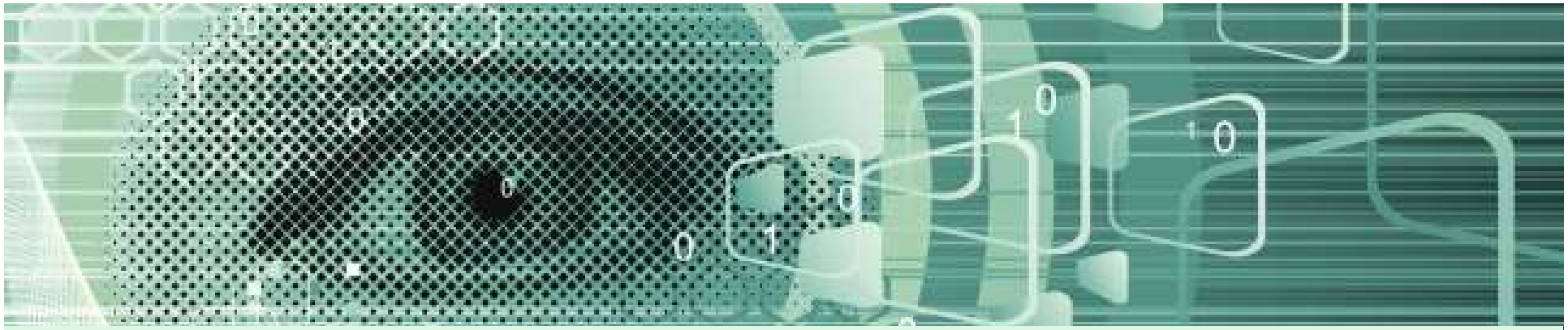


Lecture 02- Basics of Image processing

EE382-Visual localization & Perception

Danping Zou @Shanghai Jiao Tong University

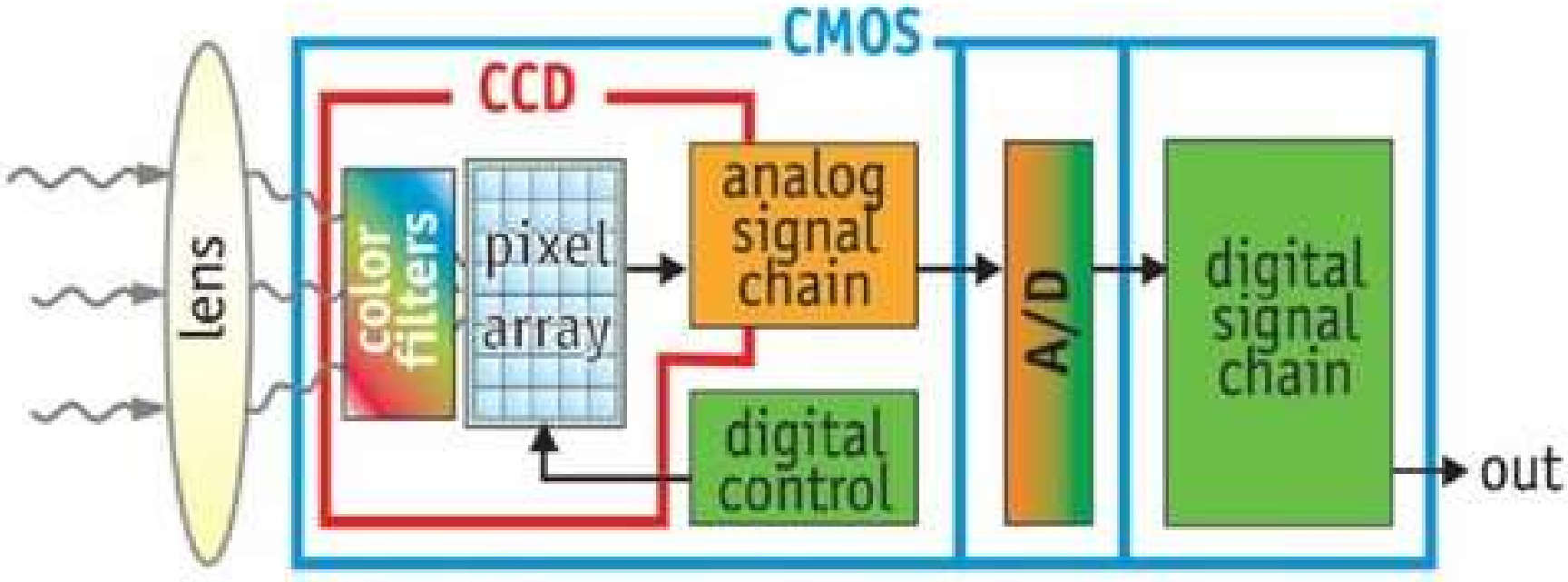
What we are about to learn.

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

- CCD & CMOS
- Global shutter V.S rolling shutter
- Performance of Image Sensor
- Human color system & Bayer pattern
- Image as a 2D function (continuous) /2D matrix (discrete)
- Linear filtering
- Image derivatives
- Histogram equalization/specialization

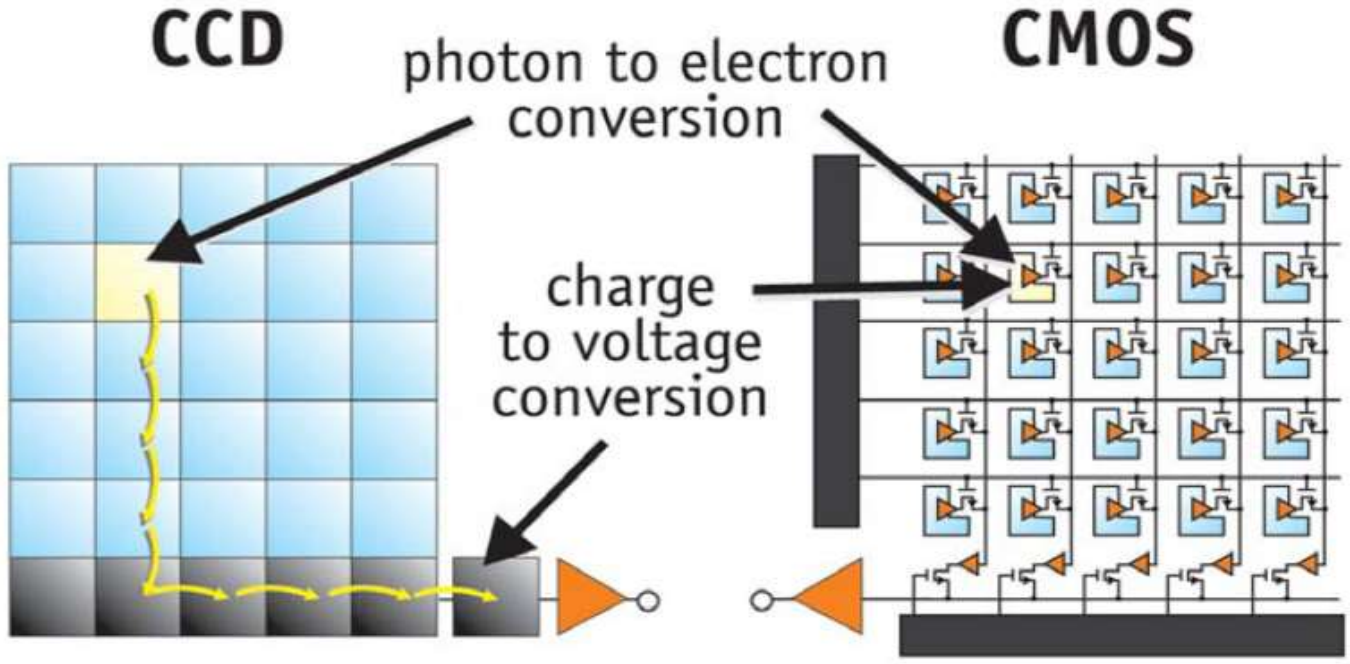
What is an digital image?

- A digital image is the data read from the **image sensor**



CCD vs CMOS

- CCD (**charge–coupled device**)
- CMOS(**complementary metal-oxide semiconductor**)



CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.

