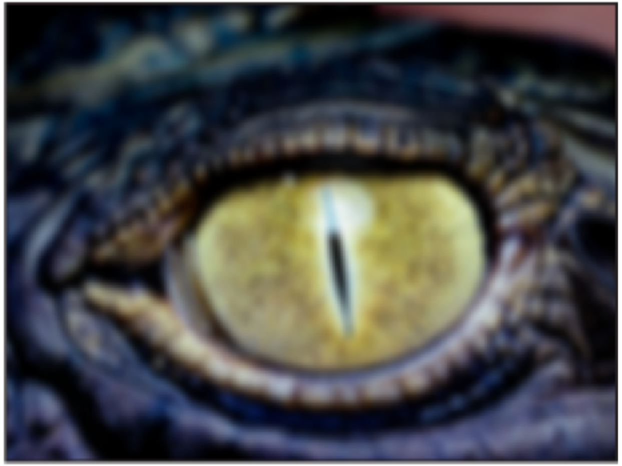
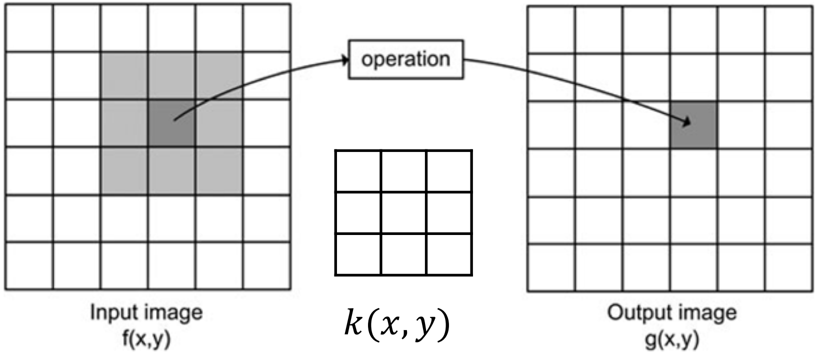


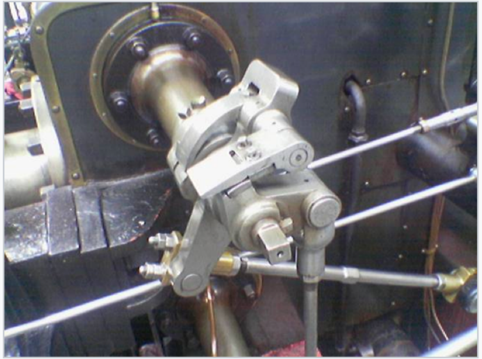
Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Image filtering: Convolution operator
e.g. sobel operator

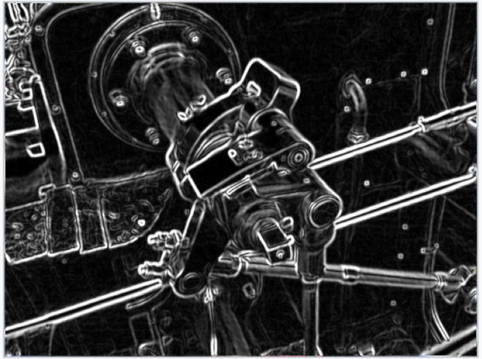


$$k(x,y) = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

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?