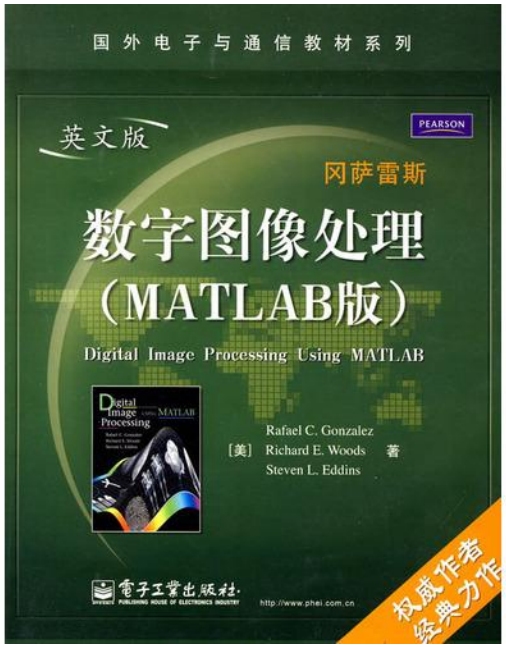
MATLAB and Its Application in Engineering

Assoc. Prof. Kirin Shi

Shanghai Jiao Tong University

Digital Image Processing

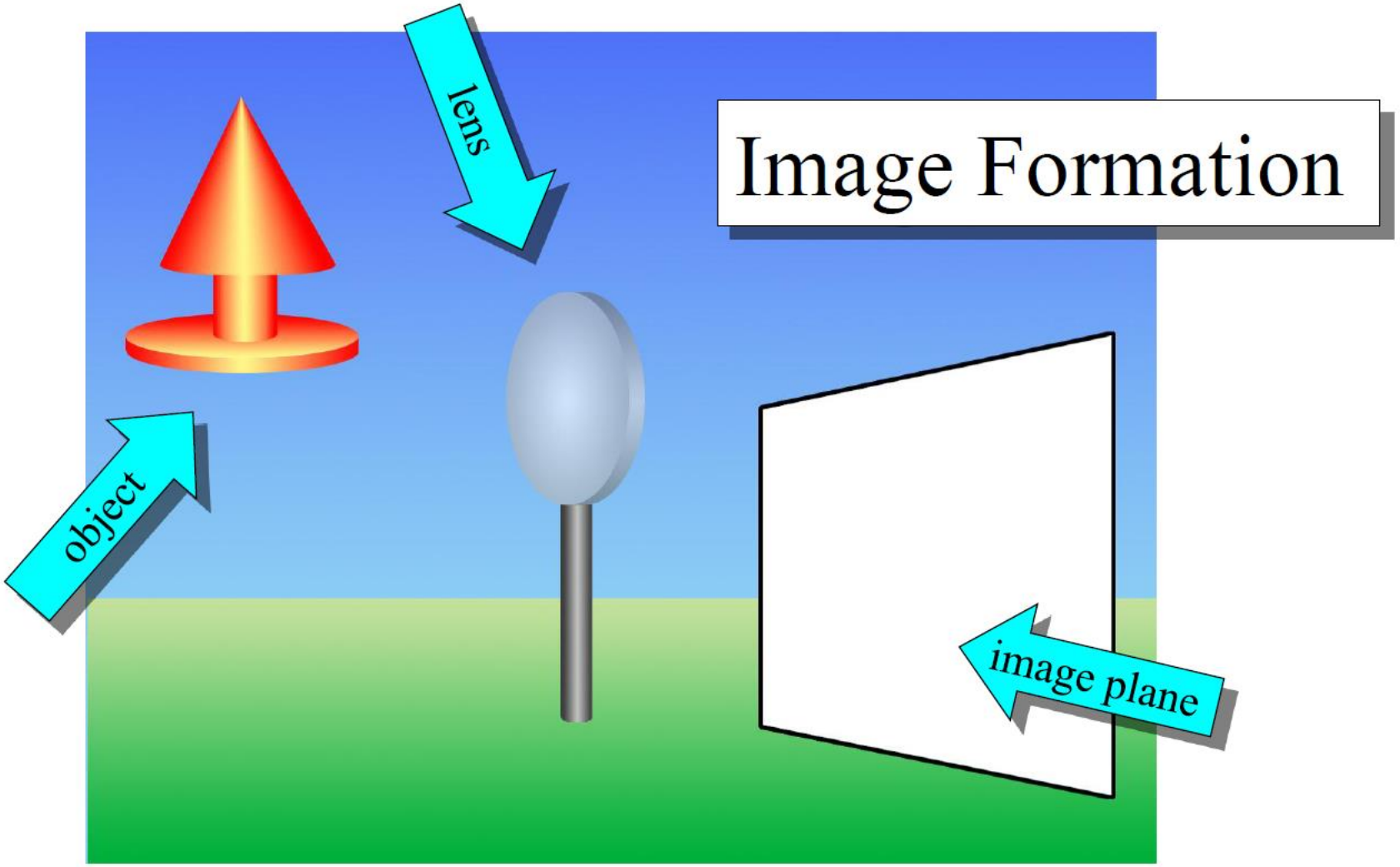


"Digital Image Processing using Matlab"

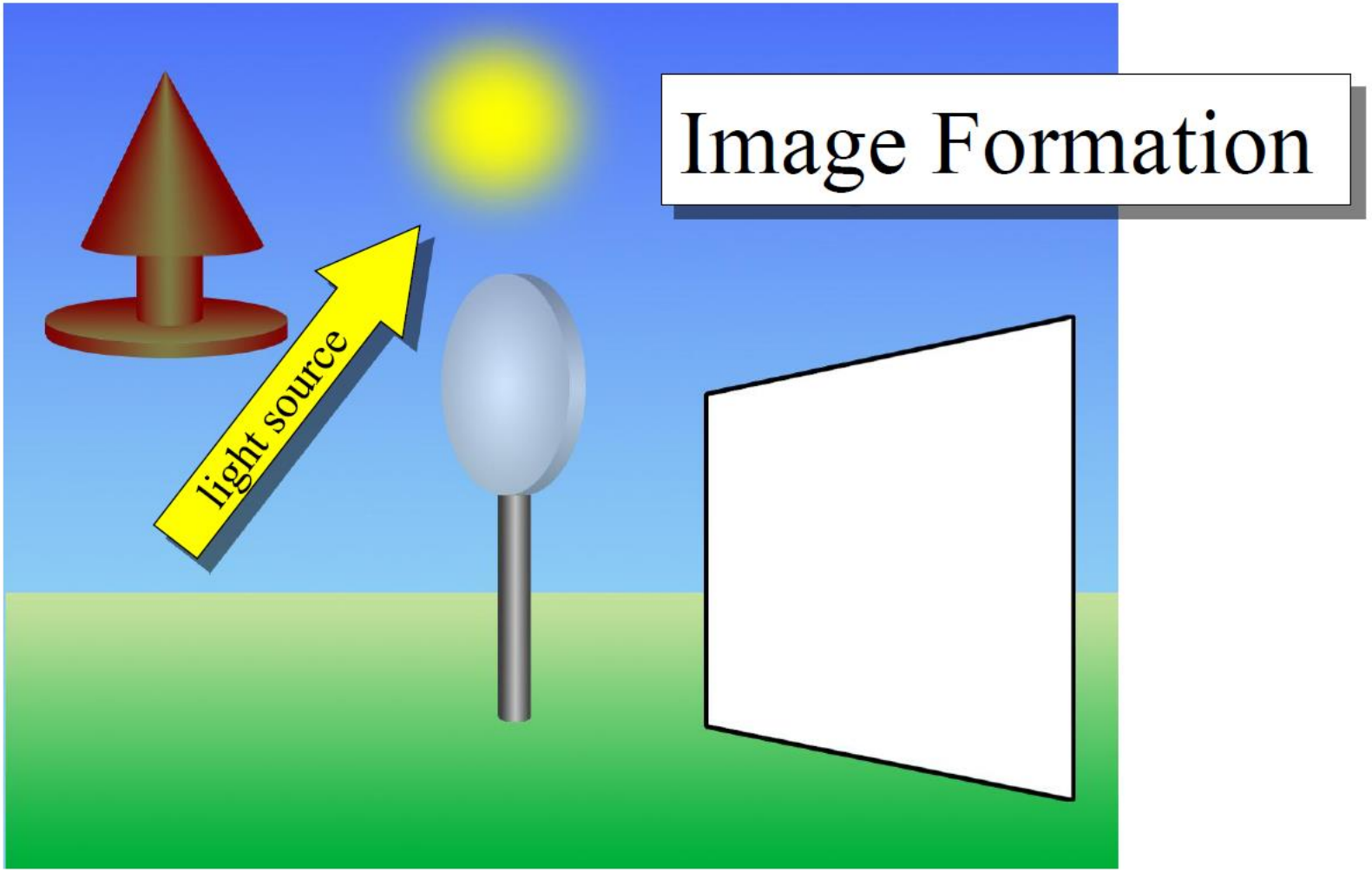
Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins

