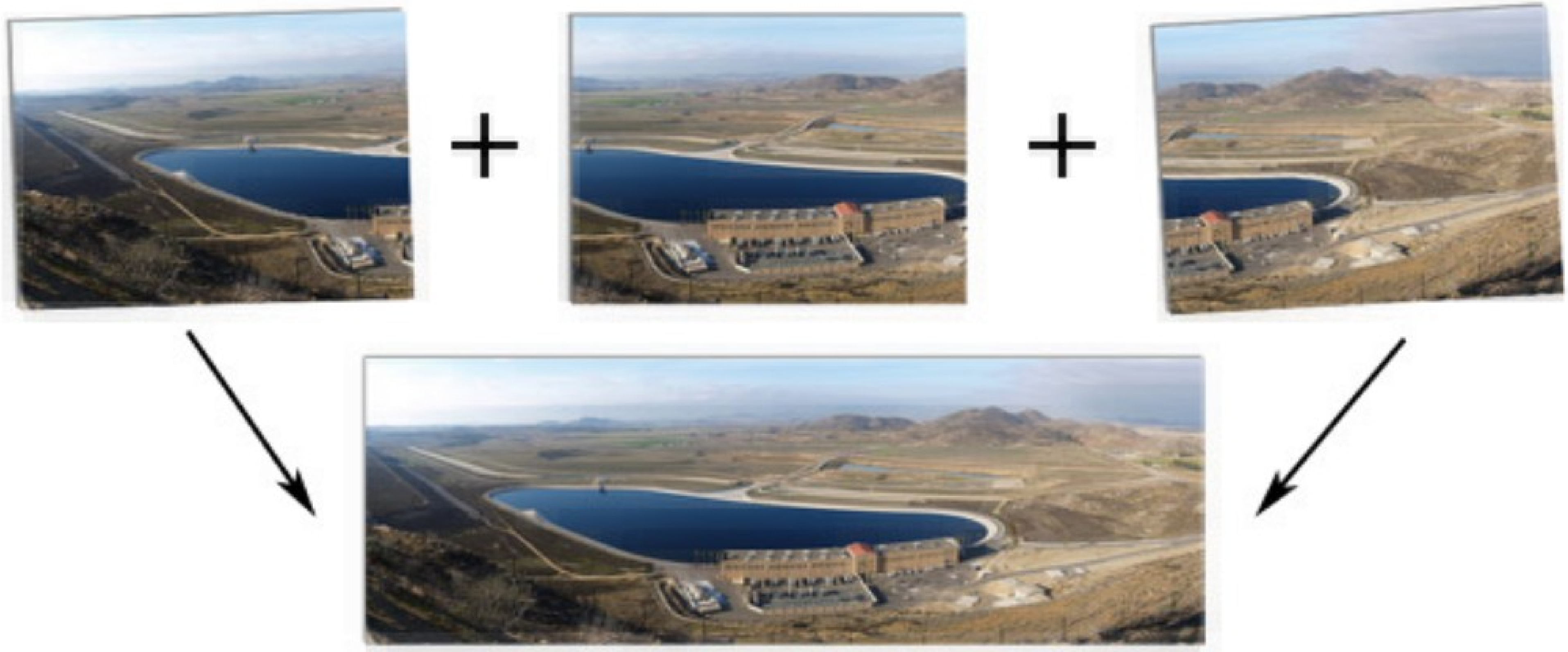
Image Processing
And
Computer Vision

Algorithms and Applications

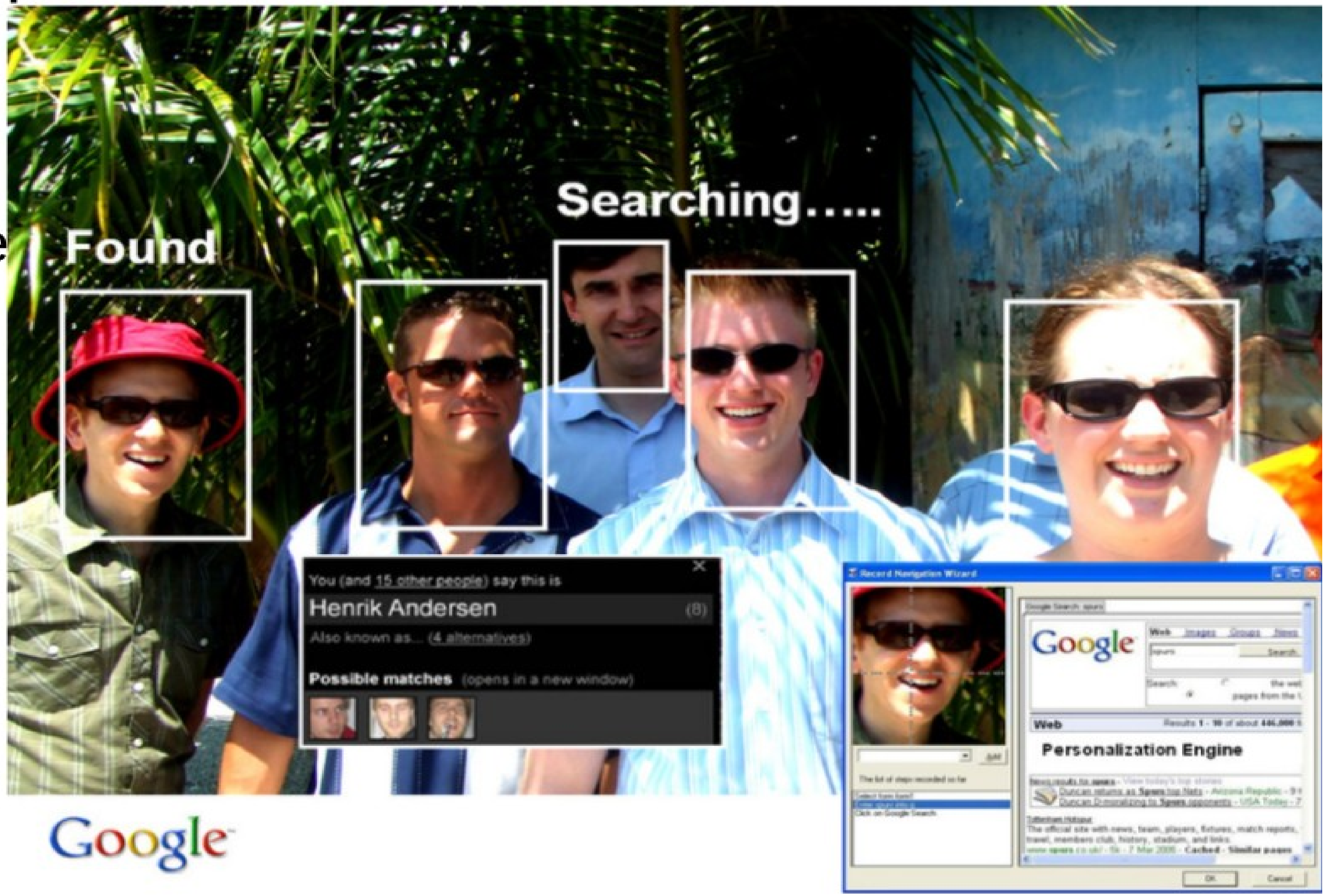
Motivation

- Editing pictures, movies



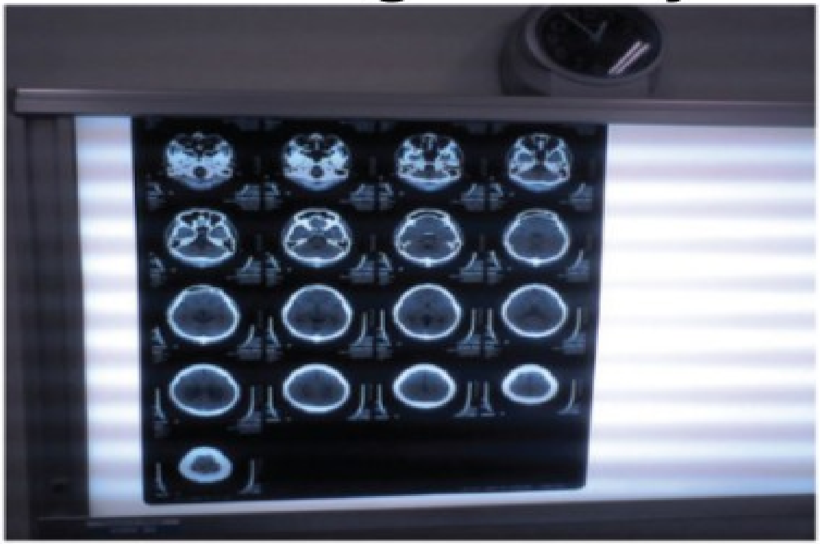
Motivation

- Internet applications
