計算認知神經科學
計算視覺

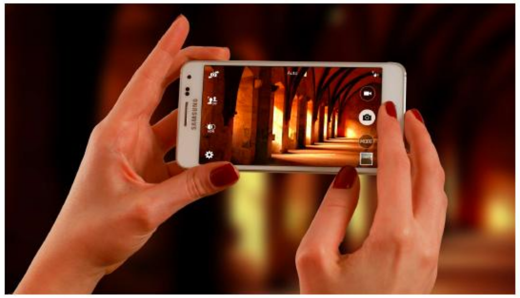
Color Ambiguity: “The Dress”

Blue / Black dress under yellow light?

White / Gold dress under blue light?

Computer Vision is everywhere!





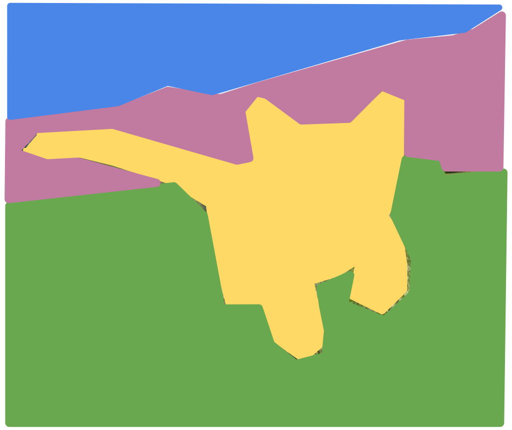


電腦視覺應用

Classification

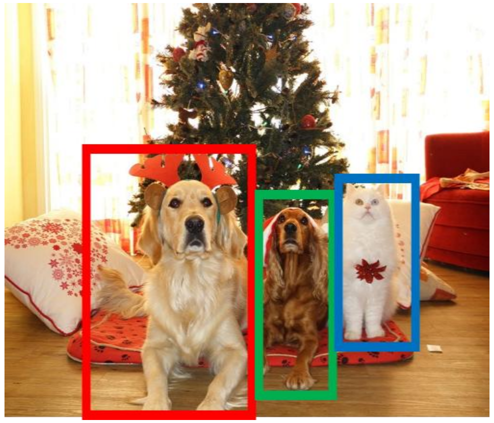
CAT

Semantic Segmentation



GRASS, CAT, TREE, SKY

Object Detection



DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

No spatial extent

No objects, just pixels

Multiple Objects

Image basics (color)

Light

- What determines the color of a pixel?