Pearson Prentice Hall, 2003

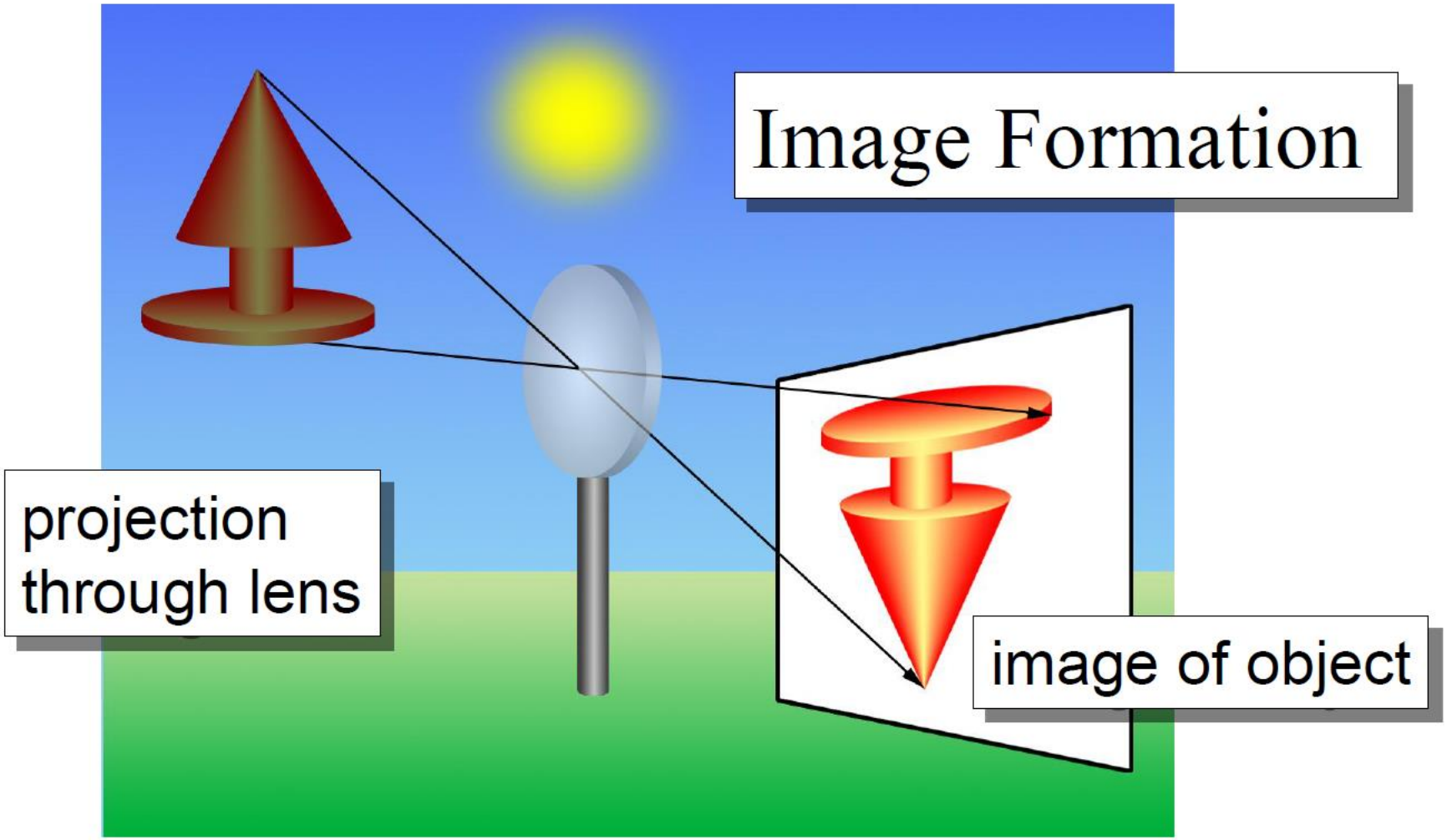
Digital image in Matlab



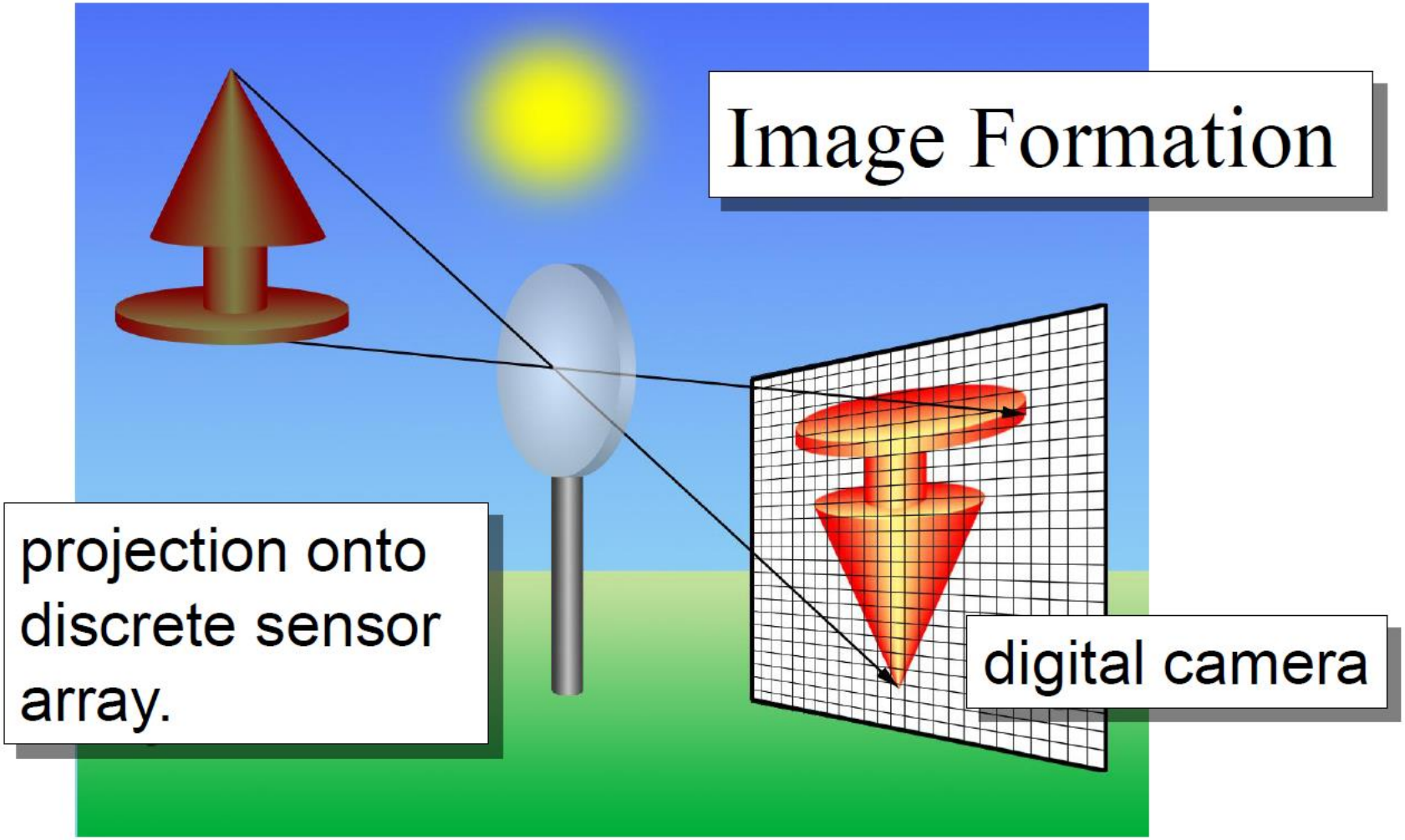
Digital image in Matlab



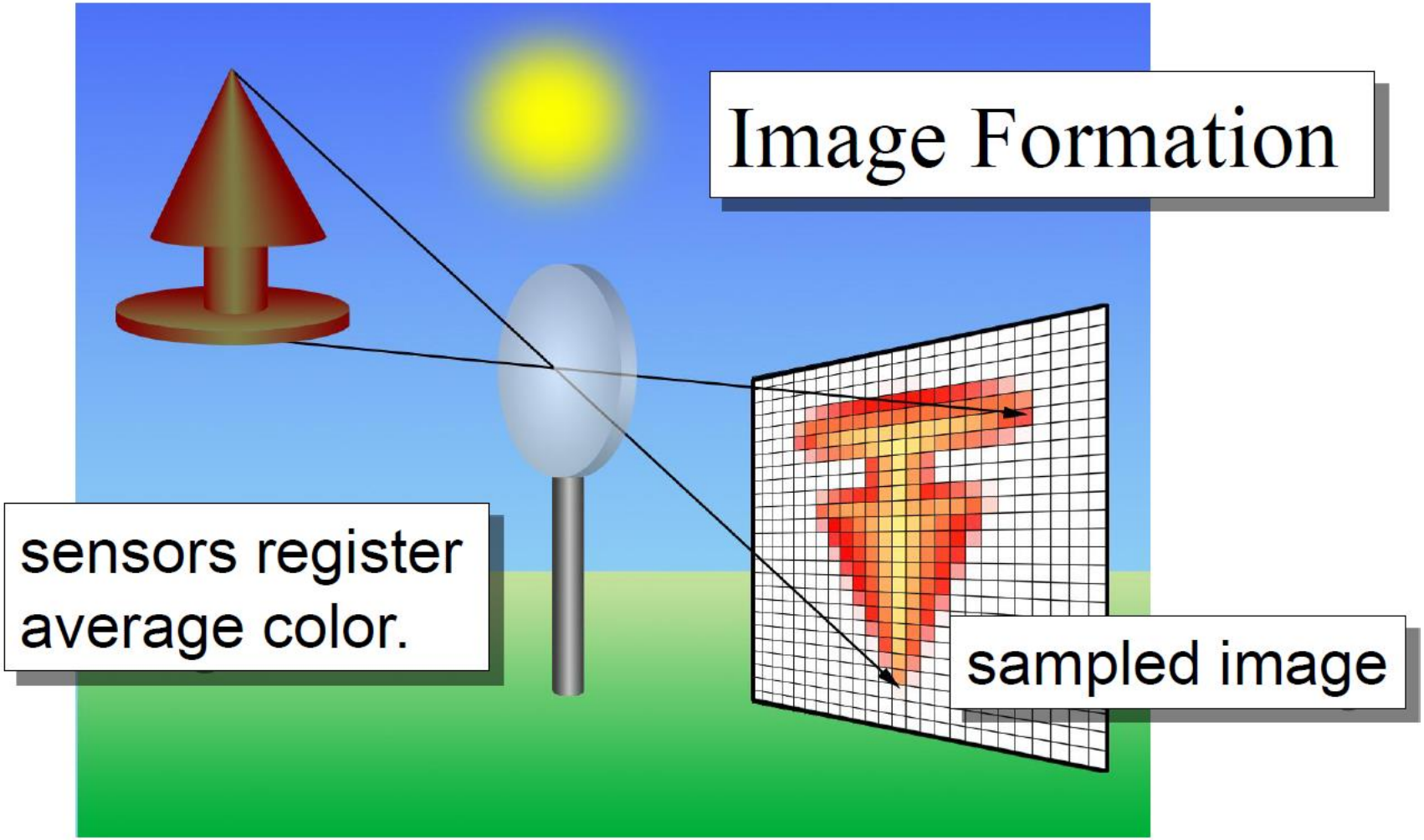
Digital image in Matlab



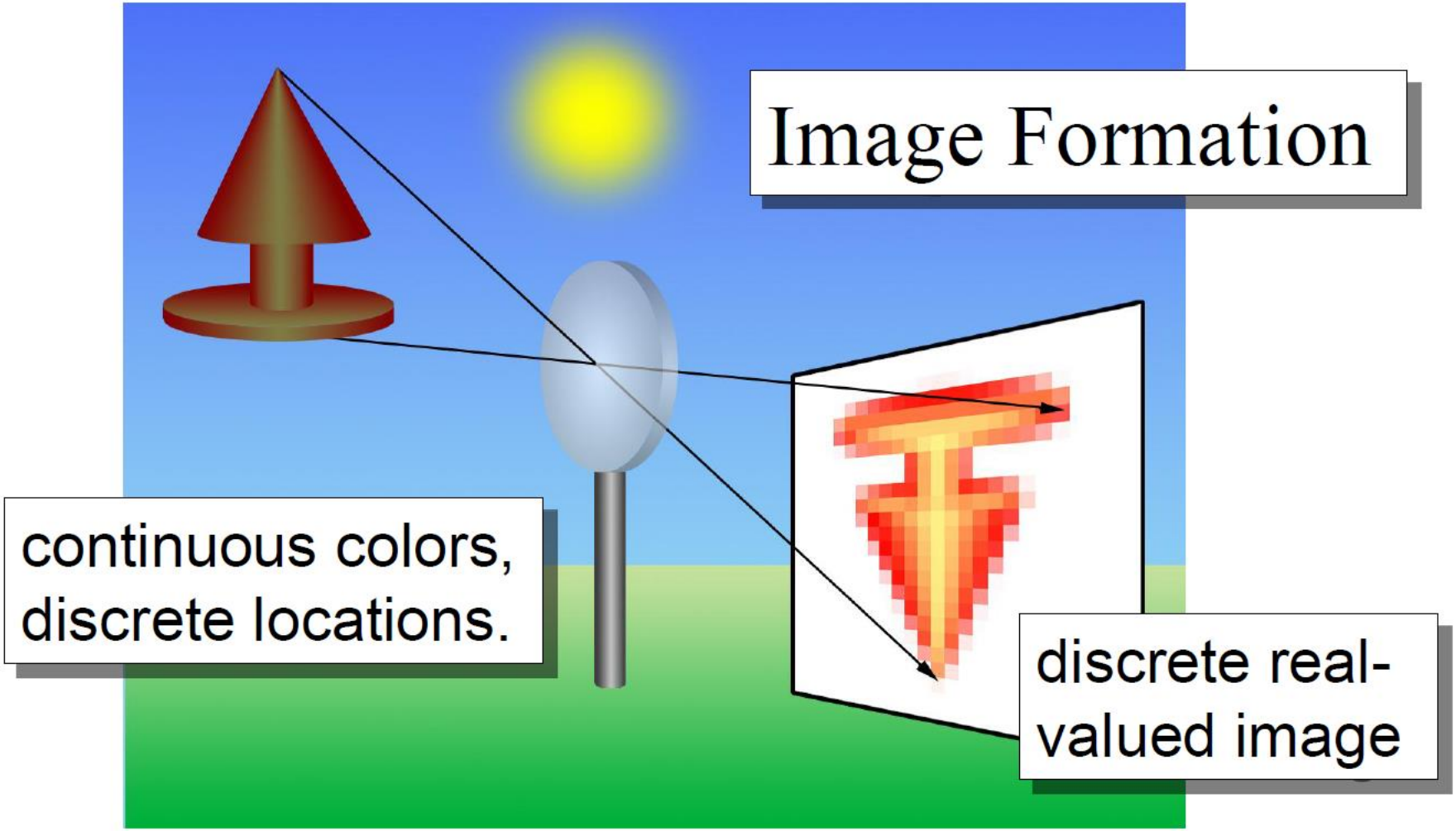
Digital image in Matlab



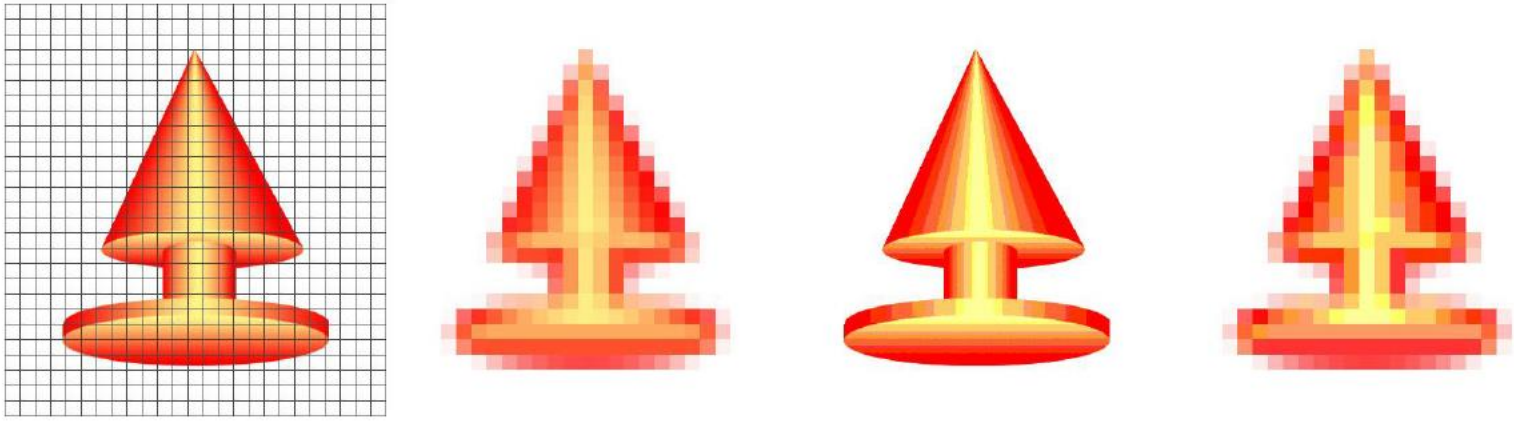
Digital image in Matlab