  - Tagging: face detect, recognize



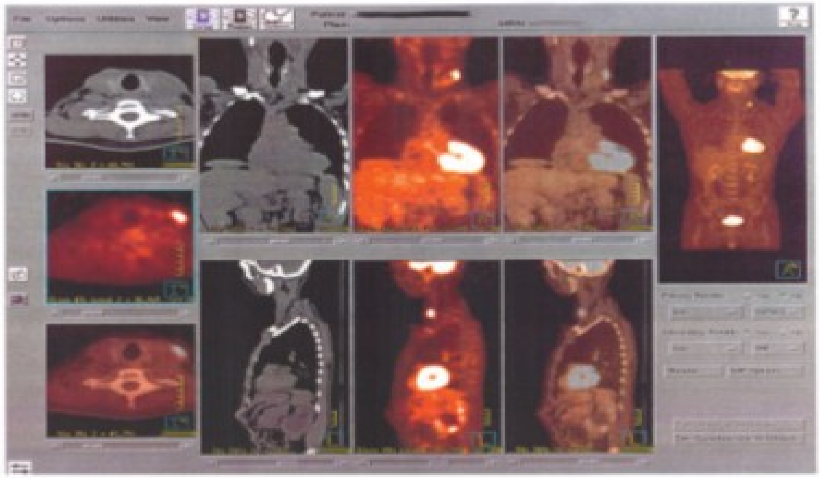
Motivation

- Medical image analysis



(a)

(b)

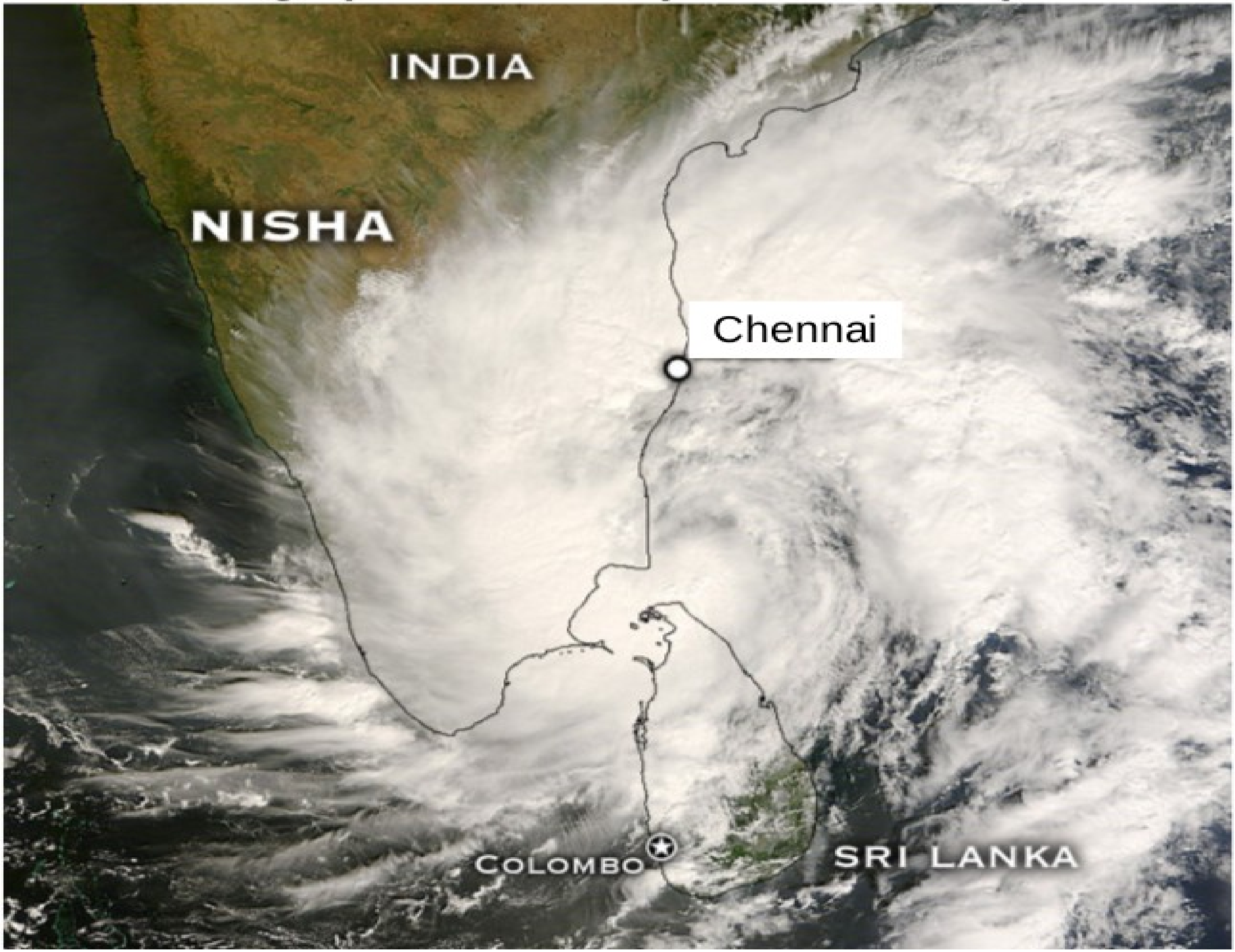


Motivation

- Remote sensing (classify: water, forested, urban, ...)

Motivation

- Remote sensing (weather prediction)