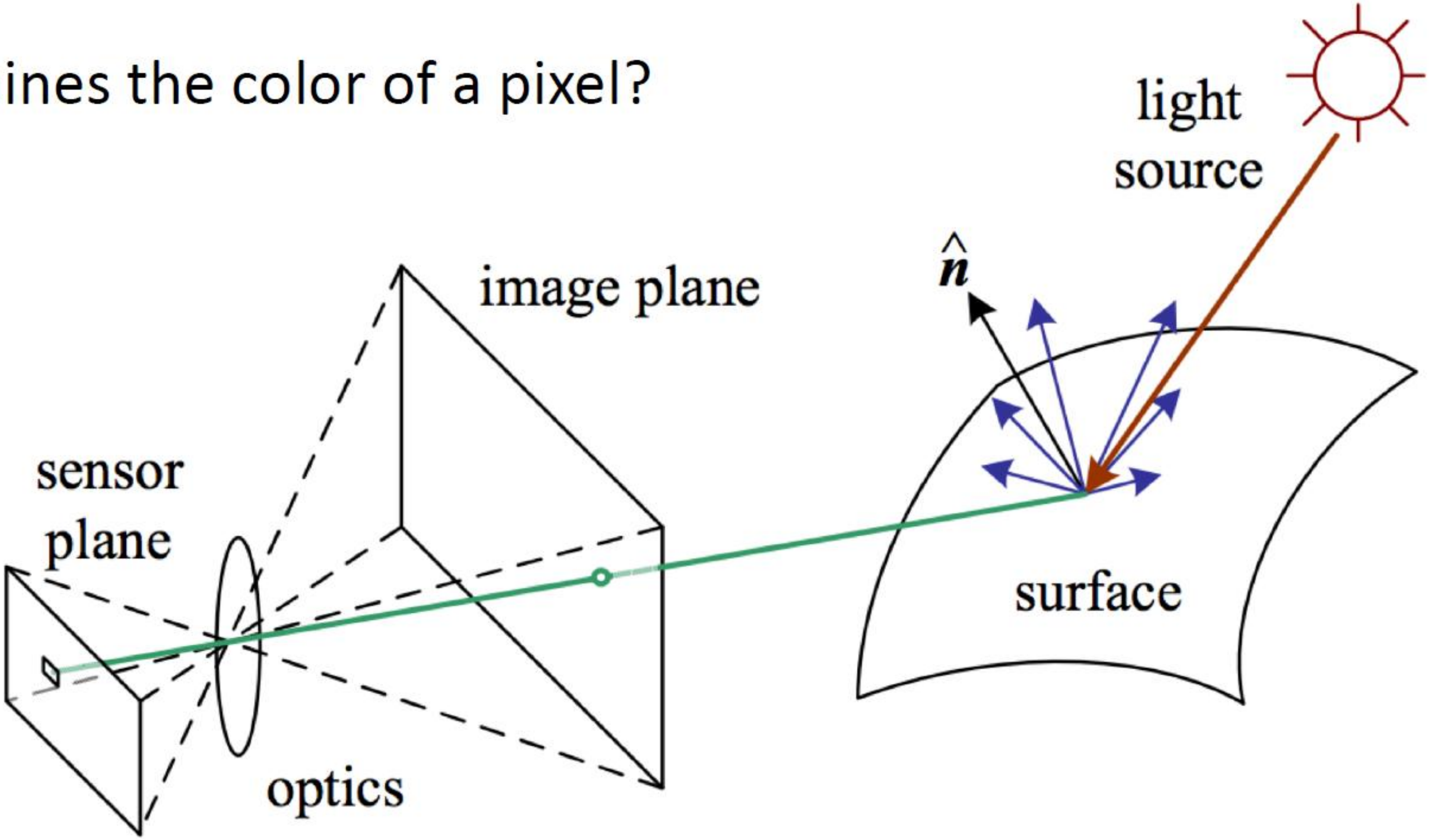
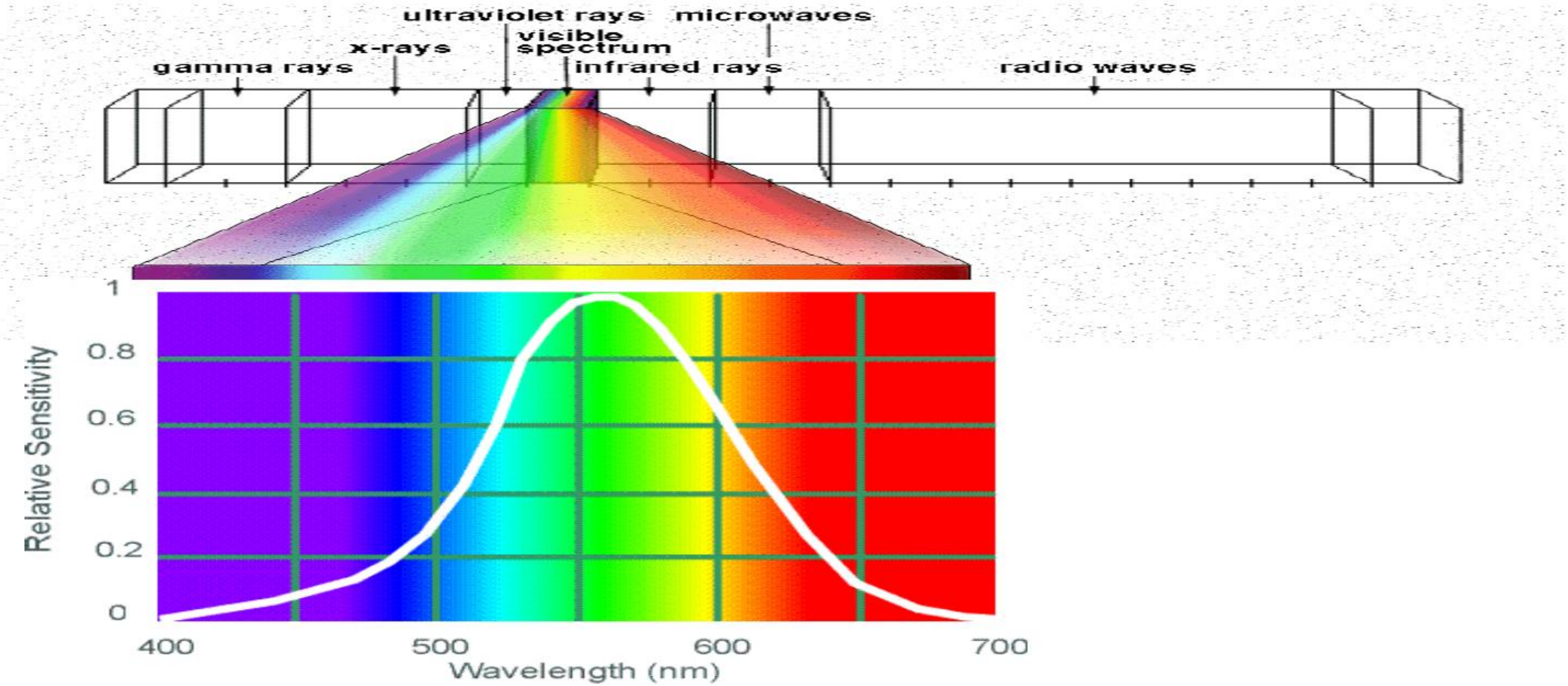


Figure from Szeliski

Electromagnetic Spectrum



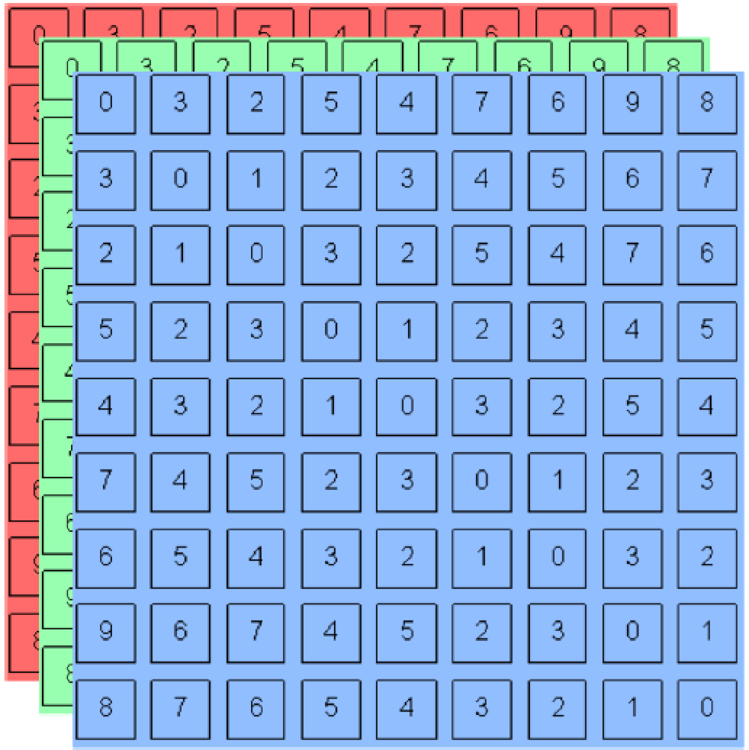
Human Luminance Sensitivity Function

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

What is an image?

A (color) image is a 3D tensor of numbers.