Digital image in Matlab



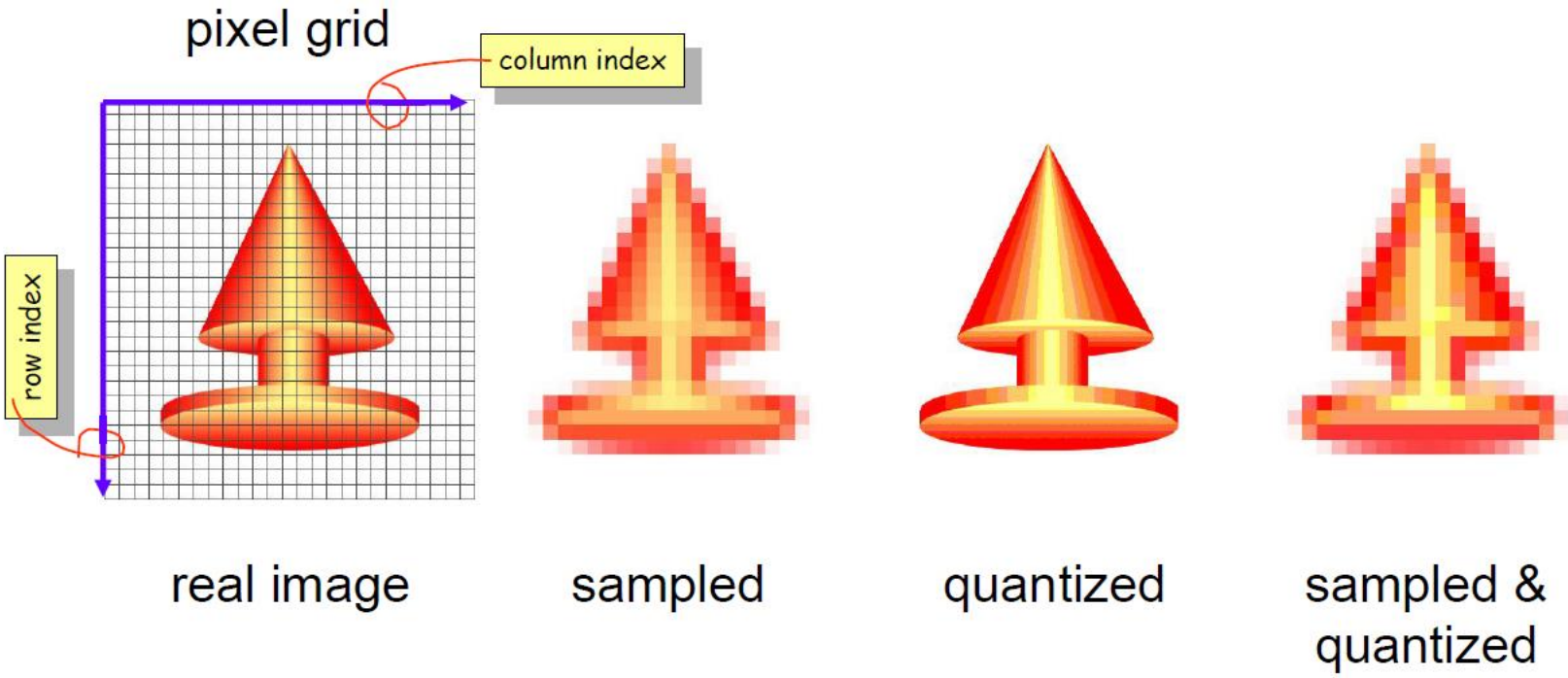
Sampling and Quantization



real image | sampled | quantized | sampled & quantized

Digital image in Matlab

Sampling and Quantization



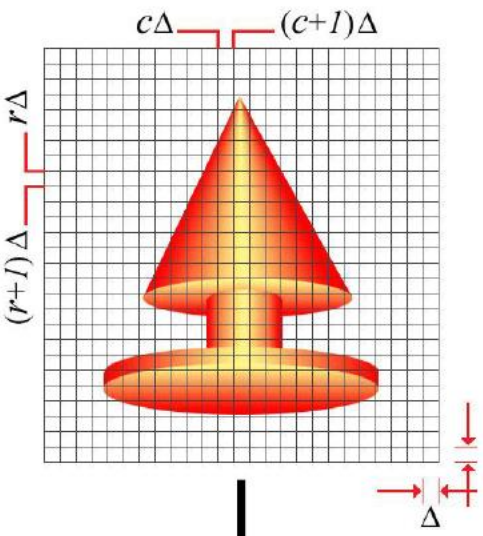
Digital image in Matlab

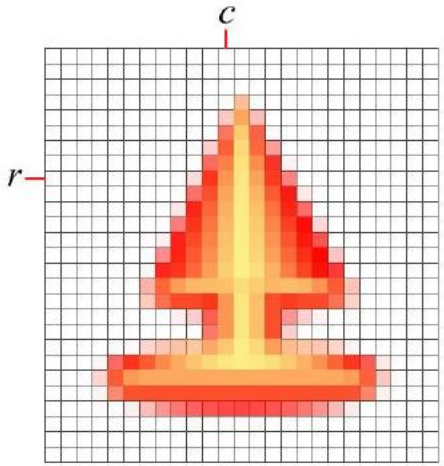
Sampling

Take the average within each square.