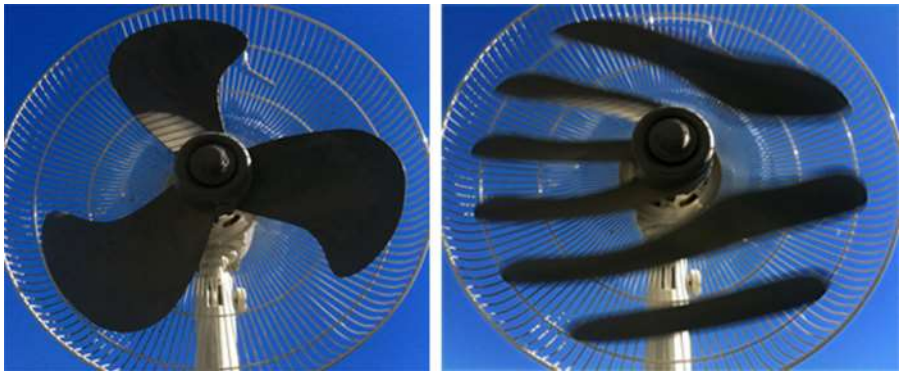
CCD vs CMOS

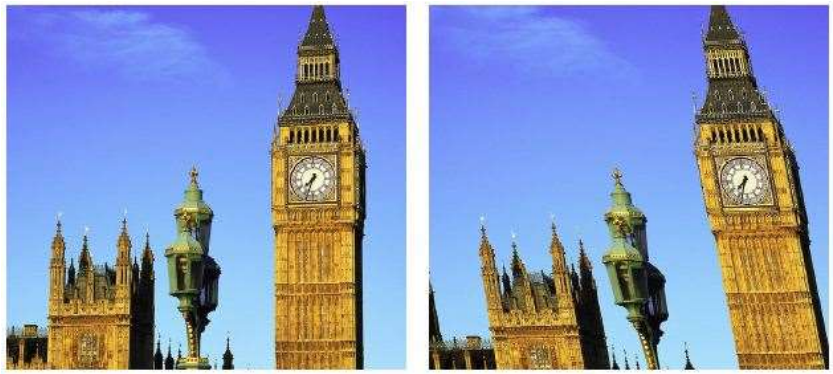
Early ages:

• **CCD**	• **CMOS**
High quality	Lower quality (skew, wobble)
Less noise	High noise
More expensive	Cheaper
More power consumption	Less power consumption
Global Shutter	*Rolling Shutter*

Note that, CMOS becomes the most popular image sensor in the market now.

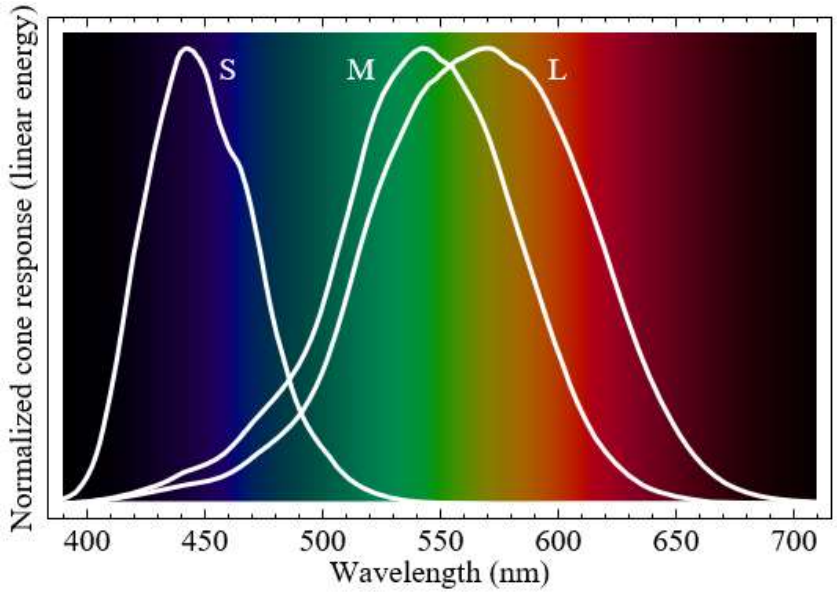
Global Shutter v.s Rolling Shutter





Color image

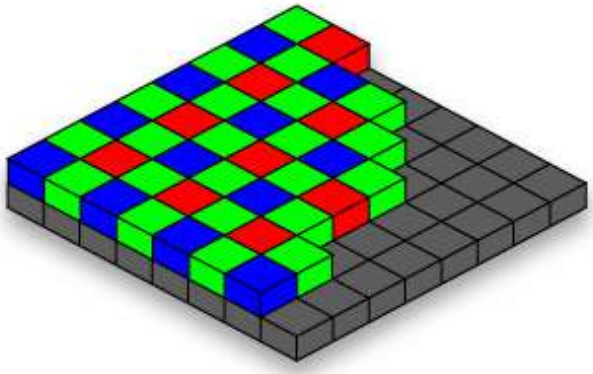
- Color vision of human（daylight)

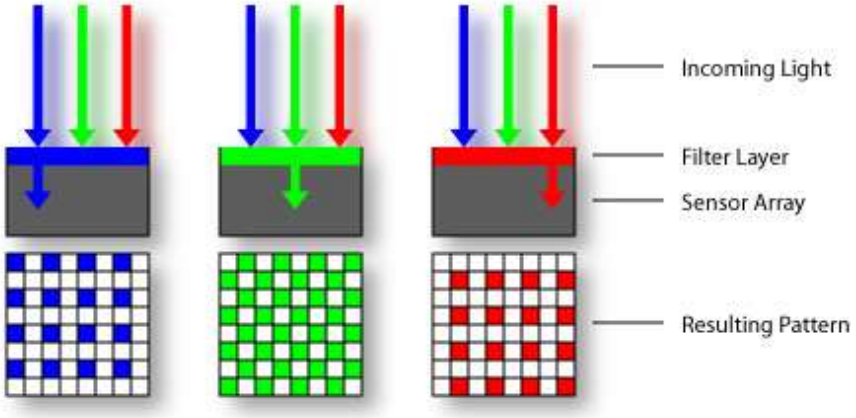


Cone type	Name	Range	Peak wavelength
S (Short)	β	400–500 nm	420–440 nm
M (Medium)	γ	450–630 nm	534–555 nm
L (Long)	ρ	500–700 nm	564–580 nm

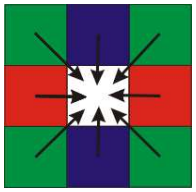
Color image

- Bayer pattern

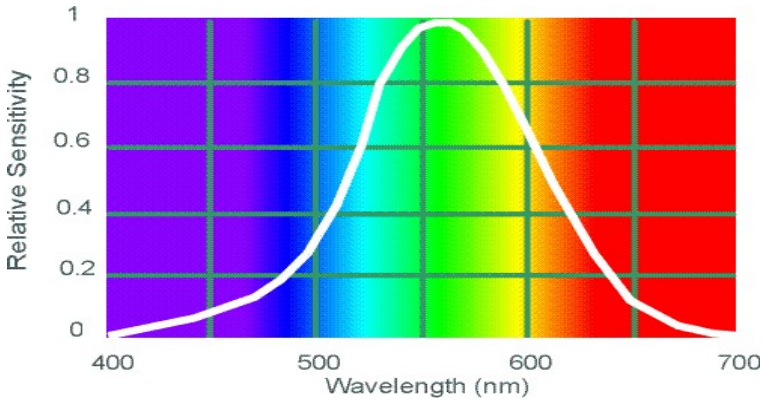




Demosaicing : interpolating the missing color from neighboring pixels



Why are more green filters?



Human Luminance Sensitivity Function

Summary

- An image is the data from the image sensor (CMOS or CCD)
- Today, CMOS becomes the dominant image sensor.
- Rolling shutter could cause deformation such as skew, wobble.
- Three cone cells receives red, green and blue lights in human vision system.
- Bayer mask is used for the image sensor to receive three colors simultaneously.