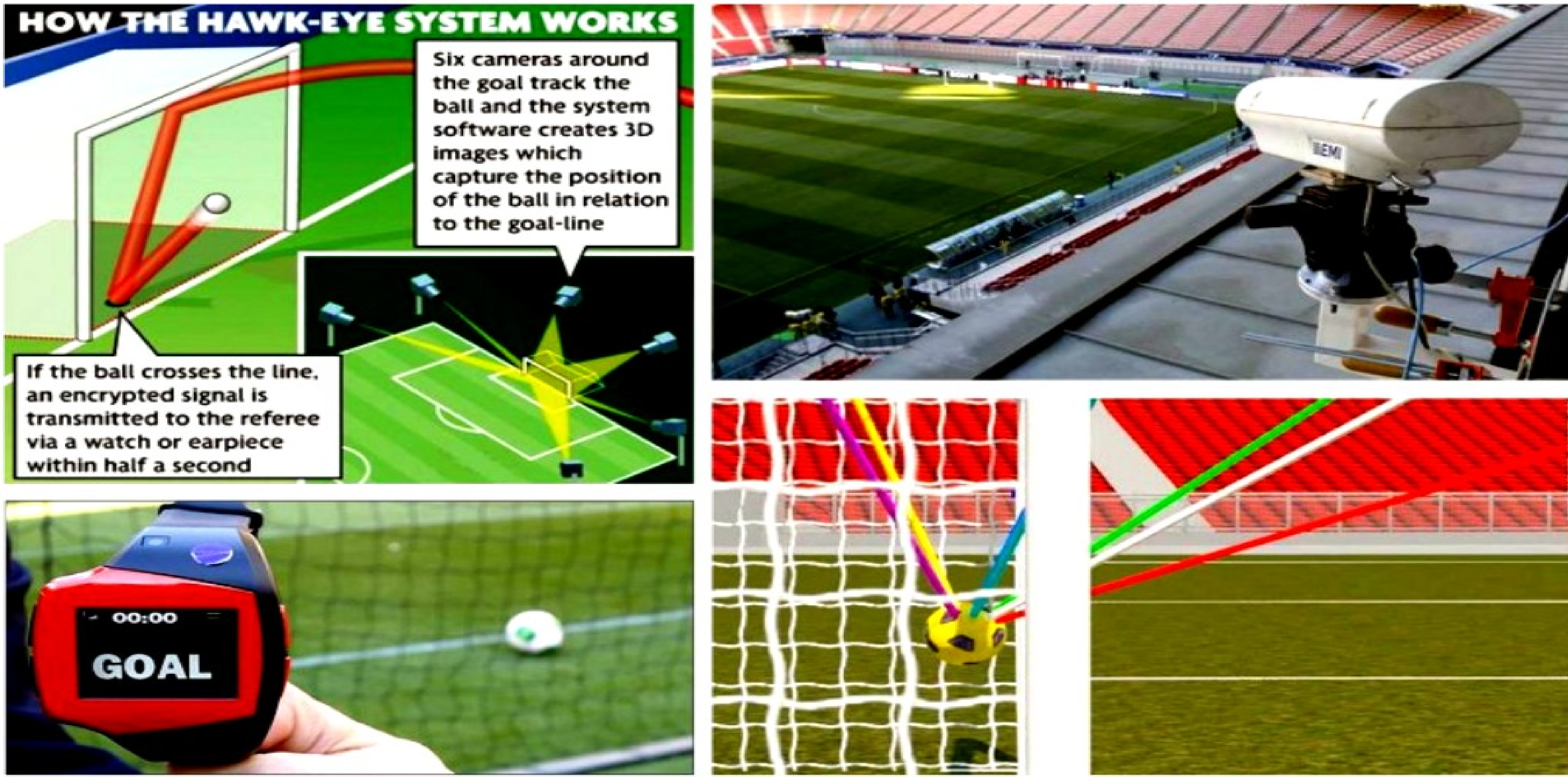


Motivation

- Surveillance

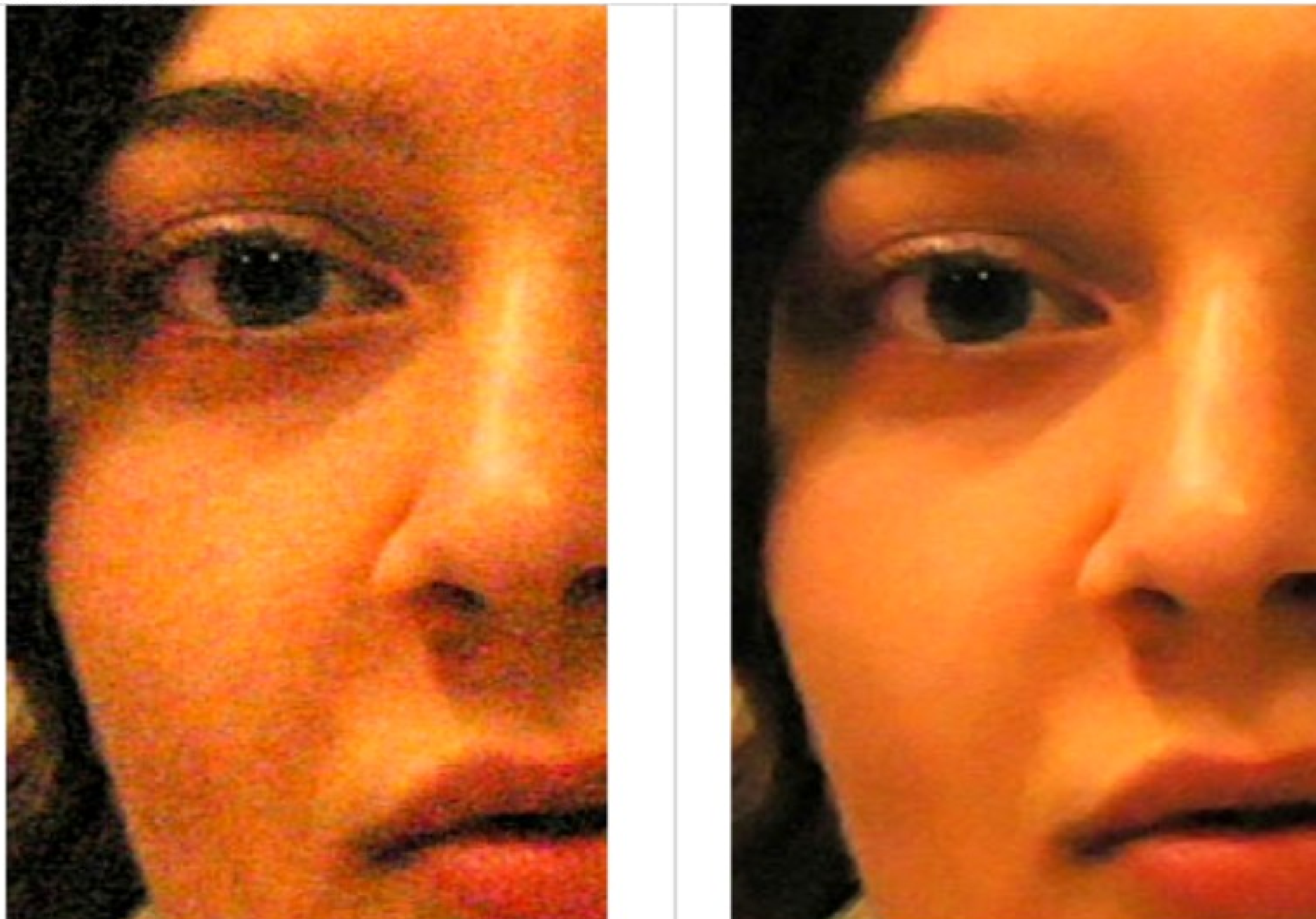
Motivation

- Sports



Motivation

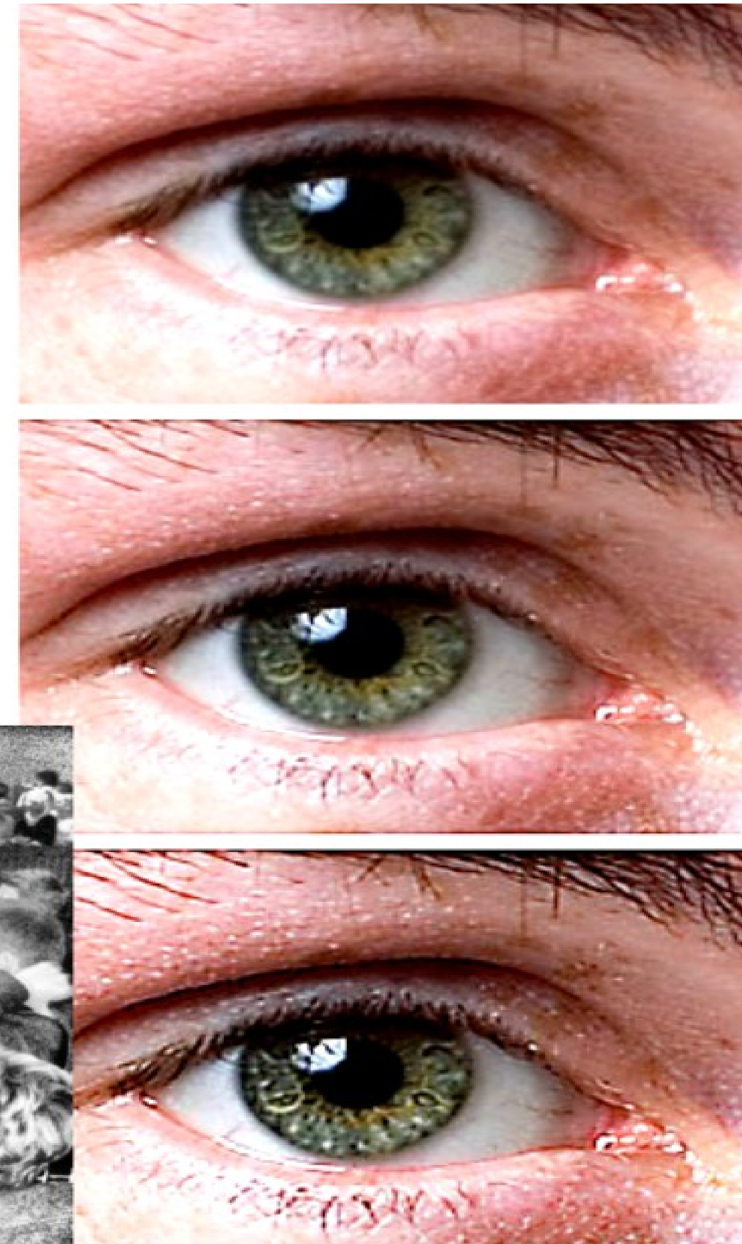
- Image restoration
  - Denoising



Motivation

Applications

- Image enhancement
  - Contrast increase
  - Sharpen



Let's Get Started

- We will talk about digital images here.
- We will be using OpenCV as our toolkit. So along with the concepts I will start refering the OpenCV functions and terminologies as well.
- We will have a demo at the end of the session.