$I_C(\rho,\chi)$

continuous image





$I_S(r,c)$

sampled image

$$I_S(r,c)=\frac{1}{\Delta^2}\int_{r\Delta}^{(r+1)\Delta}\int_{c\Delta}^{(c+1)\Delta} I_C(\rho,\chi)\delta\rho\delta\chi$$

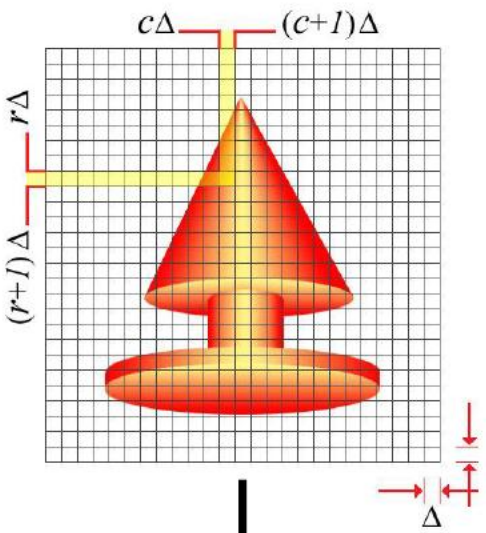
Digital image in Matlab

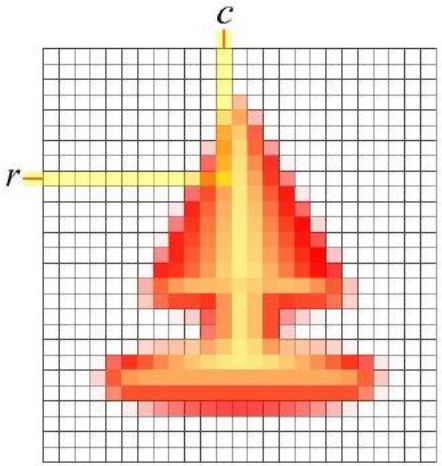
Sampling

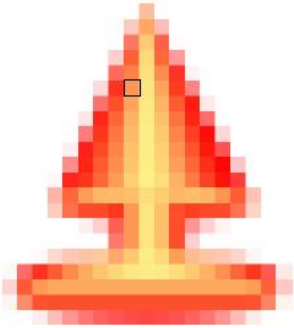
Take the average within each square.

$I_C(\rho,\chi)$

continuous image







$I_S(r,c)$

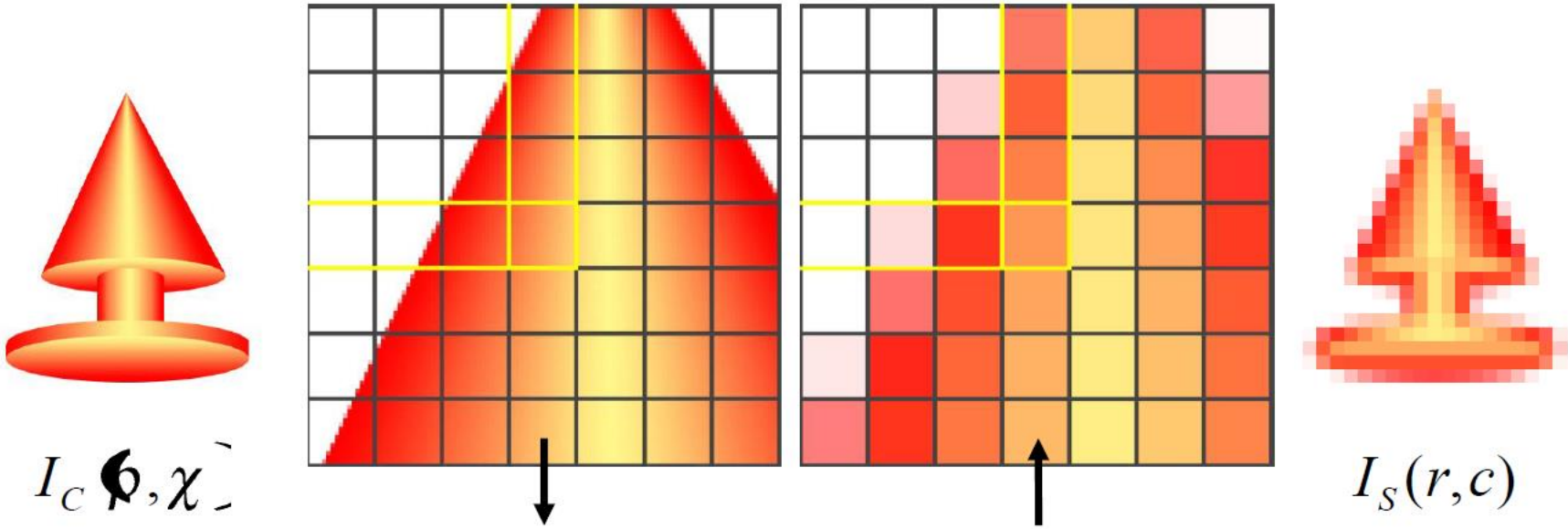
sampled image

$$I_S(r,c)=\frac{1}{\Delta^2}\int_{r\Delta}^{(r+1)\Delta}\int_{c\Delta}^{(c+1)\Delta} I_C(\rho,\chi)\delta\rho\delta\chi$$

Digital image in Matlab

Sampling

Take the average within each square.



$I_C(\rho,\chi)$

continuous image

$$I_S(r,c) = \frac{1}{\Delta^2} \int_{r\Delta}^{(r+1)\Delta} \int_{c\Delta}^{(c+1)\Delta} I_C(\rho,\chi)\delta\rho\delta\chi$$

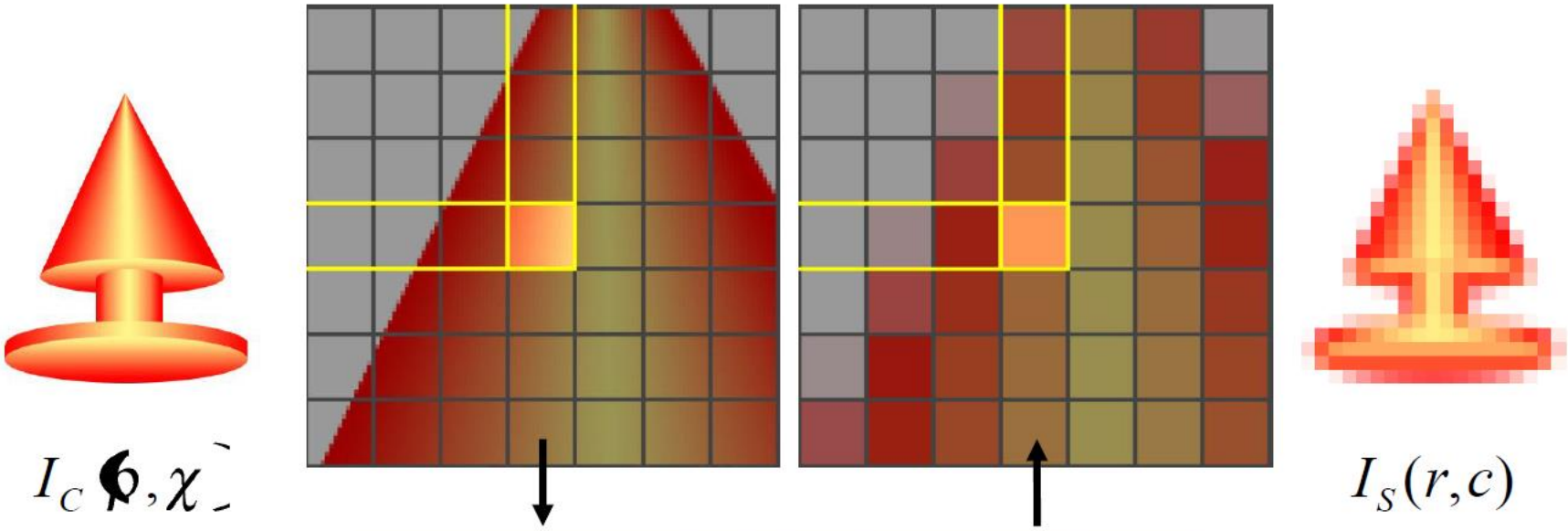
$I_S(r,c)$

sampled image

Digital image in Matlab

Sampling

Take the average within each square.



$I_C(\rho,\chi)$

continuous image

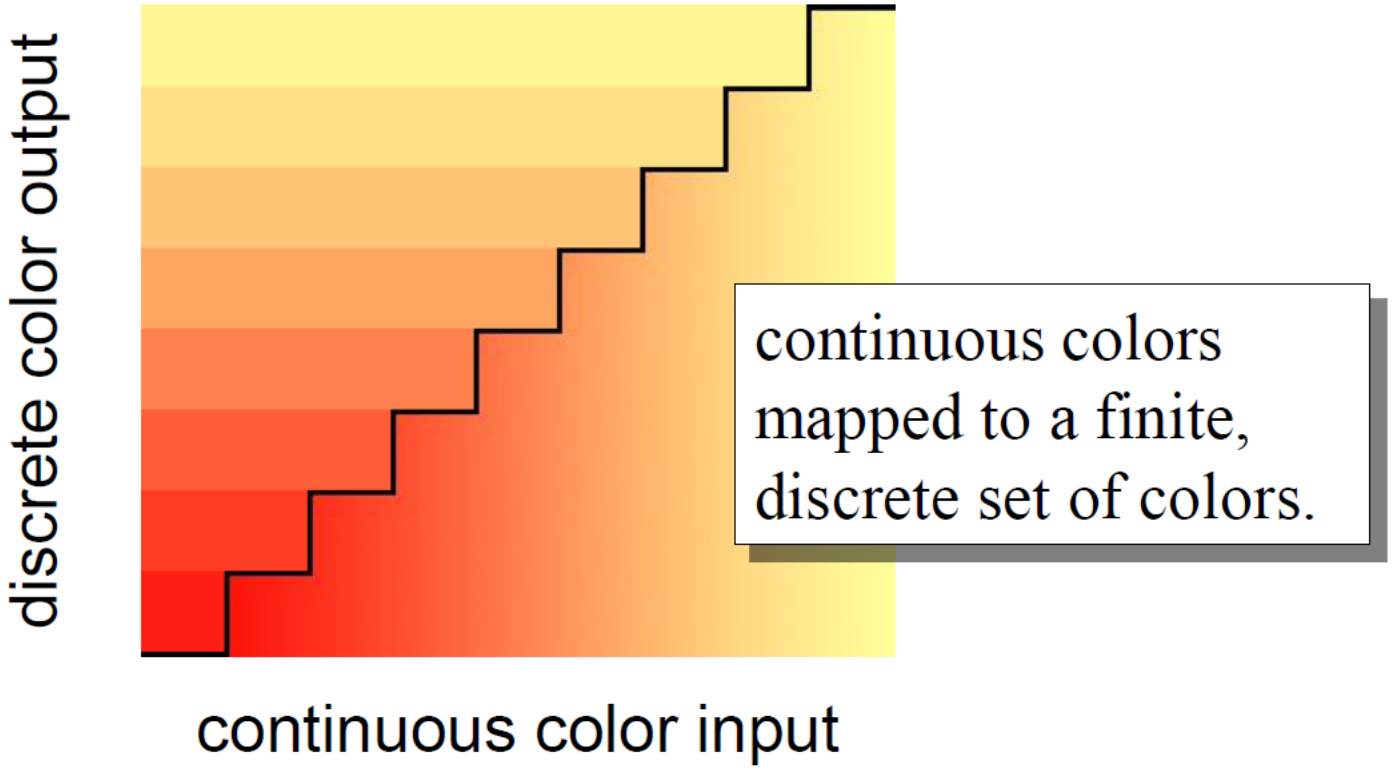
$I_S(r,c)$

sampled image

$$I_S(r,c) = \frac{1}{\Delta^2} \int_{r\Delta}^{(r+1)\Delta} \int_{c\Delta}^{(c+1)\Delta} I_C(\rho,\chi)\delta\rho\delta\chi$$