Imaging performance

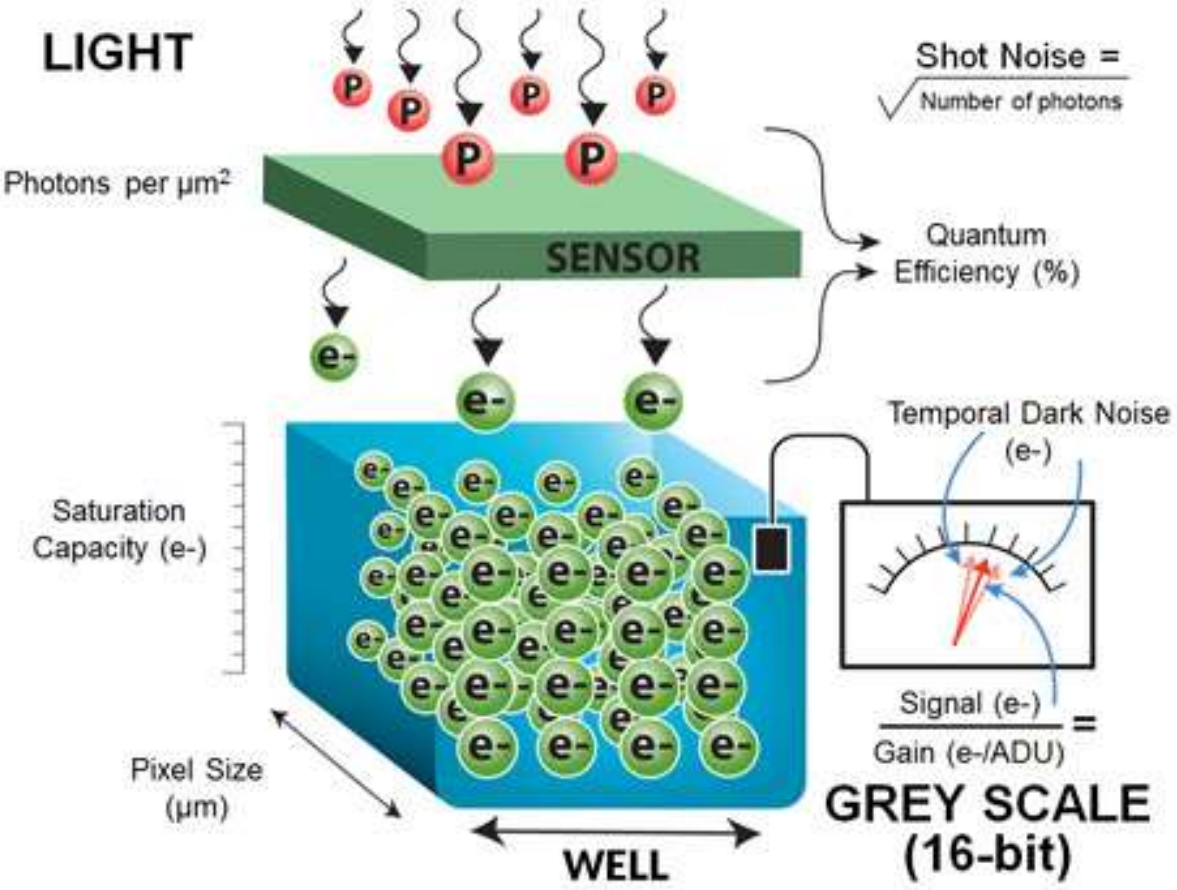
- How to measure the performance of an image sensor?
  - Frame rate ? Resolution ?
- For computer vision application
  - Image quality
  - Low-light condition

Imaging performance

- **EMVA 1288 standard**
  - **EMAV (E**uropean **M**achine **V**ision **A**ssociation)
  - **EMVA 1288** is the Standard for Measurement and Presentation of Specifications for Machine Vision Sensors and Cameras.
- Goal
  - Define measures to characterize cameras
  - Make data sheets from different vendors comparable
  - Allow the customer to select a suitable camera for his task
  - Make it difficult to “beautify” data sheets

Imaging performance

- Pixel Size
- Quantum efficiency
- Saturation Capacity
- Temporal Dark Noise



Imaging performance

- Quantum efficiency
  - Percentage of photons converted to electrons at a particular wavelength
- Saturation capacity (Well depth)
  - Amount of charge that a pixel can hold
- Temporal dark noise (Read noise)
  - Noise in the sensor when there is no signal

Imaging Performance

- Sensor Sensitivity – Signal-to-Noise

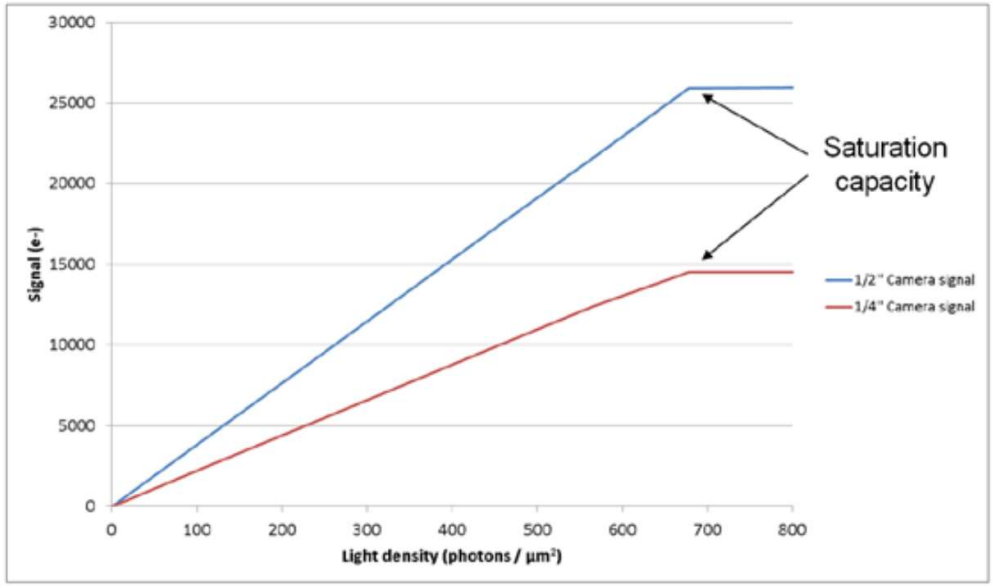
$$Signal = Light\ density \times (Pixel\ Size)^2 \times Quantum\ Efficiency$$

$$Noise = \sqrt{(Temporal\ Dark\ Noise)^2 + (Shot\ Noise)^2}$$

Imaging Performance

- Comparison of two sensors

Camera	Sensor	Pixel size (μm)	Quantum Efficiency (%)	Temporal Dark Noise (e-)	Saturation Capacity (e-)
1/4" Camera (FL3-GE-03S1M-C)	ICX618	5.6	70	11.73	14,508
1/2" Camera (BFLY-PGE-03S3M-C)	ICX414	9.9	39	19.43	25,949



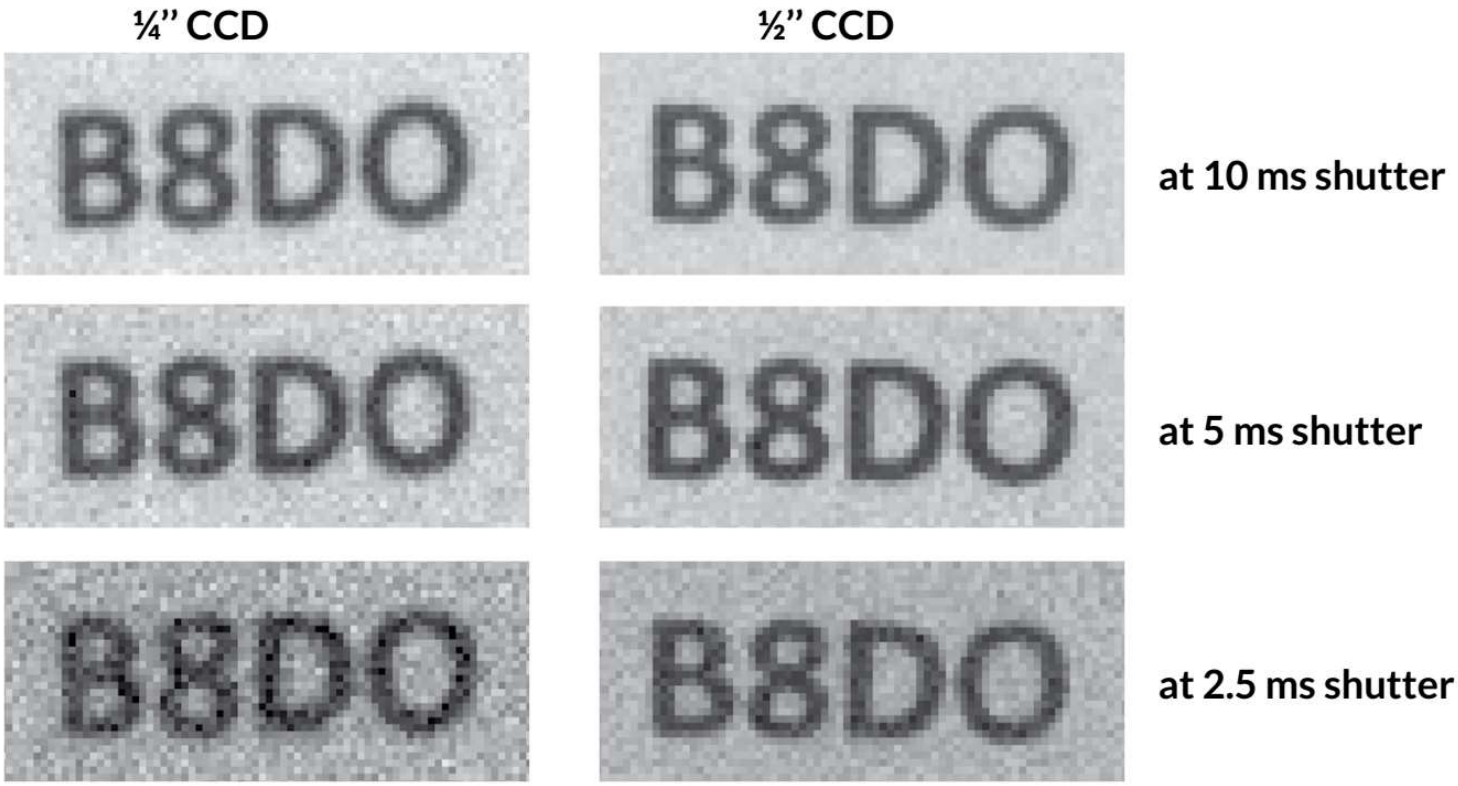
$$\frac{Signal}{Light\ Density} = (PixelSize)^2 \times Quantum\ Efficiency$$