Color Images as Tensors

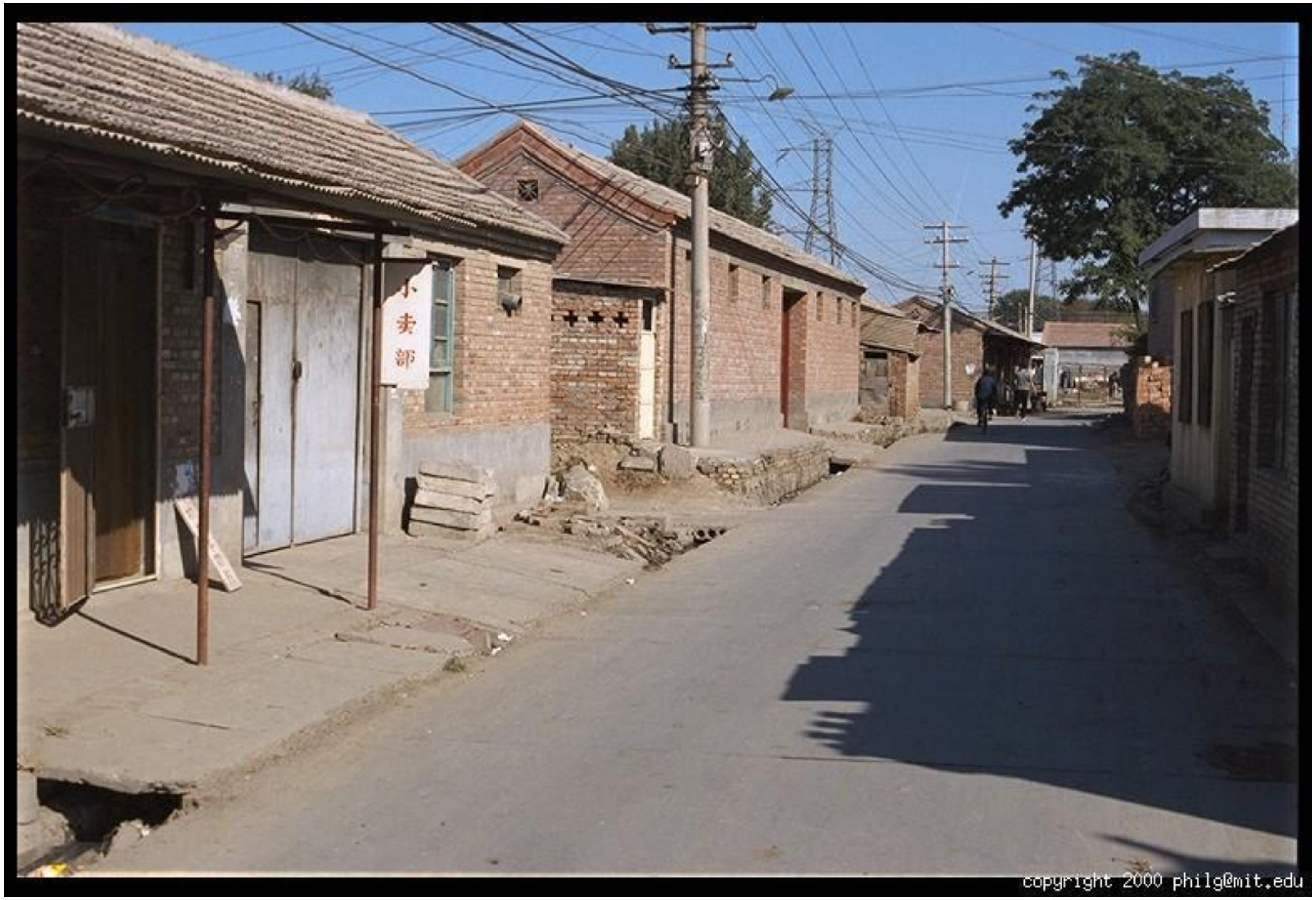


channel x height x width

Channels are usually RGB: Red, Green, and Blue

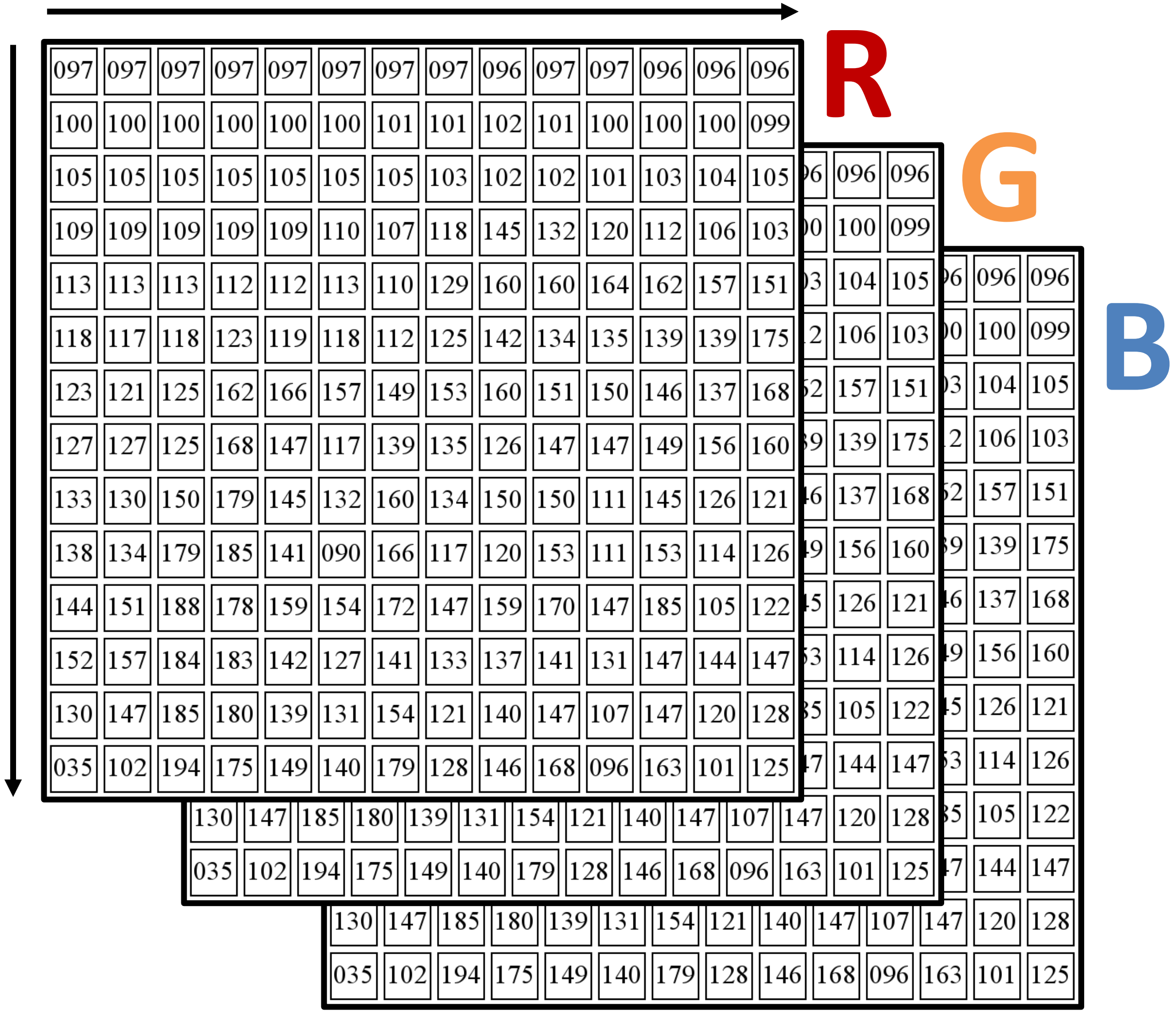
Other color spaces: HSV, HSL, LUV, XYZ, Lab, CMYK, etc