Digital image in Matlab

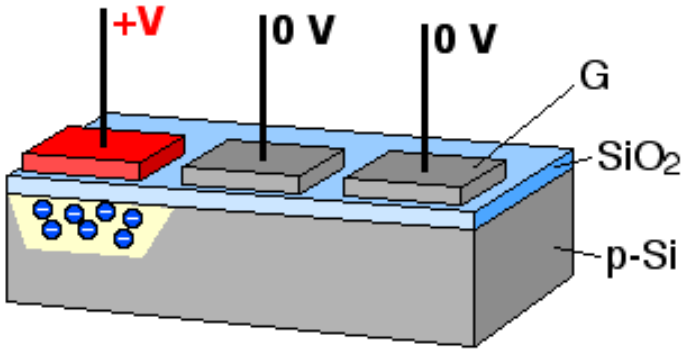
Digital Image Formation: Quantization

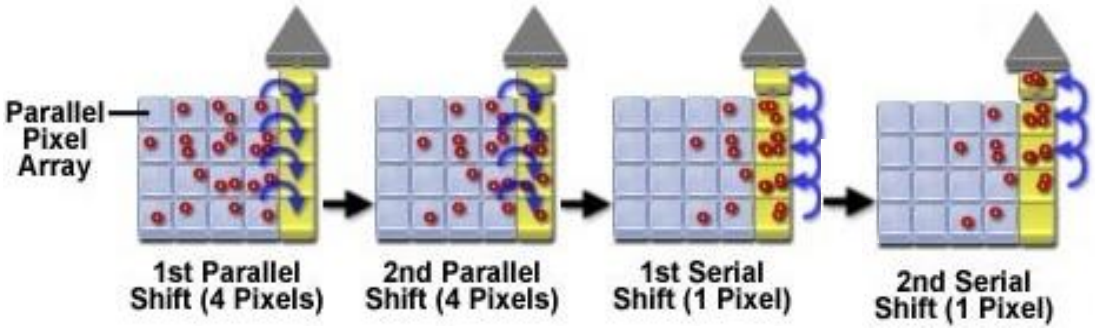


From light to pixel intensity

Shifting photoelectrons

- Positive voltage is applied sequentially to MOS capacitor gates
- Depletion region is created in sequence between adjacent MOS capacitors
- Electrons are then moved across the MOS capacitor array to a readout note (charge-coupled)





Infrared detector

Quantifying photoelectrons

- Photoelectrons are amplified and quantified at the readout note
- Light intensity collected by each pixel is then represented by a digital value
- Bit depth, colour resolution

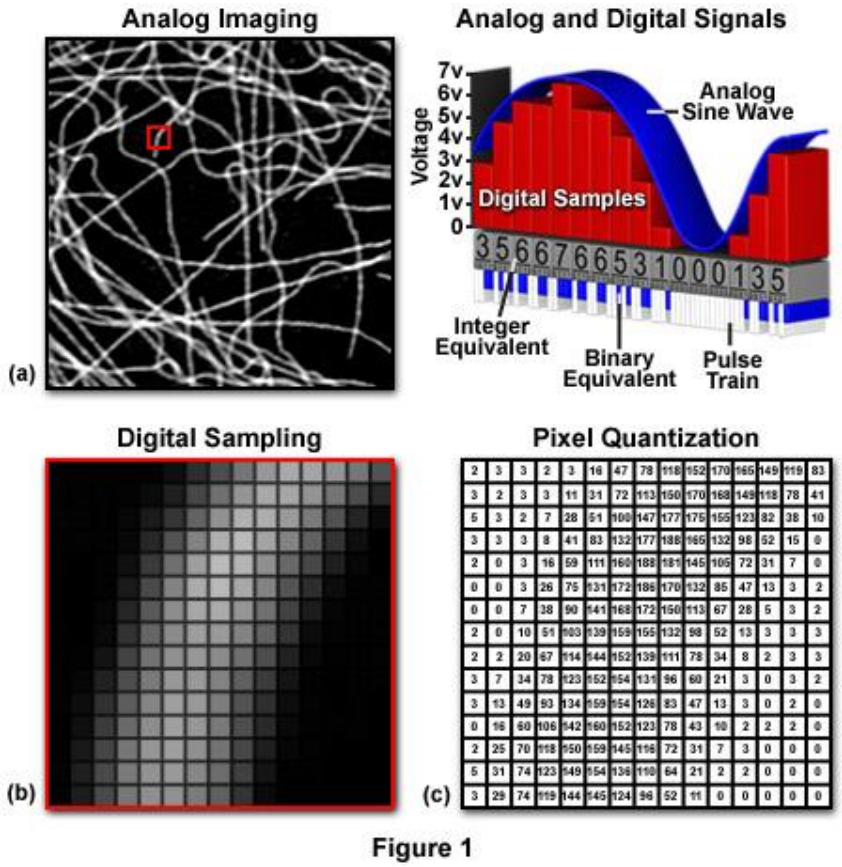


Figure 1

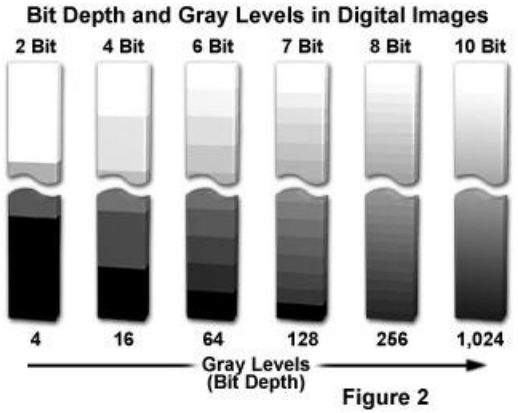


Figure 2

Infrared detector

Detector for visible light

- Silicon
- Germanium (锗)

Detector for infrared light

- Lead sulfide (PbS) (硫化铅)
- Lead selenide (PbSe) (硫化硒)
- Indium arsenide (InAs) (砷化铟)
- Indium antimonide (InSb) (锑化铟)
- ……

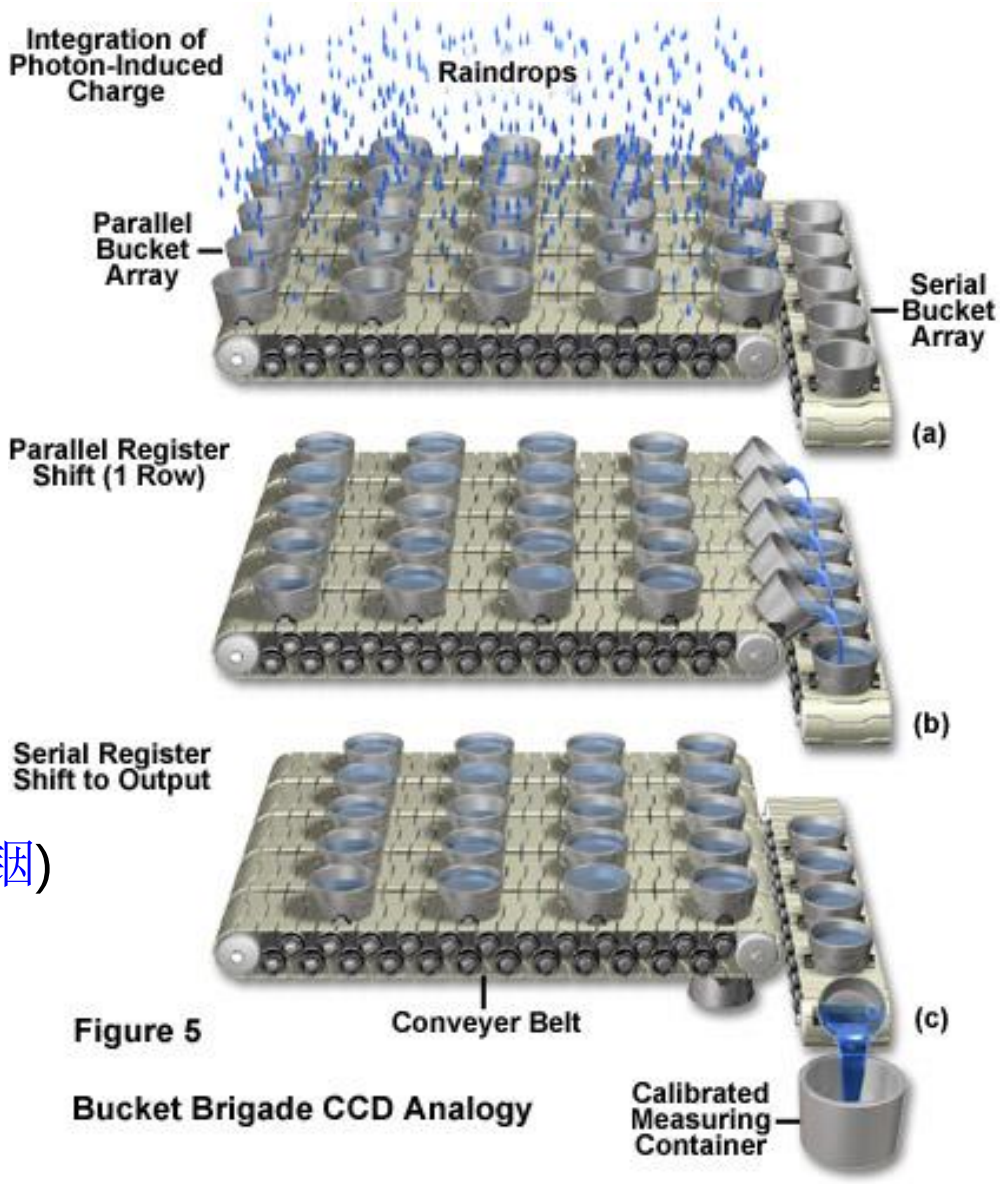


Figure 5

Bucket Brigade CCD Analogy

Image quantisation(example)

256 gray levels (8bits/pixel)

32 gray levels (5 bits/pixel)

16 gray levels (4 bits/pixel)