$$\text{ICX618} \quad 2.2 \times 10^{-13}$$

$$\text{ICX414} \quad 3.82 \times 10^{-13}$$

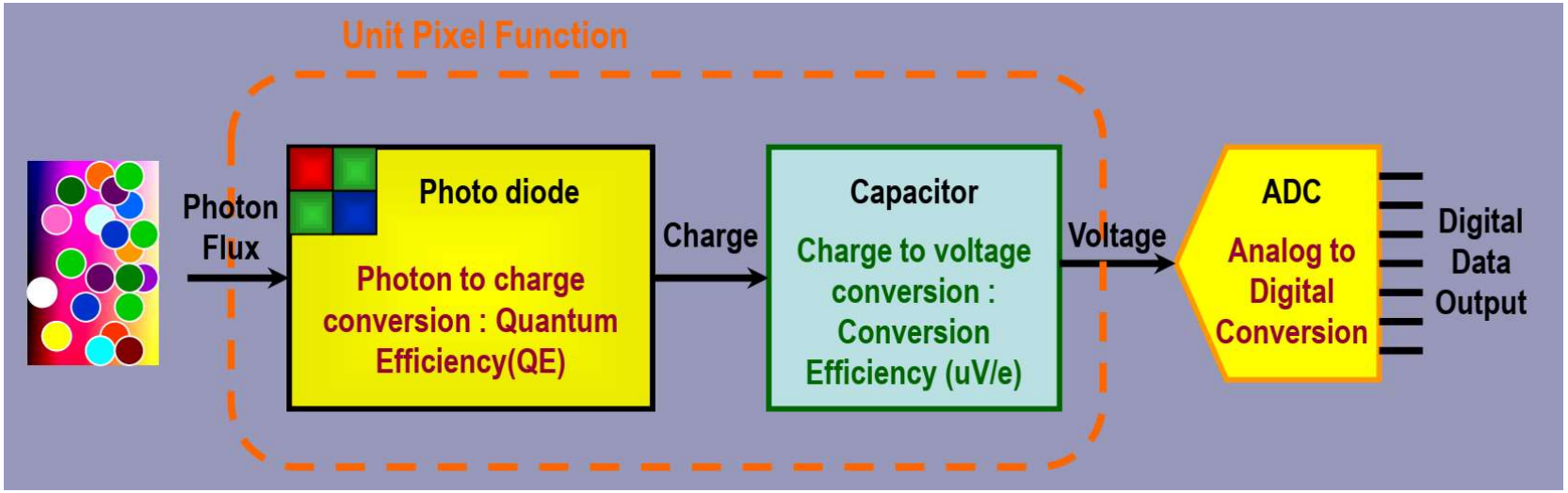
Imaging Performance

- Comparison of two sensors



Imaging Performance

- Sensor sensitivity measured by $V/(lux \cdot sec)$



$$Quantum\,Efficiency\,(Q.E) : e/ph = e/(lux \cdot m^2 \cdot sec)$$

$$Conversion\,Efficiency\,(C.E) : V/e$$

$$Sensitivity = (Q.E \times C.E) \times photodiode\,area : V/(lux \cdot sec)$$

Imaging Performance

- Some sensors of high sensitivity in low-light level

	Array Size	Pixel Size	Resolution	Sensitivity
AR0130CS	1/3’’	3.75µm	VGA@60fps	6.5V/lux-sec
OV9281	1/4’’	3µm	1200x800@120fps	1.9V/lux-sec

Summary

- For machine vision applications, sensitivity is the key consideration
- EMVA 1288 standard enable measuring the performance of different image sensors
- Sensitivity is mainly measured by Signal-to-Noise ratio
- Pixel size is an important indicator
- **V/(lux-sec)** is also used to measure the sensitivity.

Image as a function

- An image can be mathematically thought of as a function, $I : \mathbb{R}^2 \rightarrow \mathbb{R}$
  - $I(x, y)$ represents the intensity at the position (x, y)
  - $(x, y) \in \Omega$, where $\Omega : [1, w] \times [1, h]$
  - $I \in [0, 255]$

- A color image is thought as a 'vector-valued' function :