Color Images



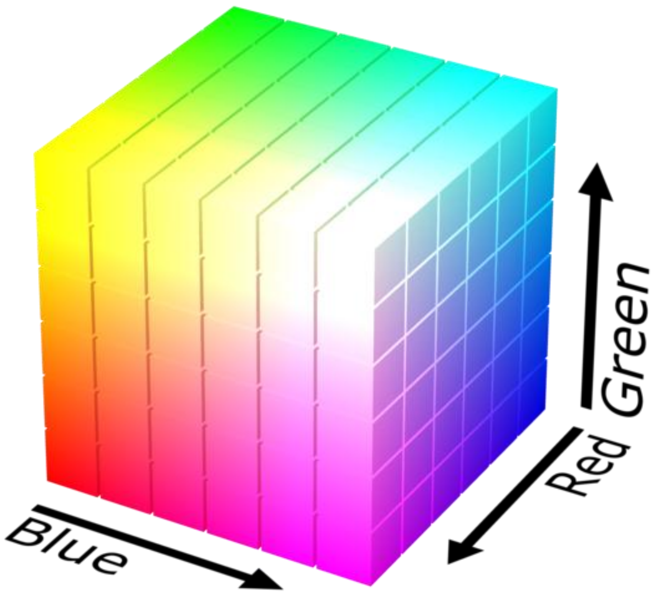
Slide Credit: J. Hays

Images in Python



One Option: RGB

Pros	Cons
1. Simple	1. Distances don't make sense
2. Common	2. Correlated



Slide Credit: J. Hays, RGB cube: https://en.wikipedia.org/wiki/RGB_color_model

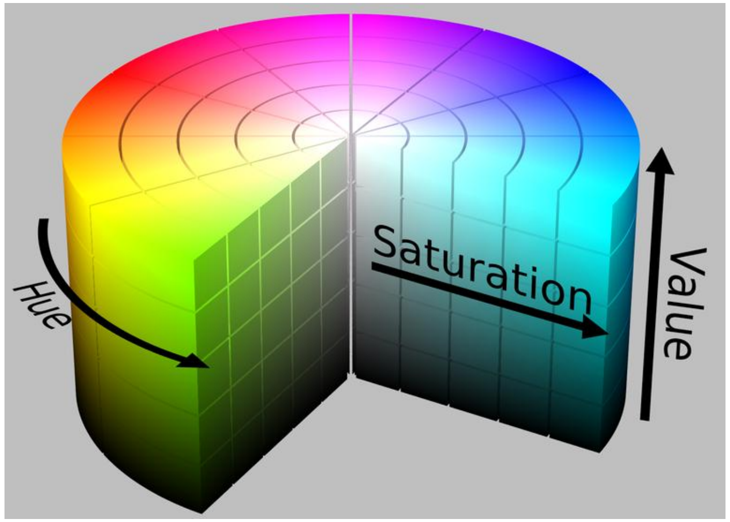
Another Option: HSV

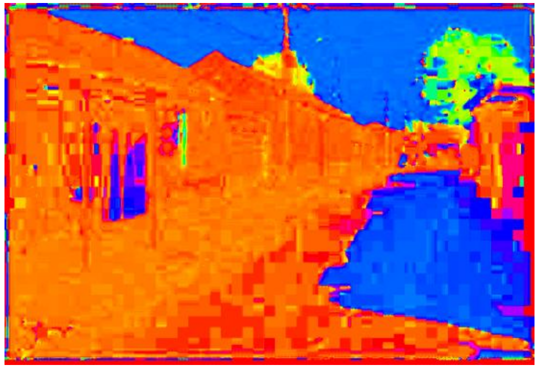
Pros

1. Intuitive for picking colors
2. Sort of common
3. Fast to convert

Cons

1. Not as good as other better spaces





H
(S=1,V=1)

S
(H=1,V=1)

V
(H=1,S=0)