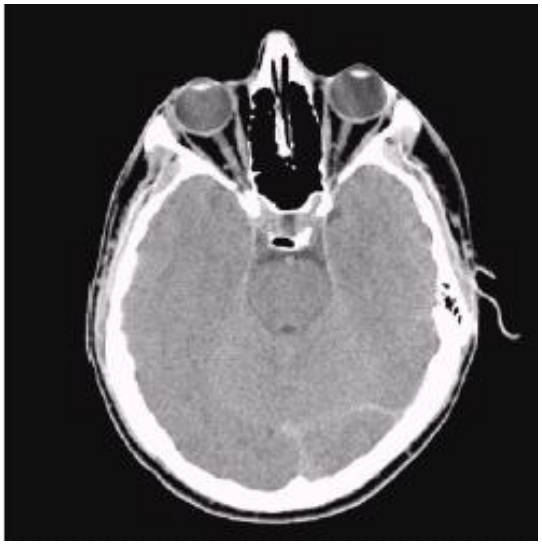
8 gray levels (3 bits/pixel)

4 gray levels (2 bits/pixel)

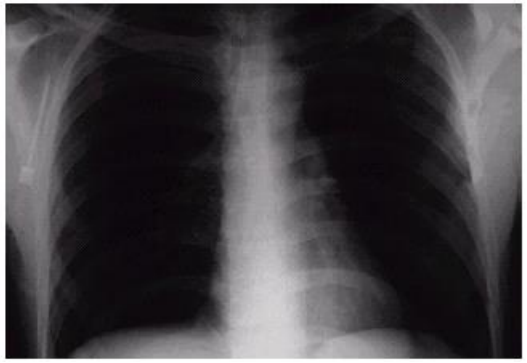
2 gray levels (1 bit/pixel)

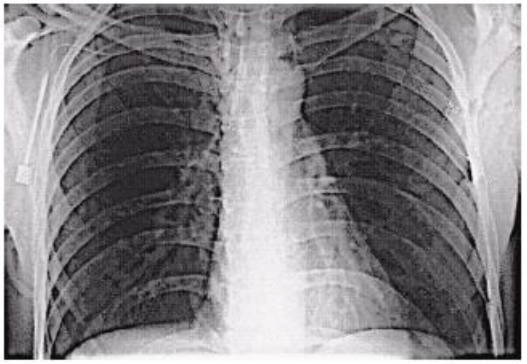
Applications

Medical applications



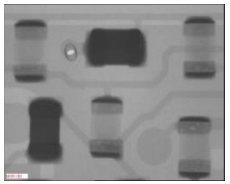
Typical head slice CAT image

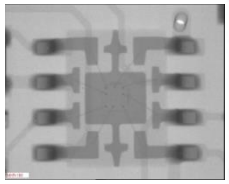


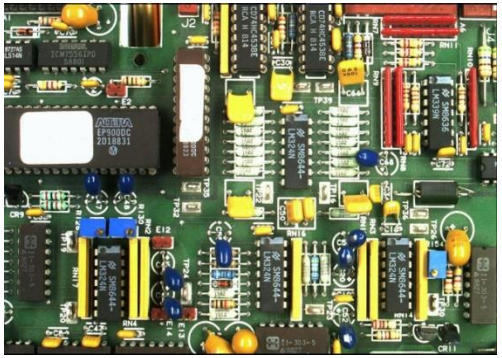


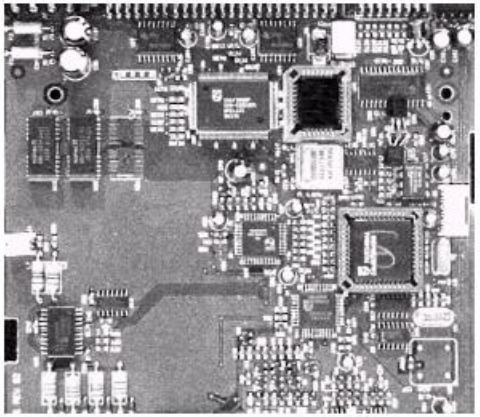
Applications

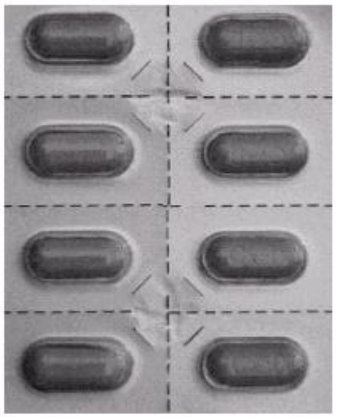
Industrial application

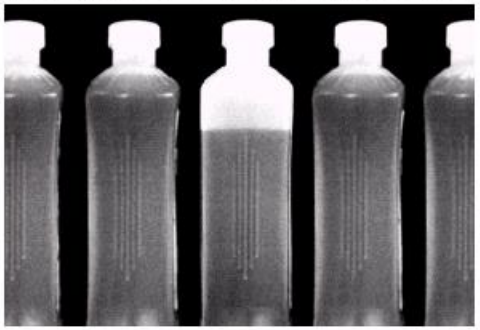


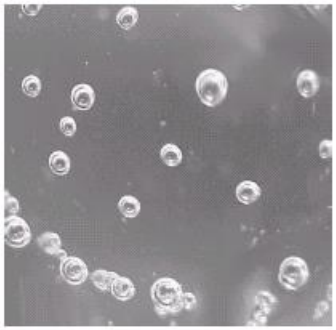


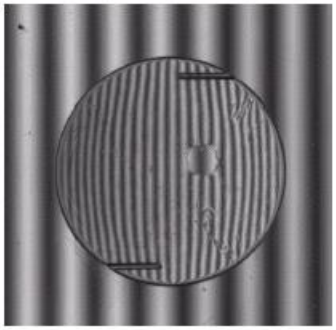










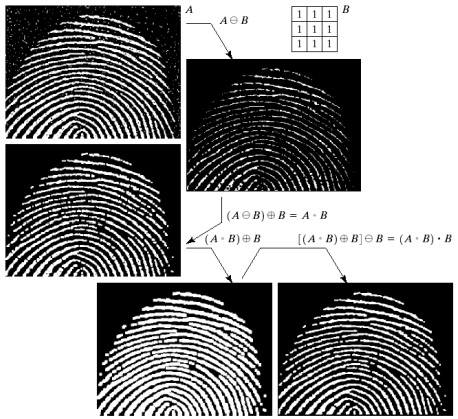


Applications

Image processing techniques are used extensively by law enforcers

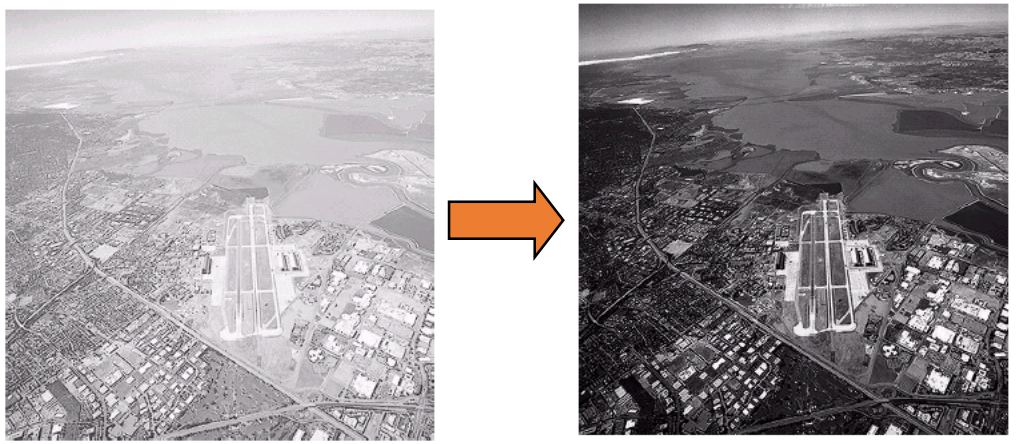
- Number plate recognition for speed cameras

  automated toll systems
- Fingerprint recognition
- Enhancement of CCTV images



Applications

Satellite images



Wide Field Planetary Camera 1

Wide Field Planetary Camera 2

Applications

Digital image in Matlab

Common image formats include:

- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and “Alpha”, a.k.a. Opacity)

For most of this course we will focus on grey-scale images

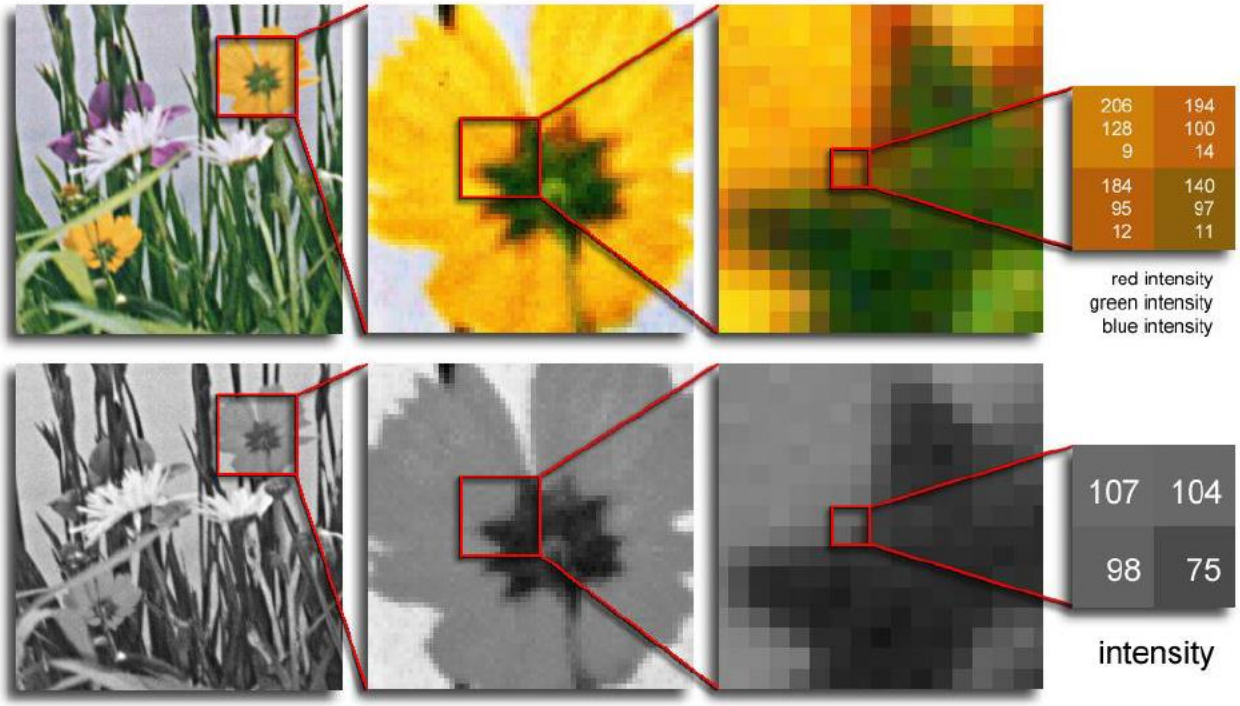
Digital image in Matlab

Digital Image

Color images have 3 values per pixel; monochrome images have 1 value per pixel.