$$I(x, y) = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}$$

Image as an array

- A gray image can be thought as a matrix of integer values

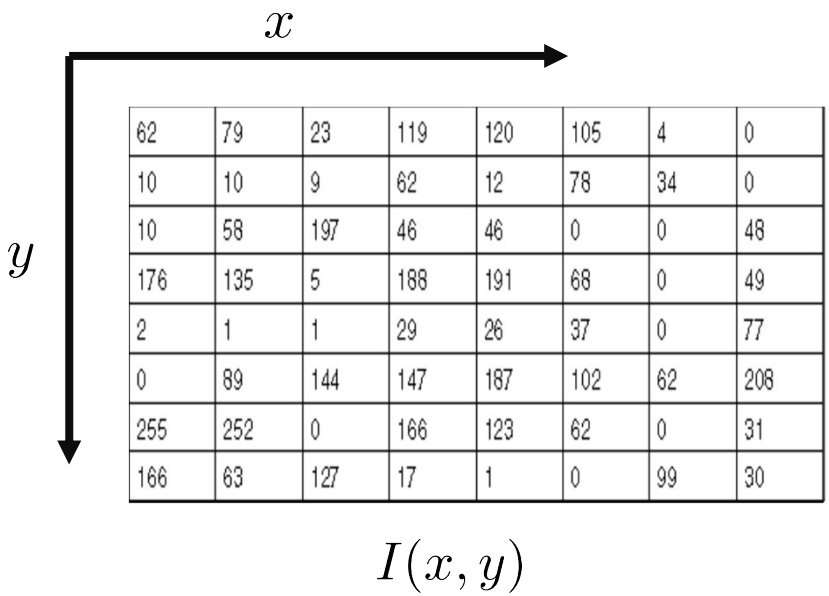


Image filtering

- Correlation

$$g = I \otimes w$$

$$g(x,y) = \sum_s \sum_t I(x+s, y+t) w(s,t)$$

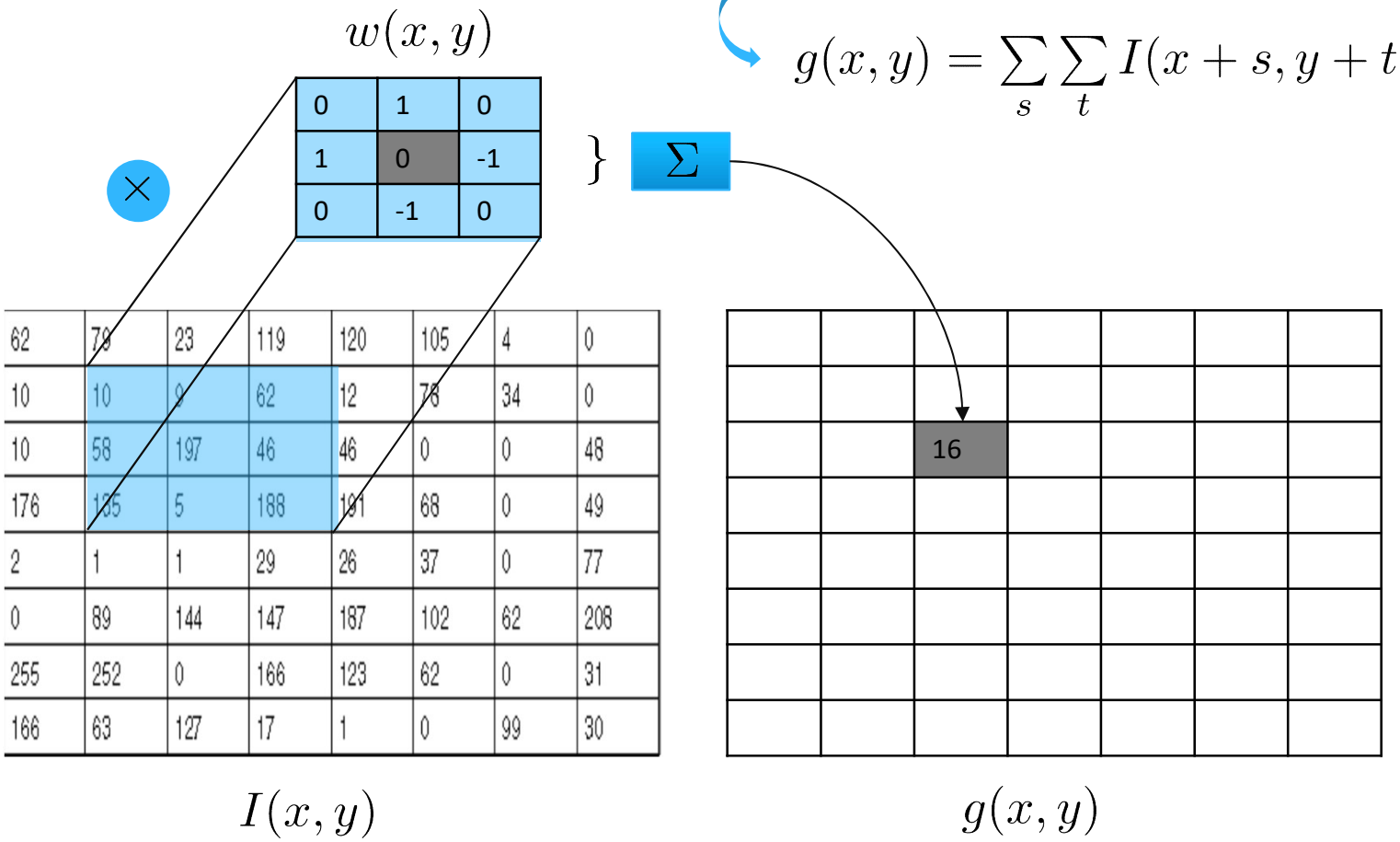
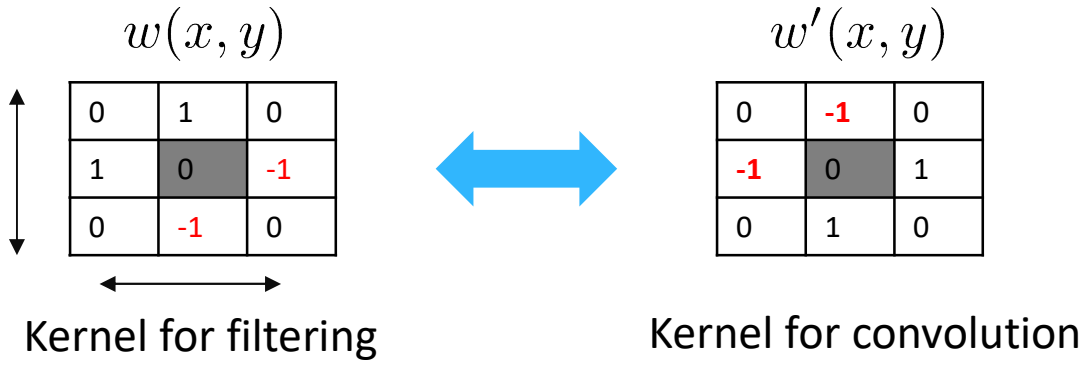


Image filtering

- Convolution is the same as correlation except the kernel is flipped both horizontally and vertically

$$g(x,y) = \sum_s \sum_t I(x-s, y-t) w(s,t)$$



Box filter

- Kernel/mask – a small matrix determining the filter results

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Box blur

Gaussian filter

$$w(x,y) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2+y^2}{2\sigma^2}$$

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussian blur (3x3)

$$\sigma \approx 0.85$$

Gaussian filter

$$w(x,y) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2+y^2}{2\sigma^2}$$

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

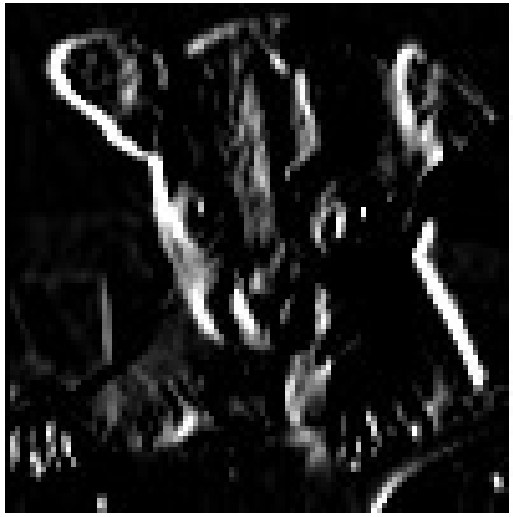
Gaussian blur (5x5)

$\sigma \approx 1.05$

Gradient operator

- Sobel operator (x – direction)

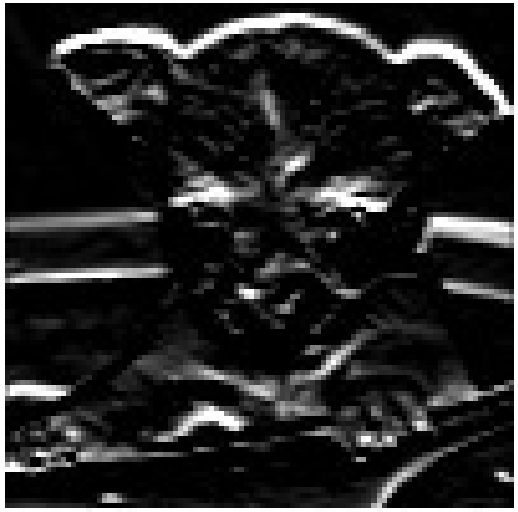
$$w(x,y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Gradient operator

- Sobel operator (y-direction)

$$w(x,y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



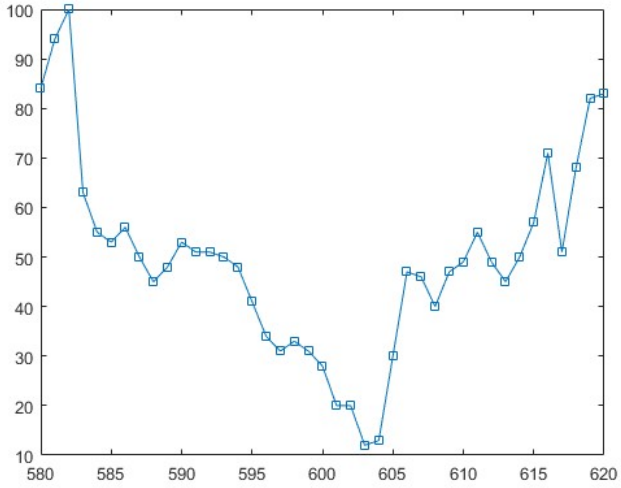
Summary

- An image is described by a 2D function (mathematic) or a matrix/array (algorithm).
- Image filtering is described by the correlation operator
- Convolution is equal to correlation except that the kernel is flipped in both directions.
- Box and Gaussian filter cause image blur
- Gradient operators can be used to detect edges.