Slide Credit: J. Hays, HSV cylinder: https://en.wikipedia.org/wiki/HSL_and_HSV

Another Option: YUV / YCbCr

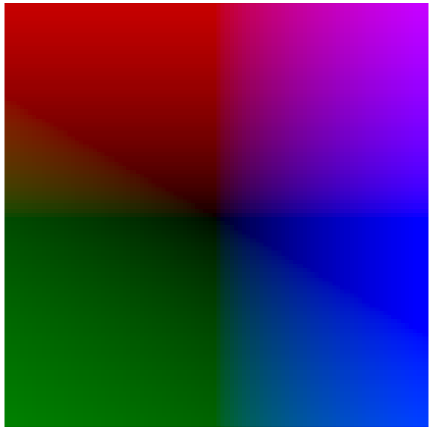
Pros

1. Great for transmission / compression

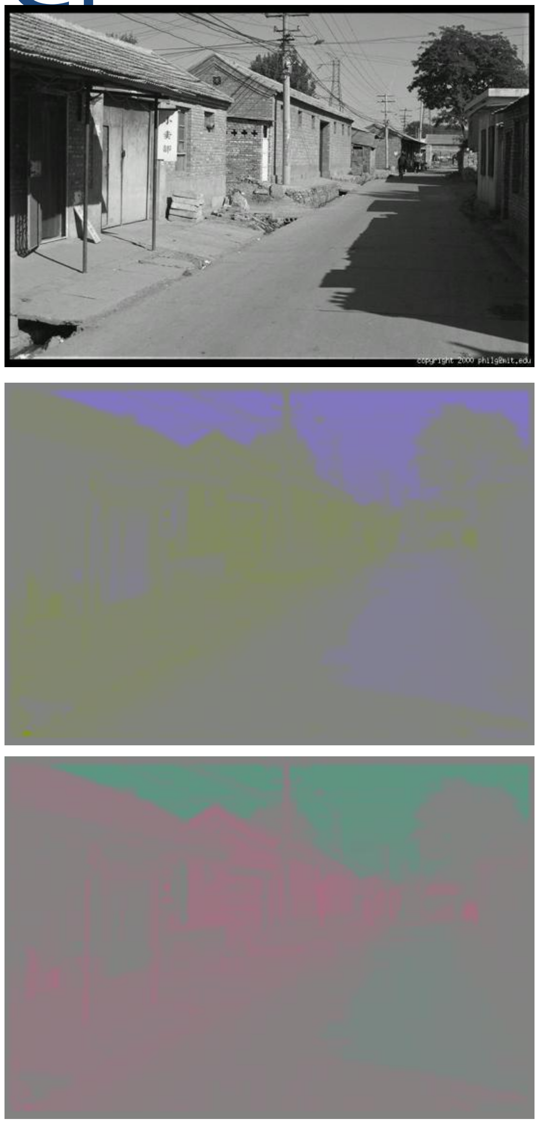
Cons

1. Not as good as other better smart color spaces

Y = 0



Y = 0.5



Y
(Cb=0.5, Cr=0.5)

Cb
(Y=0.5, Cr=0.5)

Cr
(Y=0.5, Cb=05)

Slide Credit: J. Hays, YUV cube: https://en.wikipedia.org/wiki/YUV

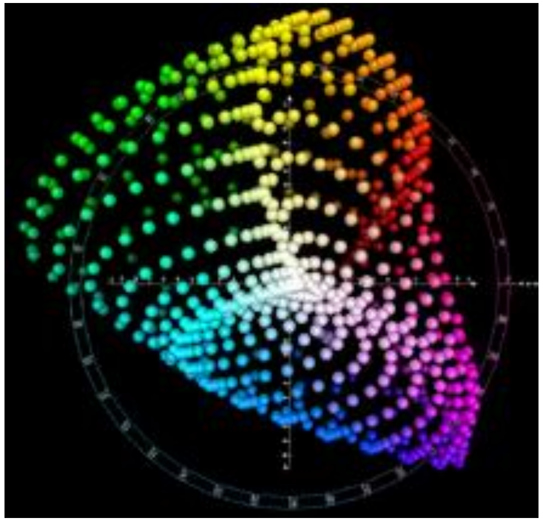
Another Option: LAB

Pros

1. Distances correspond with human judgment
2. Safe

Cons

1. Complex to calculate (don't write it yourself, lots of fp calculations)



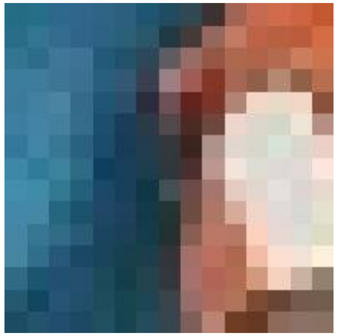
L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Slide Credit: J. Hays, Lab diagram cube: https://en.wikipedia.org/wiki/CIELAB_color_space

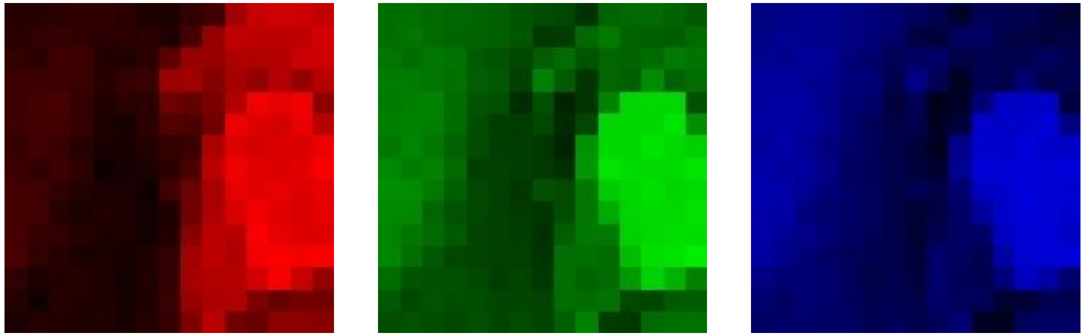
What is an image?