a grid of squares, each of which contains a single color

each square is called a pixel (for *picture element*)



Pixels

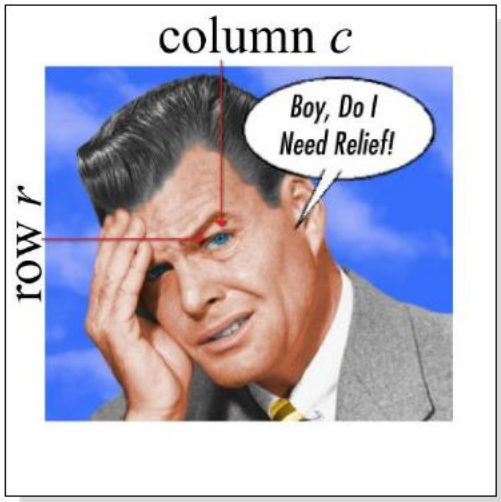
- A digital image, I, is a mapping from a 2D grid of uniformly spaced discrete points, $\{p = (r,c)\}$, into a set of positive integer values, $\{I(p)\}$, or a set of vector values, *e.g.*, $\{[R\ G\ B]^{\mathsf{T}}(p)\}$.
- At each column location in each row of I there is a value.
- The pair (p, $I(p)$) is called a "pixel" (for *picture element*).

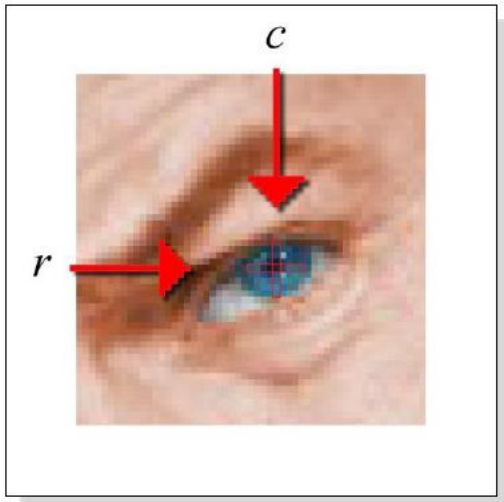
Pixels

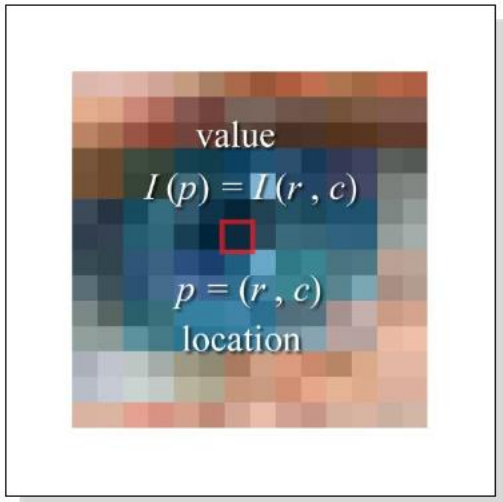
- $p = (r,c)$ is the pixel location indexed by row, r, and column, c.
- $I(p) = I(r,c)$ is the value of the pixel at location p.
- If $I(p)$ is a single number then I is monochrome.
- If $I(p)$ is a vector (ordered list of numbers) then I has multiple bands (*e.g.*, a color image).

Digital image in Matlab

Pixels







Pixel Location: $p = (r, c)$

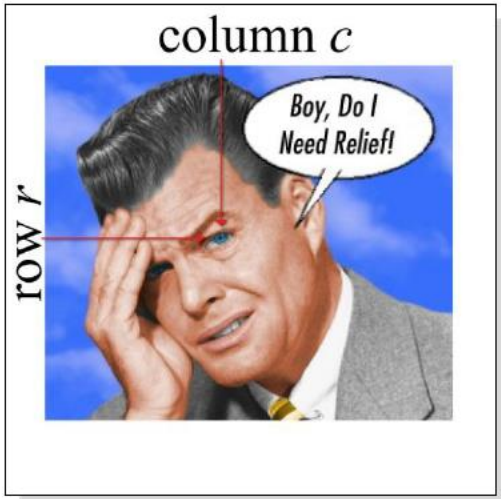
Pixel Value: $I(p) = I(r, c)$

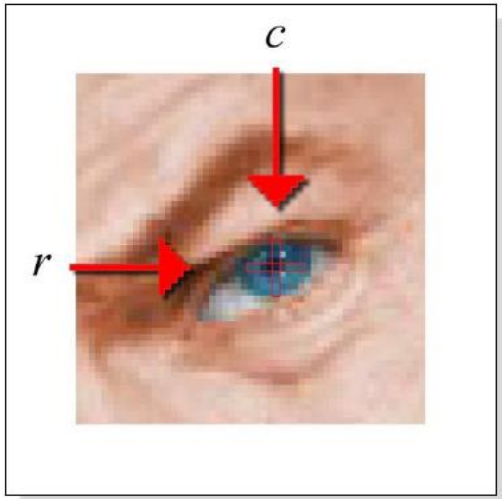
Pixel : $[p, I(p)]$

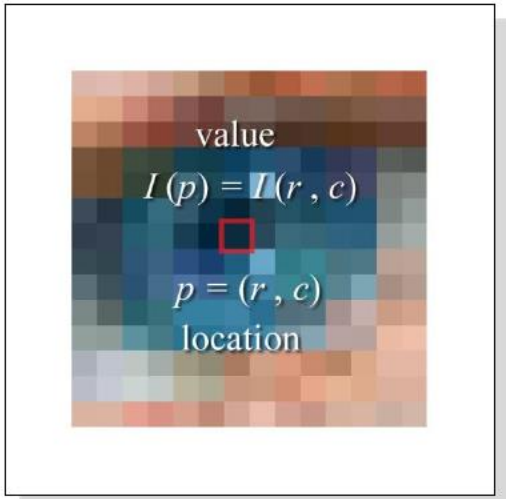
Digital image in Matlab

Pixels

Pixel : $[p, I(p)]$







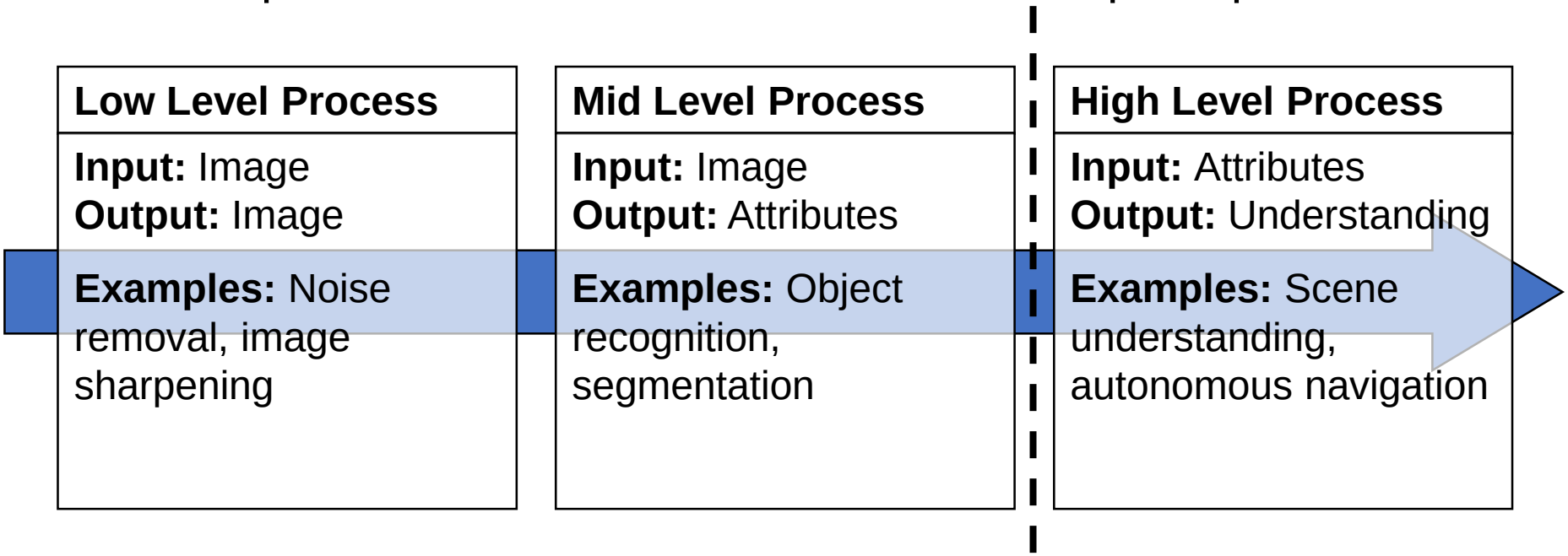
$$p = r, c$$
$$= \text{row \#, col \#}$$
$$= 272, 277$$

$$I(p) = \begin{bmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{bmatrix} = \begin{bmatrix} 12 \\ 43 \\ 61 \end{bmatrix}$$

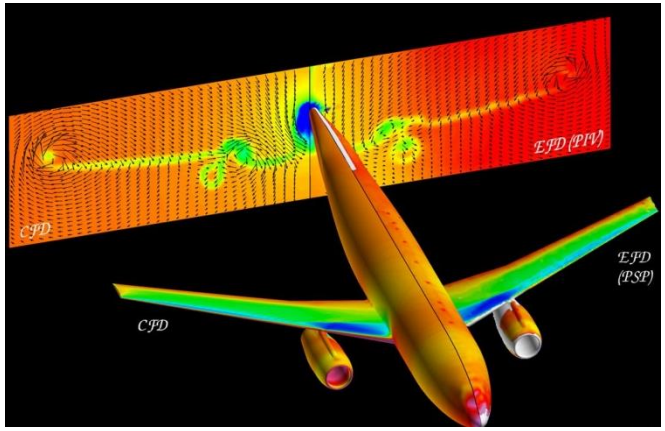
Digital image in Matlab

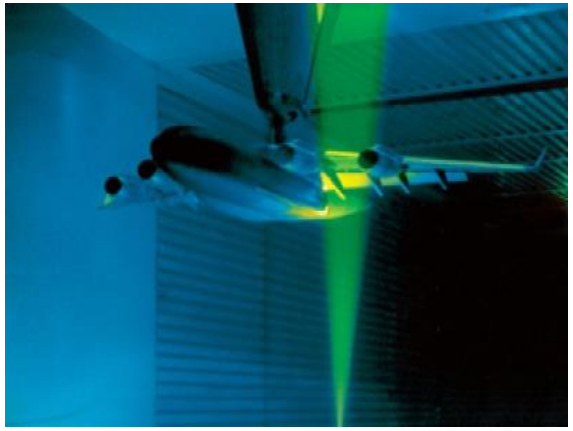
Digital image processing focuses on two major tasks