Image derivative

Let’s consider one-dimensional example:

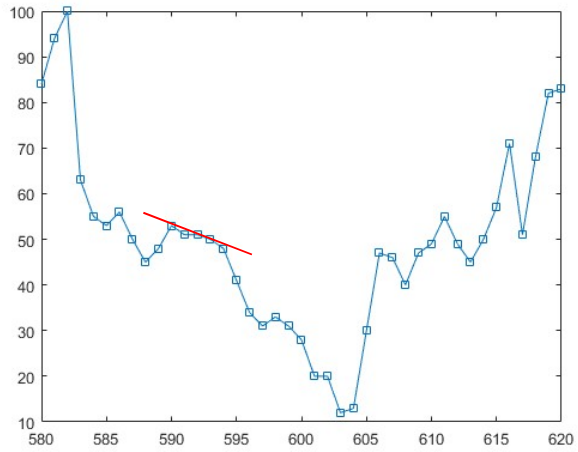
$$I(x) = I(x, y)|y = y_0$$



First derivative

- Sometimes we want to know about value of $I(x+dx), dx < 1$. It can be approximated by Taylor series expansion:

$$I(x+dx) \approx I(x) + \frac{\partial I(x)}{\partial x} dx + o(dx^2)$$



$$I_x = \frac{\partial I(x)}{\partial x}$$

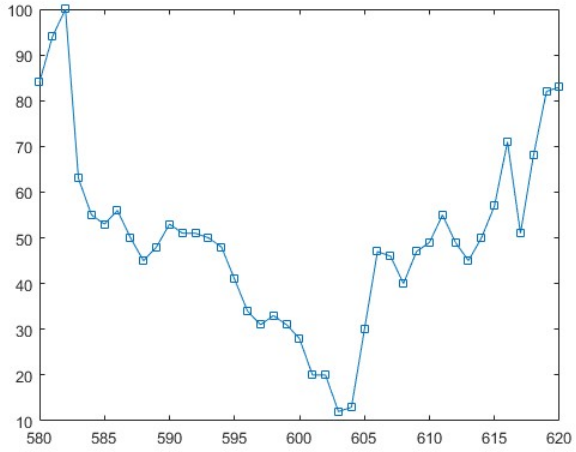
First derivative

- Compute derivatives (1D)

$$I_x = \lim_{dx \to 0} \frac{I(x+dx) - I(x)}{dx} \approx I(x+1) - I(x) \quad \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$$

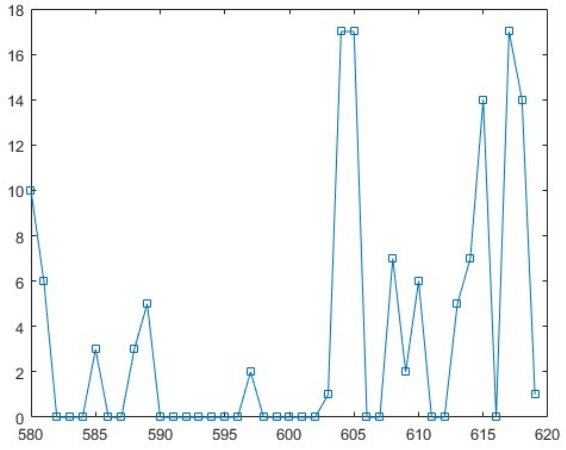
$$\approx I(x) - I(x-1) \quad \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$$

$$\approx I(x+1) - I(x-1) \quad \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$



$I(x)$

$\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$



$I_x(x)$

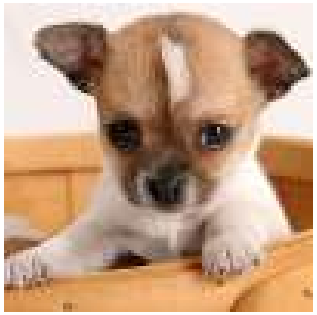
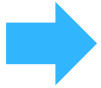
First derivative

- 2D cases, the partial derivatives are computed as

$$\frac{\partial I(x,y)}{\partial x} = I_x \approx I \otimes D_x \qquad \frac{\partial I(x,y)}{\partial y} = I_y \approx I \otimes D_y$$

Sobel operators

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad D_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

First derivative

- The image can be locally approximated by 2D Taylor series expansion:

$$I(x+dx, y+dy) \approx I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

$$= I(x,y) + \nabla I \begin{bmatrix} dx \\ dy \end{bmatrix}$$

In the next talk, I'll show how this approximation is used to derive the Harris corner detector.

Second derivative

- The second derivative is computed as

$$I_{xx} = \lim_{dx \to 0} \frac{I_x(x+dx,y) - I_x(x,y)}{dx}$$

$$\approx I_x(x+1,y) - I_x(x,y)$$

- Let $I_x(x,y) \approx I(x,y) - I(x-1,y)$, we have

$$I_{xx}(x,y) = I(x+1,y) - I(x,y) - (I(x,y) - I(x-1,y))$$

$$= I(x-1,y) - 2I(x,y) + I(x+1,y)$$

$I_{xx} = I \otimes D_{xx}$, where $D_{xx} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

Second derivative

- The second derivative is computed as

$$I_{xy} = \lim_{dy \to 0} \frac{I_x(x, y+dy) - I_x(x,y)}{dy}$$

$$\approx I_x(x, y+1) - I_x(x, y-1)$$

- Let $I_x(x,y) \approx I(x+1,y) - I(x-1,y)$, we have

$$I_{xy}(x,y) = I(x+1,y+1) - I(x-1,y+1)$$
$$-(I(x+1,y-1) - I(x-1,y-1))$$

$I_{xy} = I \otimes D_{xy}$, where $D_{xy} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$

Second derivatives

$$D_{xx} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

I_{xx}

$$D_{xy} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

I_{xy}

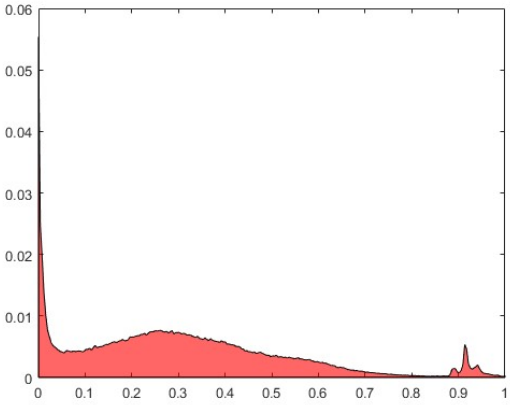
Summary

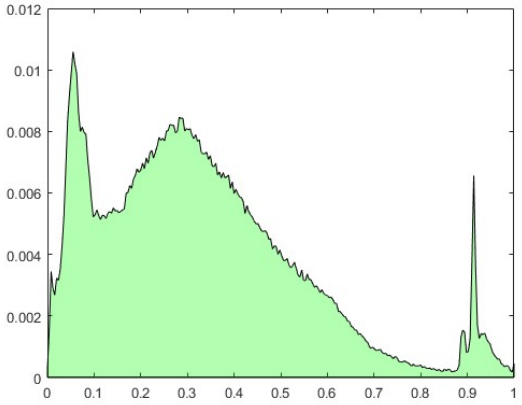
- Image can be locally approximated by Taylor series expansion.
- First derivative of an image can be computed by finite difference in x and y directions
- Second derivative can be also computed by finite difference.
- Finite difference is implemented by filtered by a gradient kernel.

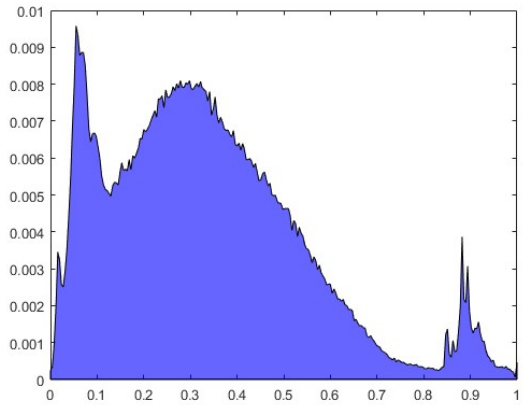
Histogram

- The histogram of an image is a distribution of number of pixels at each gray level (for 8-bit pixel, there are 255 gray levels).