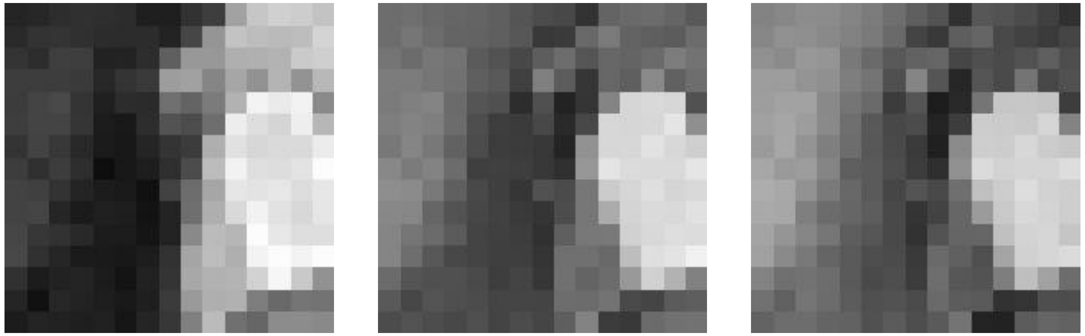
color image patch

How many bits are the intensity values?

red green blue



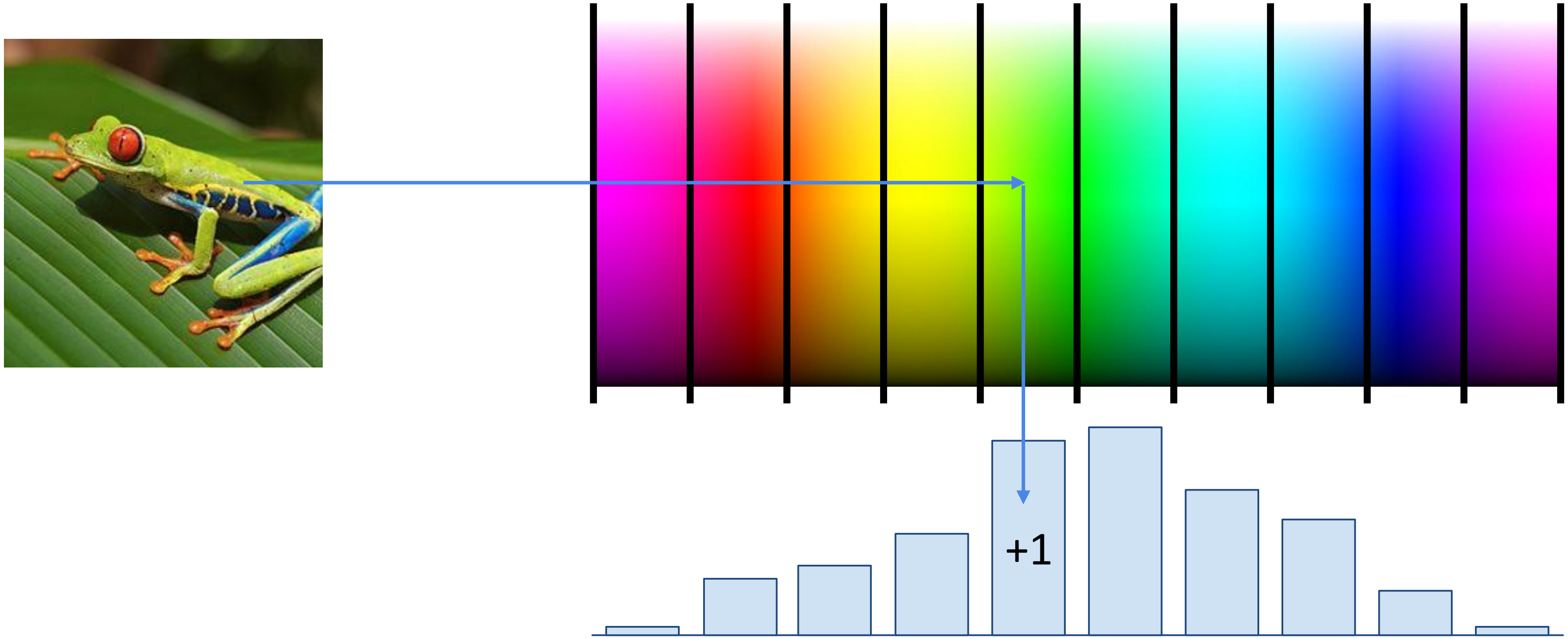
colorized for visualization



actual intensity values per channel

Each channel is a 2D array of numbers.

Image Features: Color Histogram



Frog image is in the public domain

Examples of point processing

original

$$x$$

darken

$$x - 128$$

lower contrast

$$\frac{x}{2}$$

non-linear lower contrast

$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert

$$255 - x$$

lighten

$$x + 128$$

raise contrast

$$x \times 2$$

non-linear raise contrast

$$\left(\frac{x}{255}\right)^{2} \times 255$$

Practice with Linear Filters

Original

0	0	0
0	1	0
0	0	0

?

Slide Credit: D. Lowe

Practice with Linear Filters

Original

0	0	0
0	1	0
0	0	0

The Same!

Slide Credit: D. Lowe

Practice with Linear Filters

Original

0	0	0
0	0	1
0	0	0

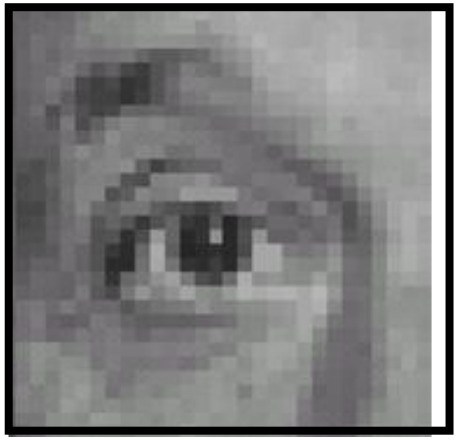
?