Digital Images

- Digital (Discrete) Image
  - **Array / grid** of numbers
    - Integers, Real
    - Signed, Unsigned (non negative)
    - Many dimensions
  - What makes data discrete ?
    - Acquisition
      - e.g., charge-coupled device (CCD) array in camera
    - Representation
      - e.g., digitization : analog print → scanned digital copy

$$\begin{array}{cc} \text{8 rows} & \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix} \\ & \text{8 columns} \end{array}$$

- *Mat* is a data structure to represent such discrete 2D arrays

Contrast and Brightness

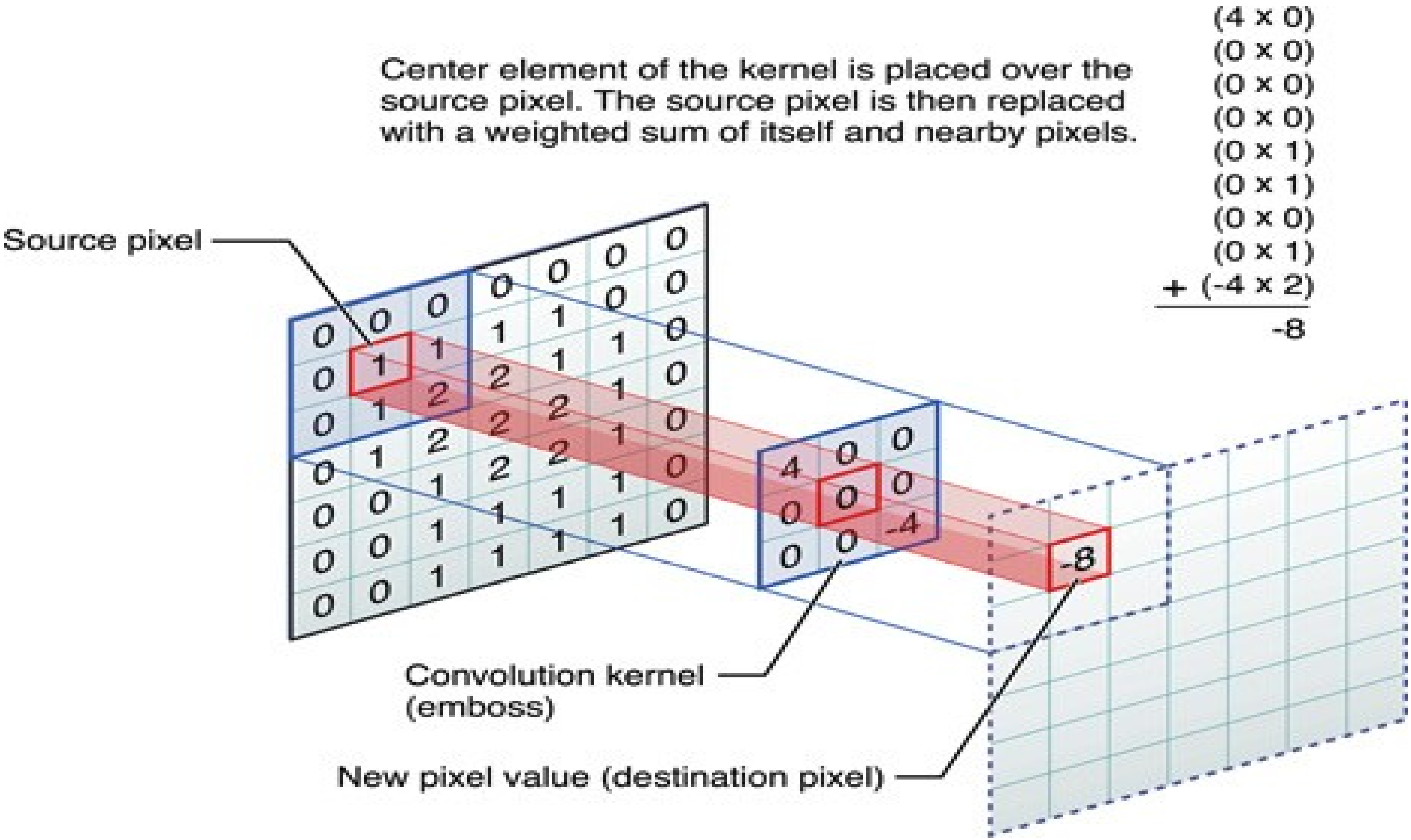
Two commonly used point processes are *multiplication* and *addition* with a constant

$G(x,y) = AF(x,y) + B$

The parameters $A>0$ and B control the *contrast* and *brightness* parameters.

Here $F(x,y)$ is the input image and $G(x,y)$ is the output image.

Kernal Operations: Convolution



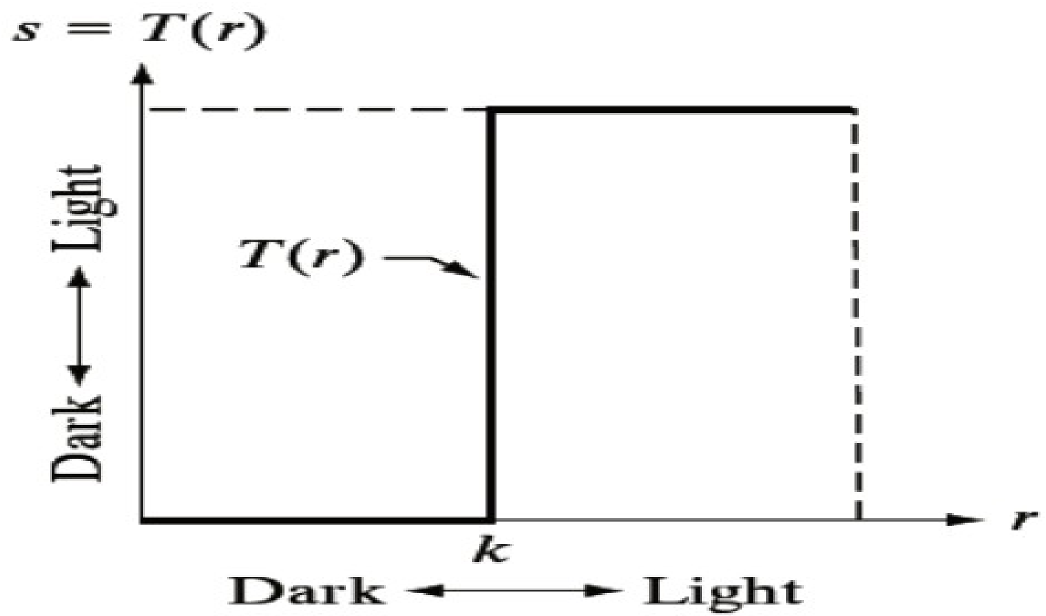
Operation1: Blurring

- Blurring or smoothing is one of the most common operations on image using kernal convolution.
- Average blur,Gaussian Blur, Median Blur are few commonly used blur kernals.

$$K = \frac{1}{K_{width} \cdot K_{height}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ . & . & . & \cdots & 1 \\ . & . & . & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