$$h(g), \text{where } g \in [0,1], \int_g h(g)dg = 1$$



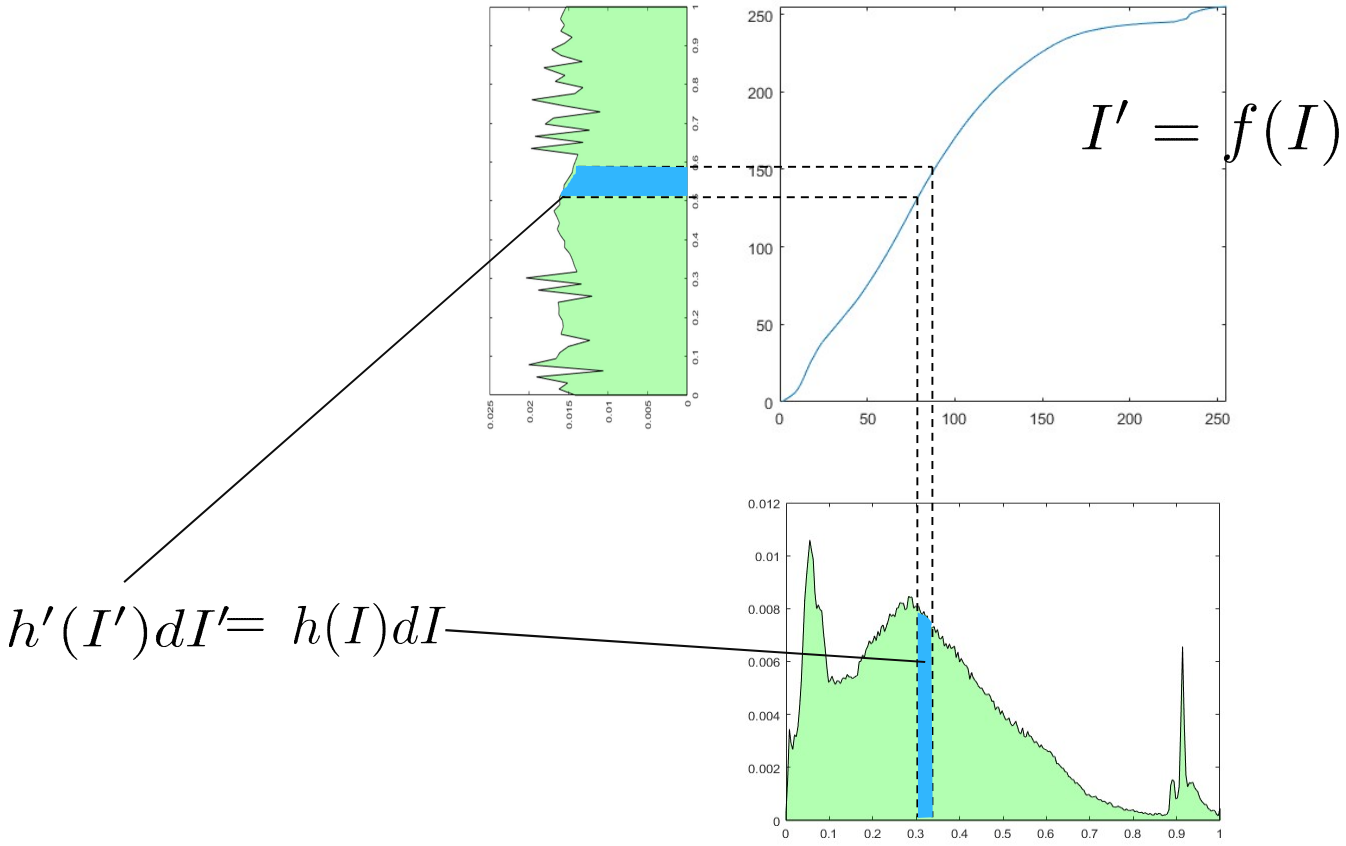




Histogram equalization

- Sometimes we want to find a brightness mapping function that maps $f(I)$ each gray level to a new gray level to change the histogram of the image:

$$h(I) \rightarrow h'(I')$$



Histogram equalization

- Suppose the new histogram is a constant $h'(I') = const.$

$$h'(I')dI' = h(I)dI$$

$$\Rightarrow dI' = h(I)dI$$

$$\Rightarrow f'(I)dI = h(I)dI \qquad (I' = f(I))$$

$$\Rightarrow f'(I) = h(I)$$

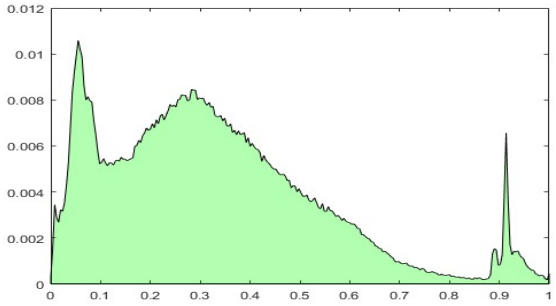
$$\Rightarrow f(I) = \int_{t=0}^{t=I} h(t)dt$$

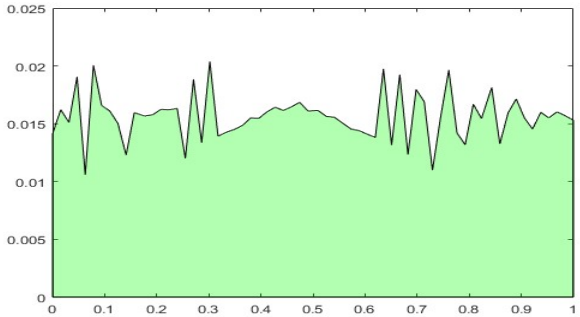
- $f(I)$ is a **c**umulative **d**istribution **f**unction (CDF) of $h(t)$

Histogram equalization

- Algorithm:
  - 1.Compute the histogram of an input image $h(I)$
  - 2.Calculate the cumulative distribution function of $h(I)$, namely $f(I)$
  - 3. Map the intensity of each pixel by

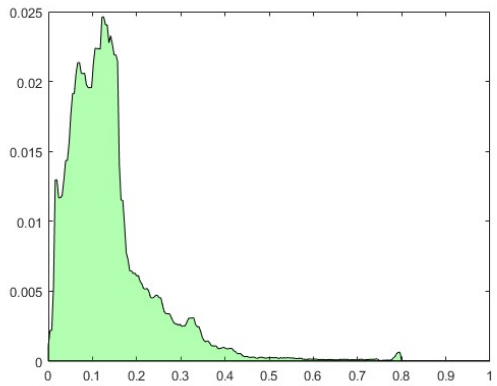
$$I'(x,y) = f(I(x,y))$$





Histogram specialization

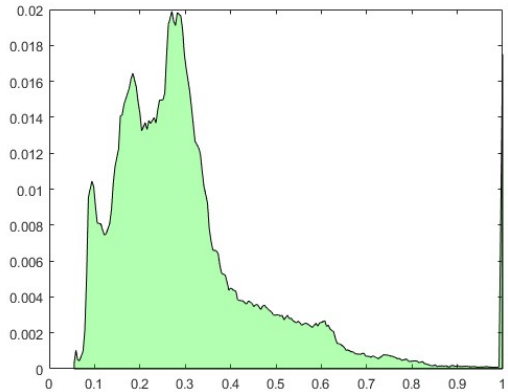
- Case 2: The new histogram is an arbitrary histogram



$h(I)$

$I' = f(I)$

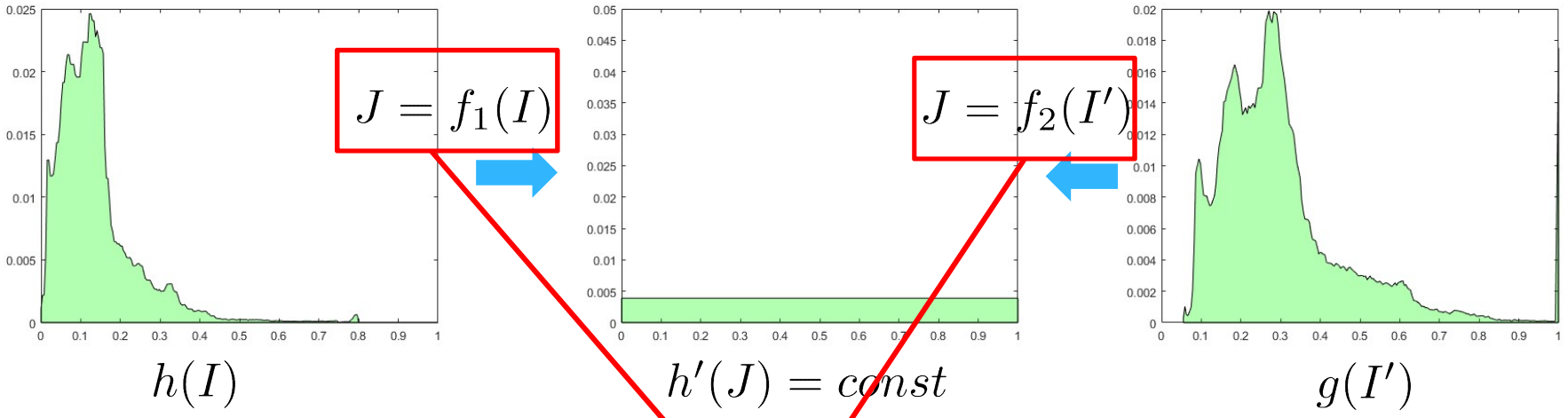
?