Slide Credit: D. Lowe

Practice with Linear Filters

Original

0	0	0
0	0	1
0	0	0



Shifted

<u>LEFT</u>

1 pixel

Slide Credit: D. Lowe

Practice with Linear Filters

Original

0	1	0
0	0	0
0	0	0

?

Slide Credit: D. Lowe

Practice with Linear Filters

Original

0	1	0
0	0	0
0	0	0

Shifted
DOWN
1 pixel

Slide Credit: D. Lowe

Practice with Linear Filters

Original

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

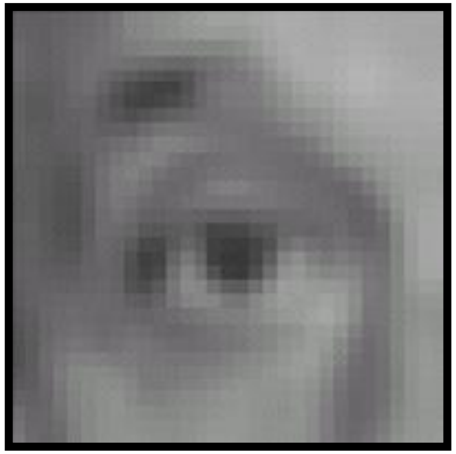
?

Slide Credit: D. Lowe

Practice with Linear Filters

Original

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



Blur
(Box Filter)

Slide Credit: D. Lowe

1D Case

Signal	10	12	9	11	10	11	12
Filter	1/3	1/3	1/3				
Output		10.33					

1D Case

Signal	10	12	9	11	10	11	12
Filter		1/3	1/3	1/3			
Output		10.33	10.66				

1D Case

Signal	10	12	9	11	10	11	12
Filter			1/3	1/3	1/3		
Output		10.33	10.66	10			

1D Case

Signal	10	12	9	11	10	11	12
Filter				1/3	1/3	1/3	
Output		10.33	10.66	10	10.66		

1D Case

Signal	10	12	9	11	10	11	12
Filter					1/3	1/3	1/3
Output		10.33	10.66	10	10.66	11	

Applying a 2D Filter

Input

I11	I12	I13	I14	I15	I16
I21	I22	I23	I24	I25	I26
I31	I32	I33	I34	I35	I36
I41	I42	I43	I44	I45	I46
I51	I52	I53	I54	I55	I56

Filter

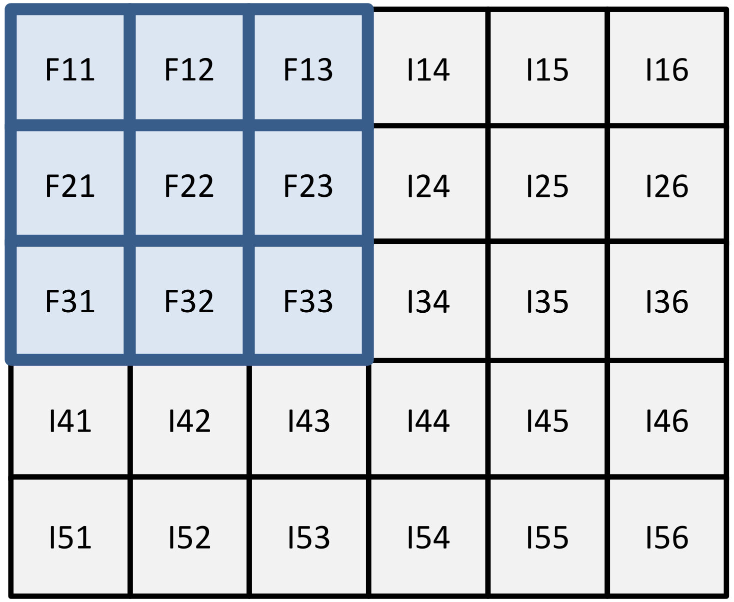
F11	F12	F13
F21	F22	F23
F31	F32	F33

Output

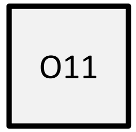
O11	O12	O13	O14
O21	O22	O23	O24
O31	O32	O33	O34

Applying a 2D Filter

Input & Filter



Output



O11 = I11*F11 + I12*F12 + … + I33*F33

Applying a 2D Filter

Input & Filter

I11	F11	F12	F13	I15	I16
I21	F21	F22	F23	I25	I26
I31	F31	F32	F33	I35	I36
I41	I42	I43	I44	I45	I46
I51	I52	I53	I54	I55	I56

Output

O11	O12

$$O12 = I12*F11 + I13*F12 + \ldots + I34*F33$$

Applying a 2D Filter

Input

I11	I12	I13	I14	I15	I16
I21	I22	I23	I24	I25	I26
I31	I32	I33	I34	I35	I36
I41	I42	I43	I44	I45	I46
I51	I52	I53	I54	I55	I56

Filter

F11	F12	F13
F21	F22	F23
F31	F32	F33

Output

How many times can we apply a 3x3 filter to a 5x6 image?

Applying a 2D Filter

Input

I11	I12	I13	I14	I15	I16
I21	I22	I23	I24	I25	I26
I31	I32	I33	I34	I35	I36
I41	I42	I43	I44	I45	I46
I51	I52	I53	I54	I55	I56

Filter

F11	F12	F13
F21	F22	F23
F31	F32	F33

Output

O11	O12	O13	O14
O21	O22	O23	O24
O31	O32	O33	O34

$$Oij = Iij*F11 + Ii(j+1)*F12 + \ldots + I(i+2)(j+2)*F33$$

Image filtering

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10							

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	$\boxed{20}$						

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30					

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$g[\cdot,\cdot]$ $\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

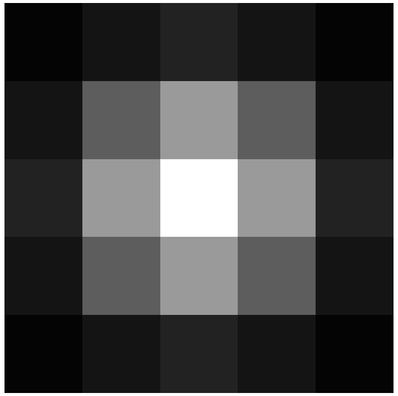
Credit: S. Seitz

Recognize the Filter?