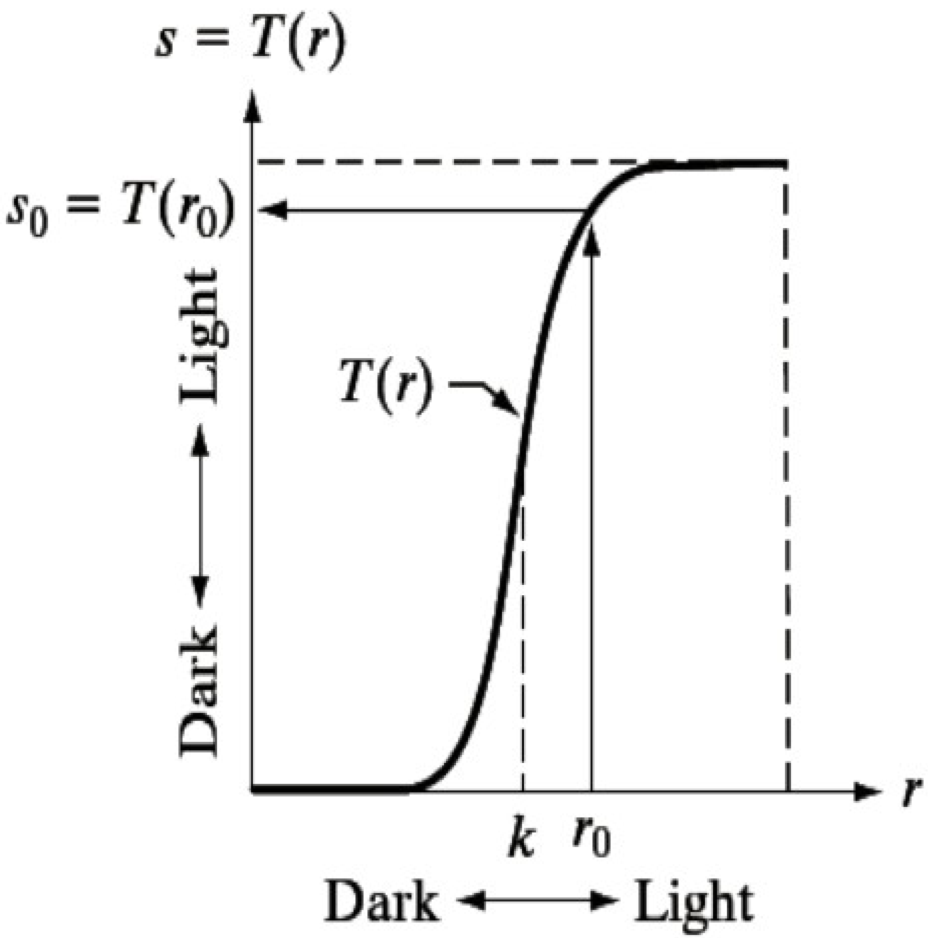
Operation2: Thresholding

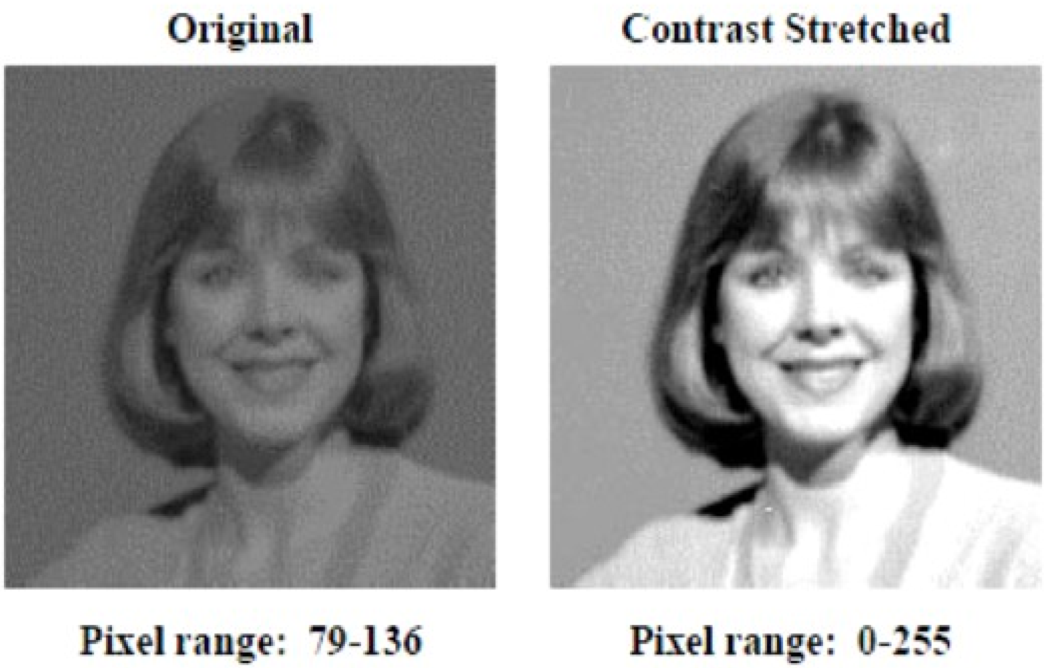
- Binary and Inverted Binary are two common thresholding avaliable in OpenCV, there is a function named threshold. Thresholding can be understood by this graph.



Operation3: Contrast Stretching

- Contrast stretching





Operation4: Edge Detection

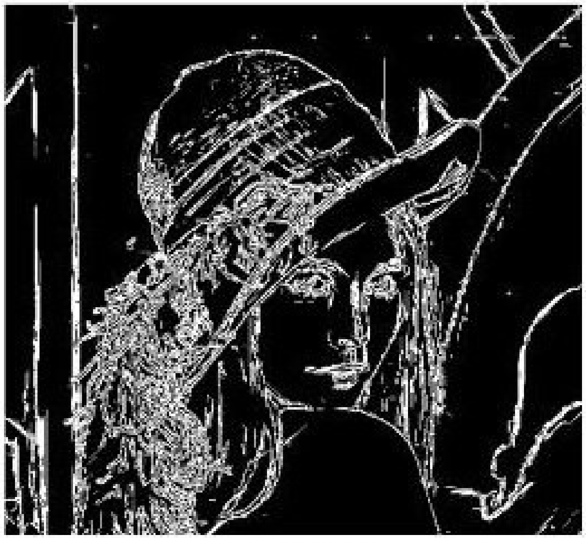
- There are many edge detection algorithms, most popular are Sobel, Laplace and Canny, these three have their functions predefined in OpenCV. The basic concept is the same.
- When we take the derivative of the image we can find spikes at edges as over there we get a sharp change in intensity.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

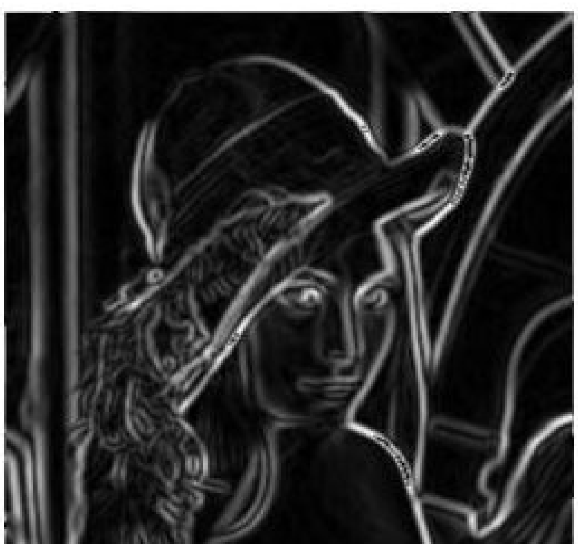
$$G = |G_x| + |G_y|$$

Edge Detection



Canny

Roberts



Prewitt

Sobel

Any Questions??