$g(I')$

Histogram specialization

- Consider two brightness mapping functions transform both histograms into a constant histogram:

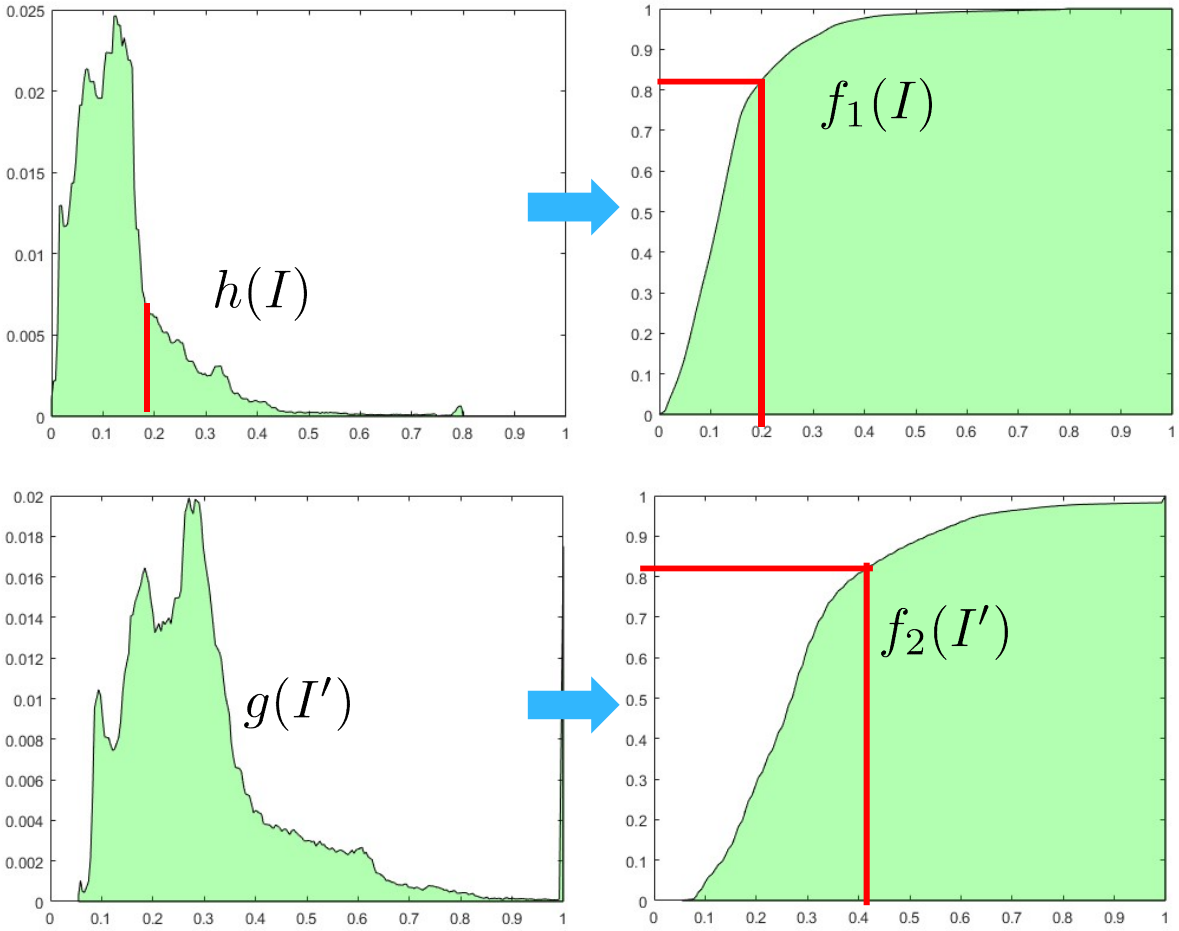


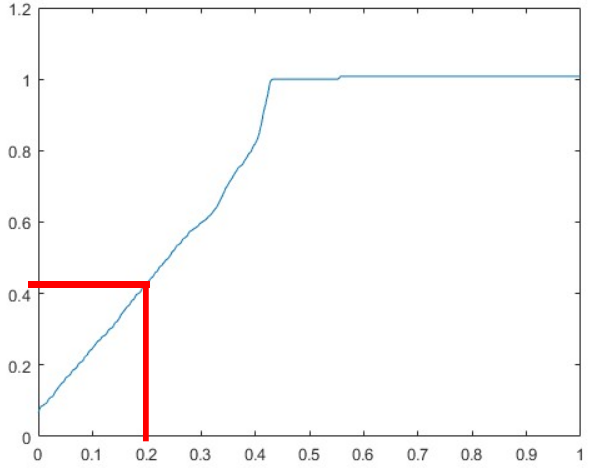
Histogram specialization

- Algorithm:
  - 1. Calculate the cumulative histogram of the first image : $f_1(I)$
  - 2. Calculate the cumulative histogram of the second image: $f_2(I')$
  - 3. build a lookup table $I' = f_2^{-1} \circ f_1(I)$ by finding the corresponding gray level I'_j of each gray level I_i , where $I'_j = \arg\min_j |f_1(I_i) - f_2(I'_j)|$
  - 4. map the new intensity of each pixel by finding the lookup table

Histogram specialization

- Illustration



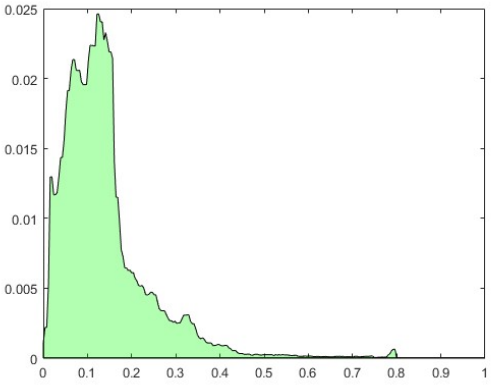


$$I' = f_2^{-1} \circ f_1(I)$$

$$0.4 \leftarrow 0.2$$

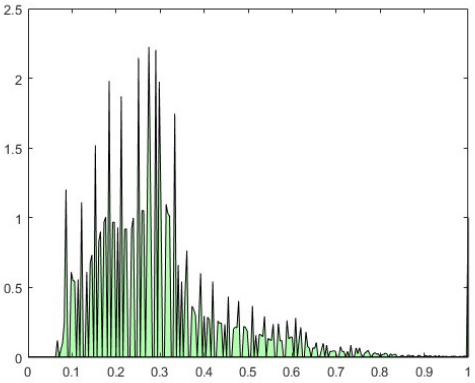
Histogram specialization

Original



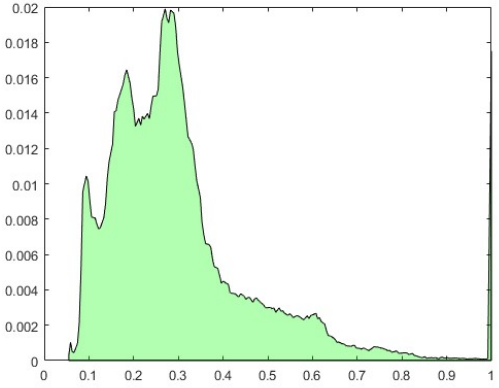
$h(I)$

Adjusted



$h'(I)$

Target



$g(I')$

Summary

- Histogram of an image is a distribution of pixels at different brightness levels.
- Histogram equalization distributes pixels equally at all gray levels and can improve the details of the image appearance.
- Histogram specialization adjust the brightness of the original image to make it be close to the target image.
- Histogram can address the illumination change to some extend and make feature matching easier.

Homework

- Write a small program to implement a histogram specialization algorithm described in P45 .
  - In any language (C++, java, python, Matlab)
  - Implement **each step** described in P45
  - Tests on at least 5 pairs of images with different exposure time.
  - Readme.txt about how to compile and run the program
  - A short report about the experimental results

- Deadline 30th,September
  - Email Title ：学号-姓名-作业1
  - Attachment : 学号-姓名-作业1.zip
  - vls_class@163.com