It's a Gaussian!

$$Filter_{ij} \propto \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\begin{bmatrix} 0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\ 0.013 & 0.060 & 0.098 & 0.060 & 0.013 \\ 0.022 & 0.098 & 0.162 & 0.098 & 0.022 \\ 0.013 & 0.060 & 0.098 & 0.060 & 0.013 \\ 0.003 & 0.013 & 0.022 & 0.013 & 0.003 \end{bmatrix}$$



Sharpening Filter

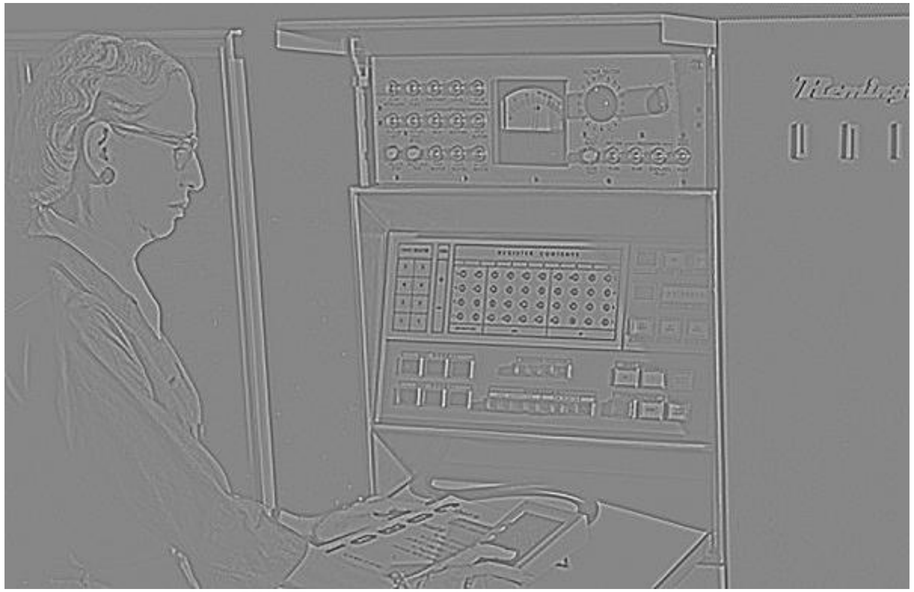
Image

-

Smoothed

Details

=

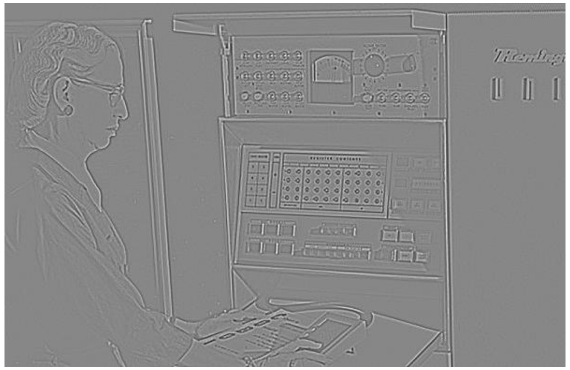


Sharpening Filter

Image

+α

Details



“Sharpened” α=0

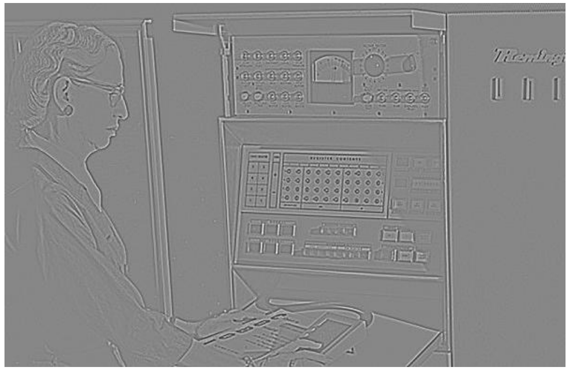
=

Sharpening Filter

Image

+α

Details



"Sharpened" α=1

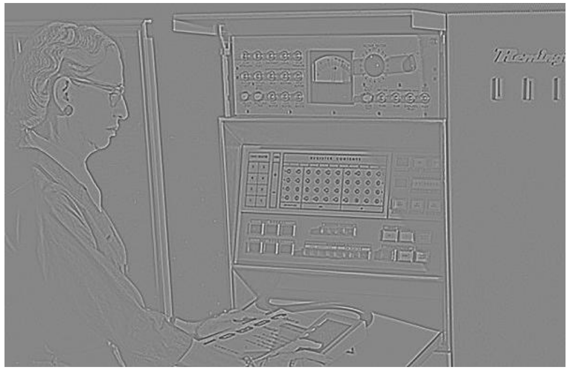
=

Sharpening Filter

Image

+α

Details



"Sharpened" α=2

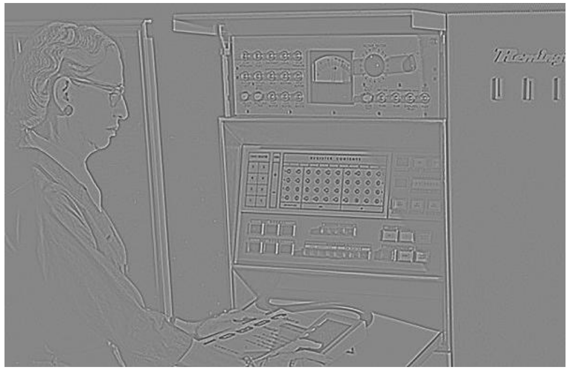
=

Sharpening Filter

Image

+α

Details



“Sharpened” α=10

=