Histogram Processing: 1

- For image enhancement, histogram based methods are really good.
- We will look at a really common and simple method called histogram equilization. In OpenCV we have a function as

  equalizeHist(src, dst);
- What Histogram Equalization does is to stretch out this range. It takes the values from poplulated areas and give it to unpopulated ones in the spatial(intensity) domain.
- For working:

  http://en.wikipedia.org/wiki/Histogram_equalization

Histogram Processing: 2

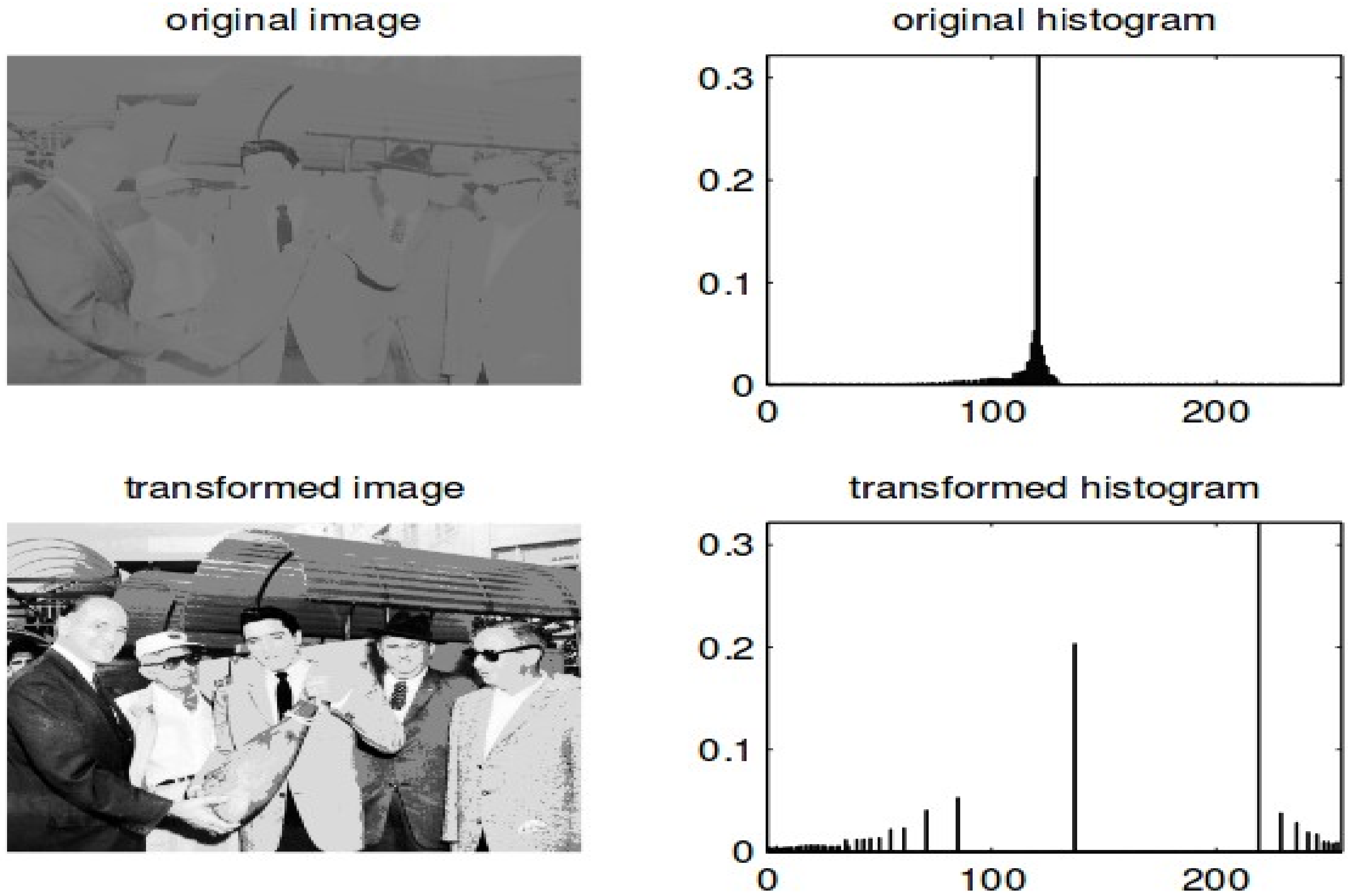


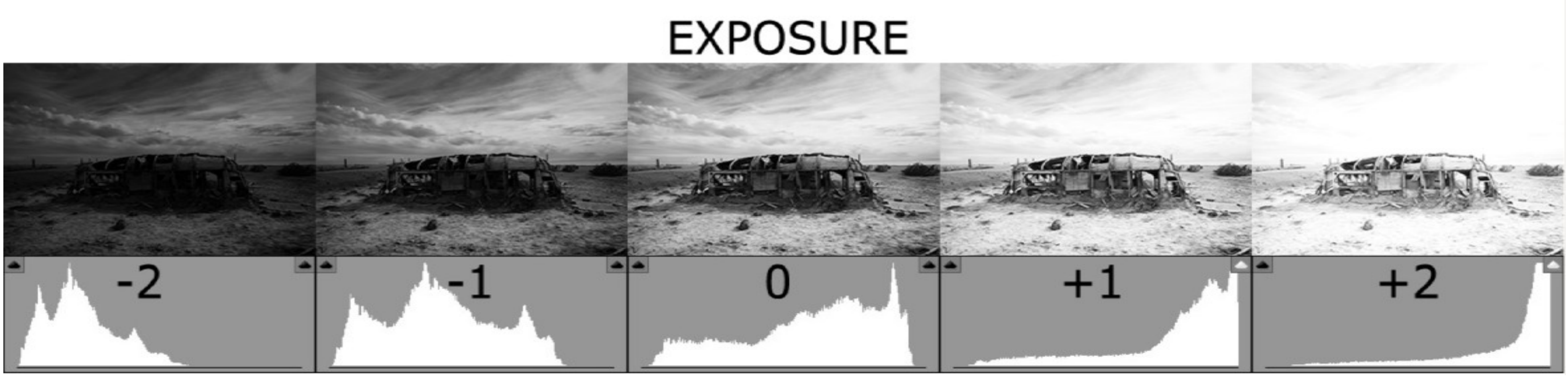
Figure 1: Histogram equalization applied to low contrast image

Histogram Processing: 3

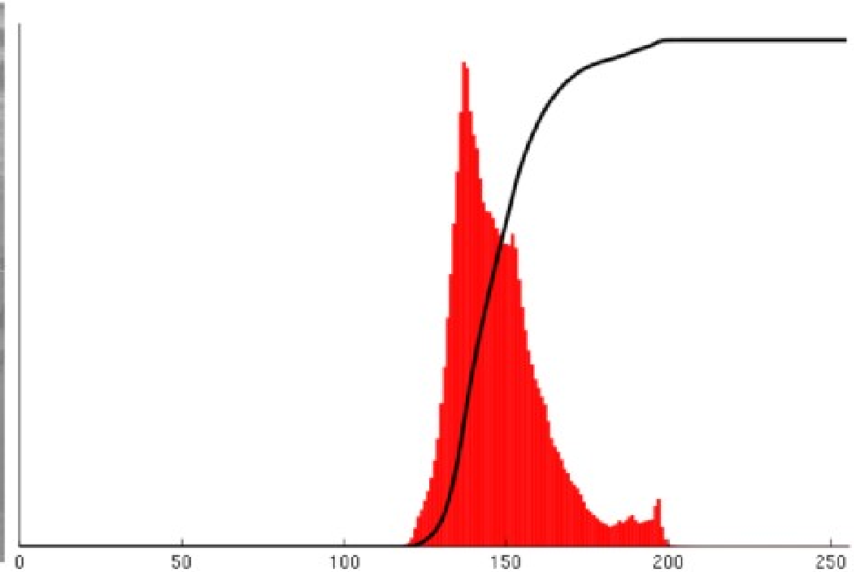
- Adaptive Histogram Equilization (AHE) and Contrast Limited Adaptive Histogram Equilization(CLAHE) are better versions of this algorithm. If interested you can have a look here.

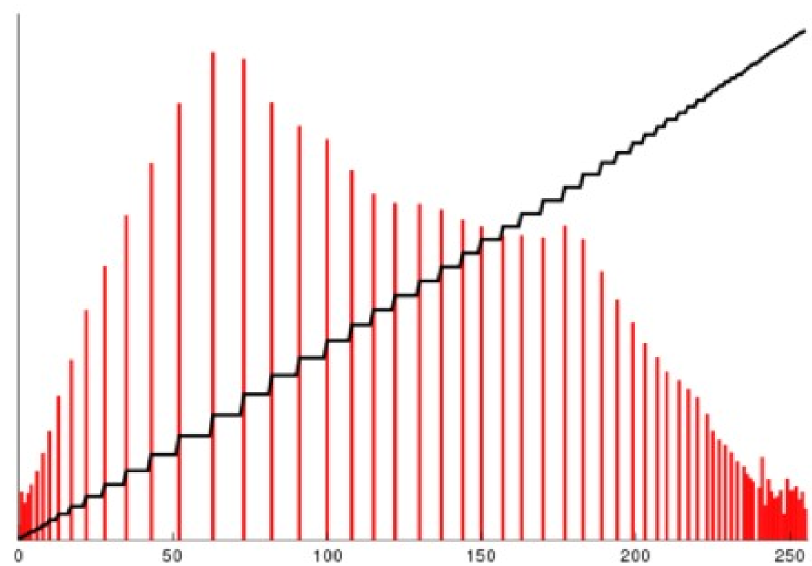
  http://en.wikipedia.org/wiki/Adaptive_histogram_equalization

- Histograms are related to photograph exposure
  - Aperture size (amount of light per unit time)
  - Shutter speed (time of exposure)

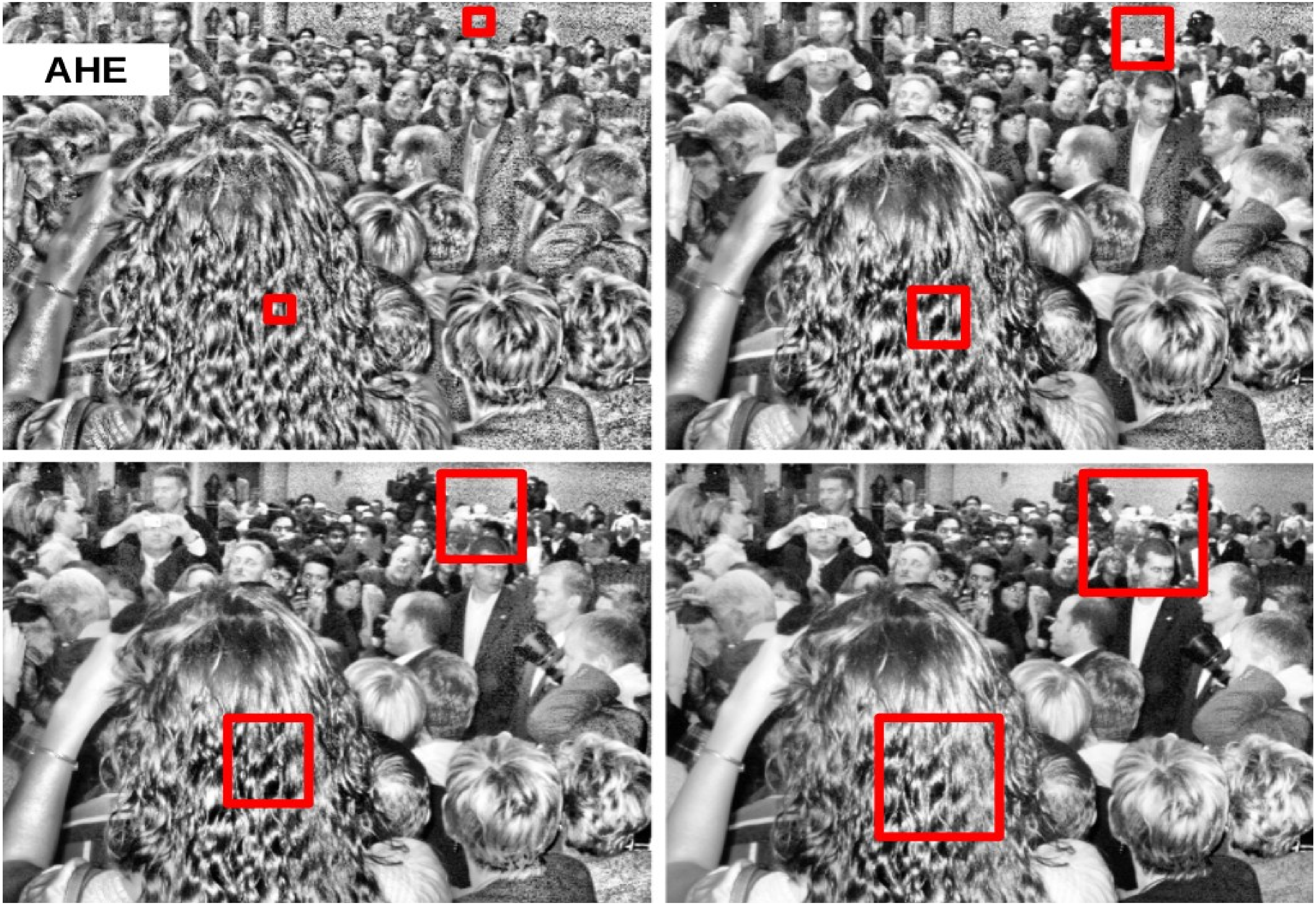


Histogram Processing: 4





Histogram Processing: 5



Histogram Processing: 6

Any Questions??

Fuzzy!!

- I like to cover fuzzy based image enhancement, so if everyone is willing to do this we will do it.
- If we are doing it we can do it after the demo of before it.

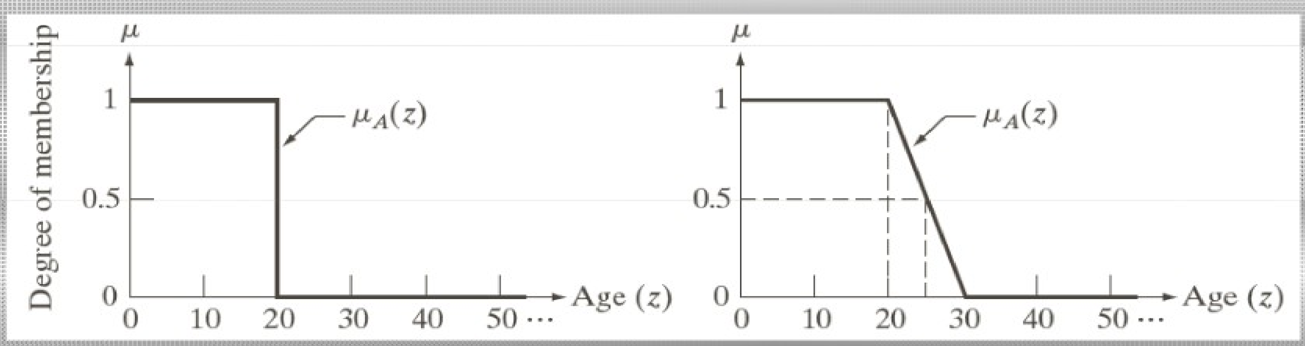
Fuzzy Logic

- We will perform contrast enhancement using this method.
- It's a set theory method for expressing data more reliastically.

A fuzzy set is defined in terms of a membership function $\varphi_A : U \rightarrow [0,1]$.

A *characteristic function* is a special case of a membership function and a regular set is a special case of a fuzzy set.

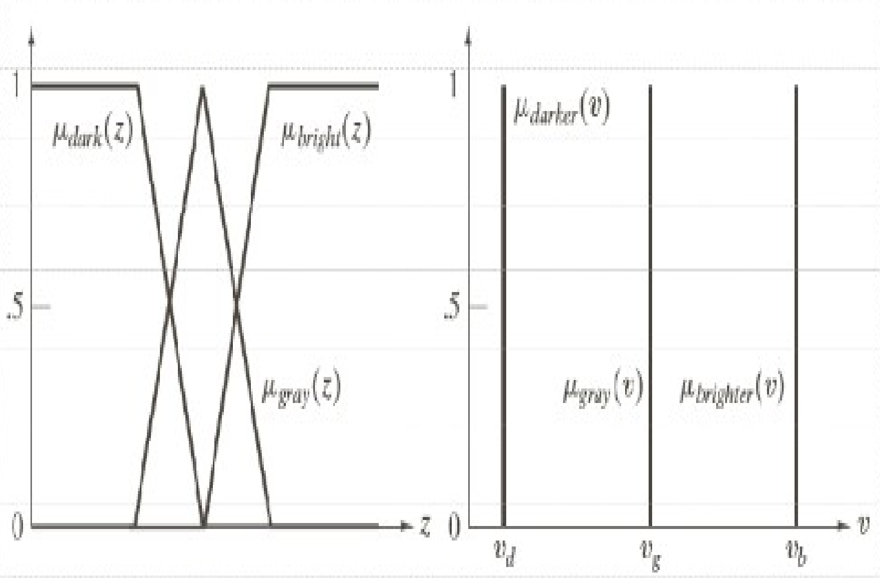
Illustrating The membership functions of regular and fuzzy set



Fuzzy: Contrast Enhancement 1

Using fuzzy sets for intensity transformation

1. Define a set of rules to change pixel intensity.
2. Transfer the rules into fuzzy set
3. User the rules to change intensity



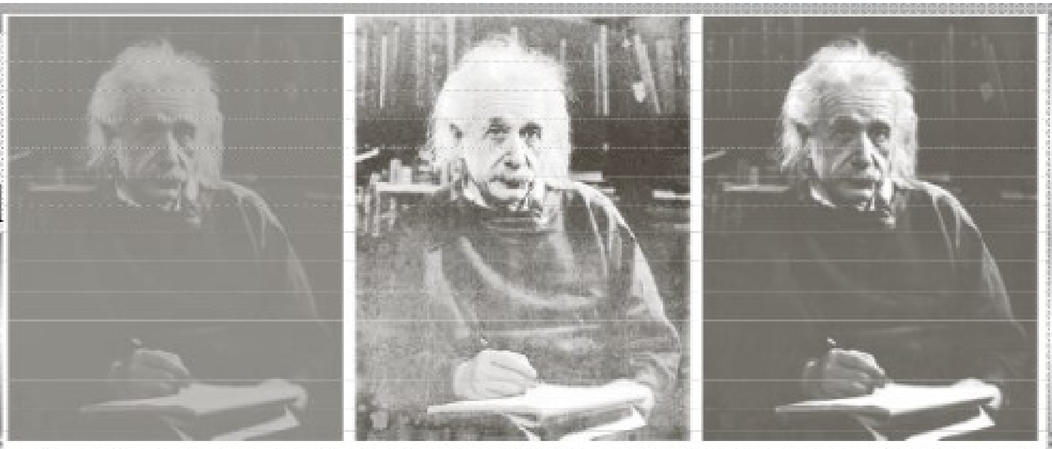
Example:

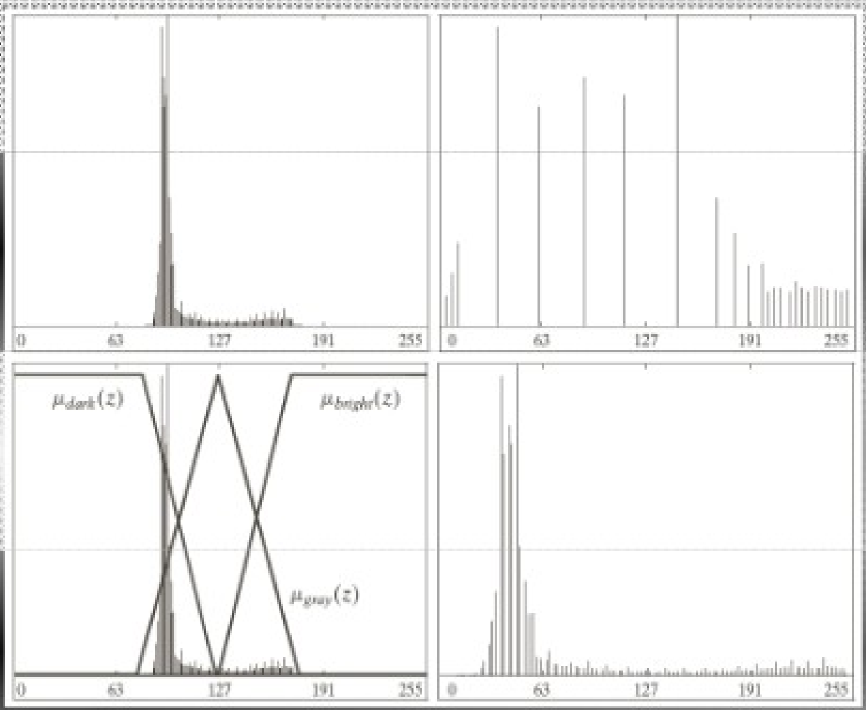
1. If a pixel is dark, then make it darker
2. If a pixel is gray, then make it gray
3. If a pixel is bright, then make it brighter

Fuzzy: Contrast Enhancement 2

Using fuzzy sets for intensity transformation

1. Define a set of rules to change pixel intensity.
2. Use fuzzy set to apply this rules.





Things For Demo : 1

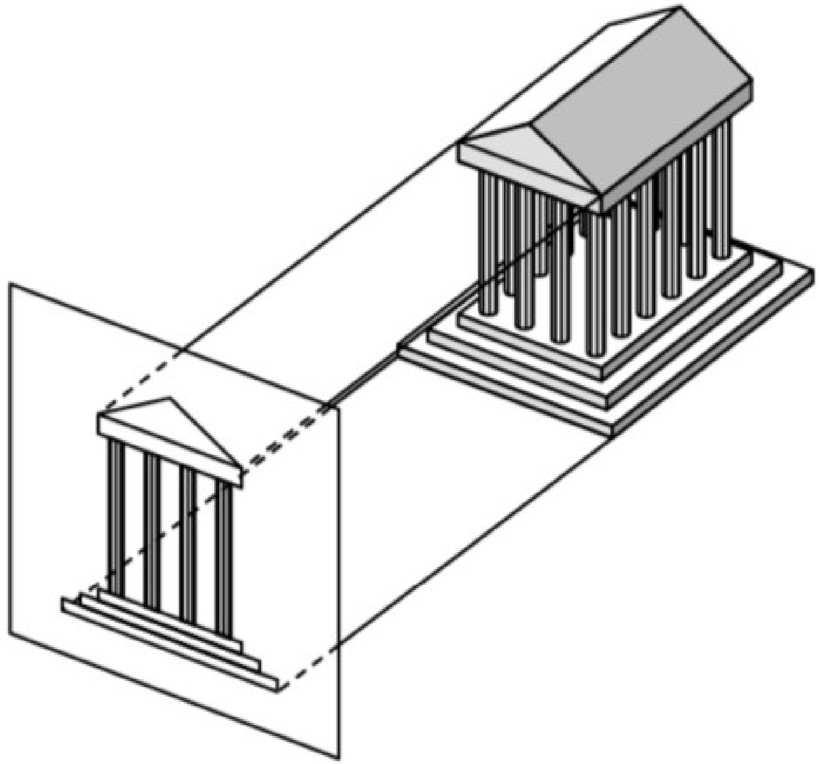
- Basic definition is given by the function

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

- Area (for binary images) or sum of grey level (for greytone images): M_{00}
- Centroid: { x, y } = {M_{10}/M_{00}, M_{01}/M_{00} }
- Useful in tracking applications, how to use moment functions will be explained in demo.

Things For Demo : 2

Orthographic or Parallel Projection



http://www.cs.cmu.edu/afs/cs/academic/class/15462-s09/www/lec/06/lec06.pdf

Pendulum

- The folder given to you will have a video of a makeshift pendulum that I recorded yesterday in orthographic projection. We will find the length of this pendulum and hence the time period.
- Given:

> 4 screenshots of the video instances.

> The breadth of the image is 480 in pixels and 11 inches in world (see the scale in the side of the video).

Pendulum

- Flow of the code:

  > We read the 4 screenshots.

  > We process each of them sequentially with techniques we have learned in a logical fashion.

  > We get the center of the hanging ball in each image say a1,a2,a3,a4.

  > use SOT to find r for each 4C3 combinations and average to get r_avg in pixels.

  > then use 11inches/480 pixels = length/r_avg.

  >Voila!

Thank You