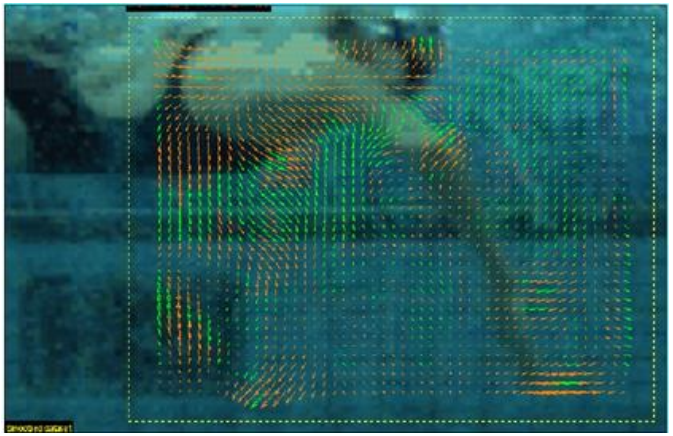
- Improvement of pictorial information for human interpretation
- Processing of image data for storage, transmission and representation for autonomous machine perception

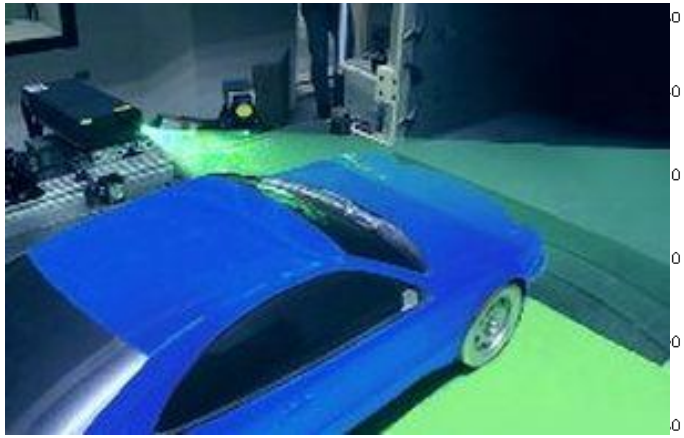


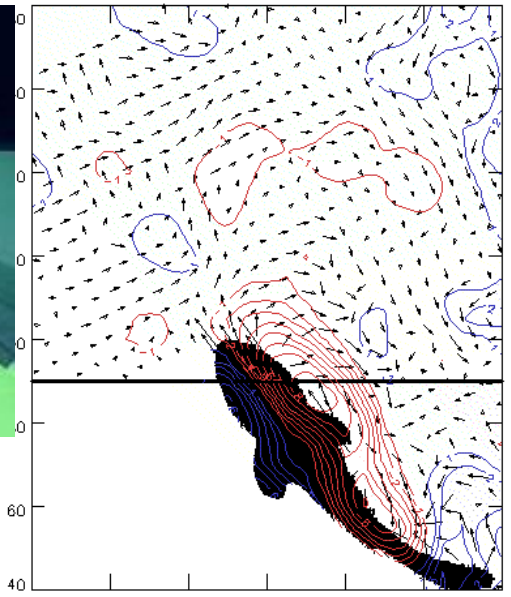
Planar/Volumetric Velocity measurement



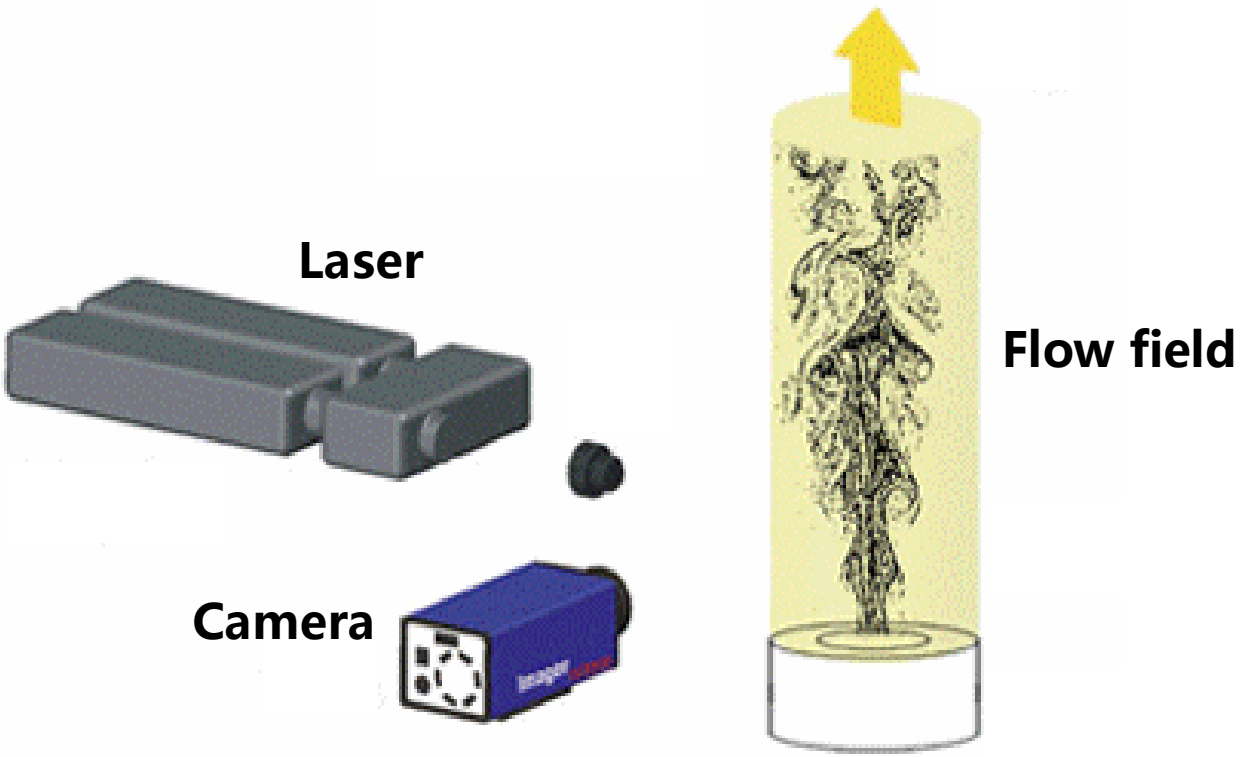






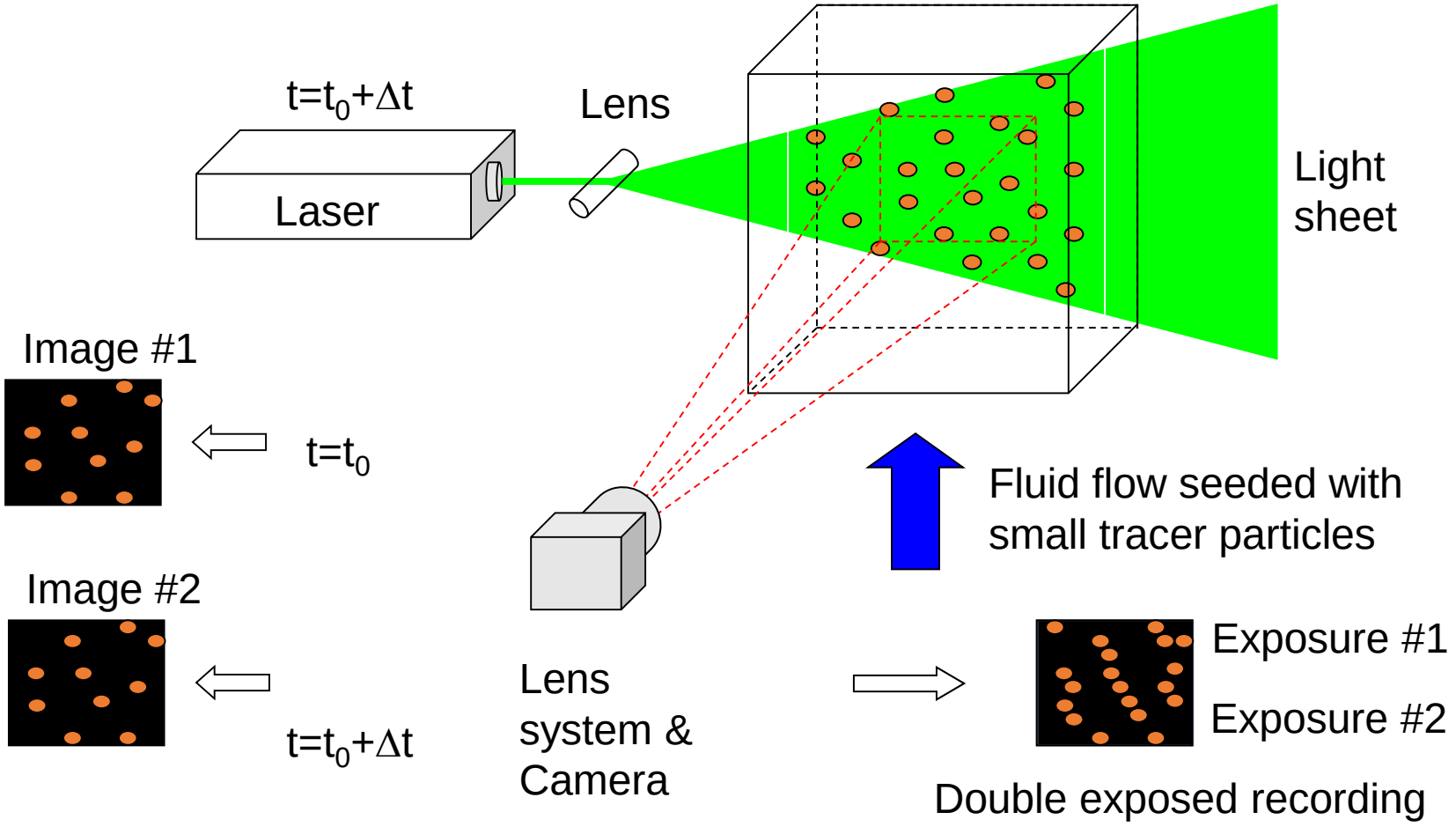


Particle Image Velocimetry



Particle Image Velocimetry

Basic Principle (2D PIV)



PIV image processing

2D cross correlation

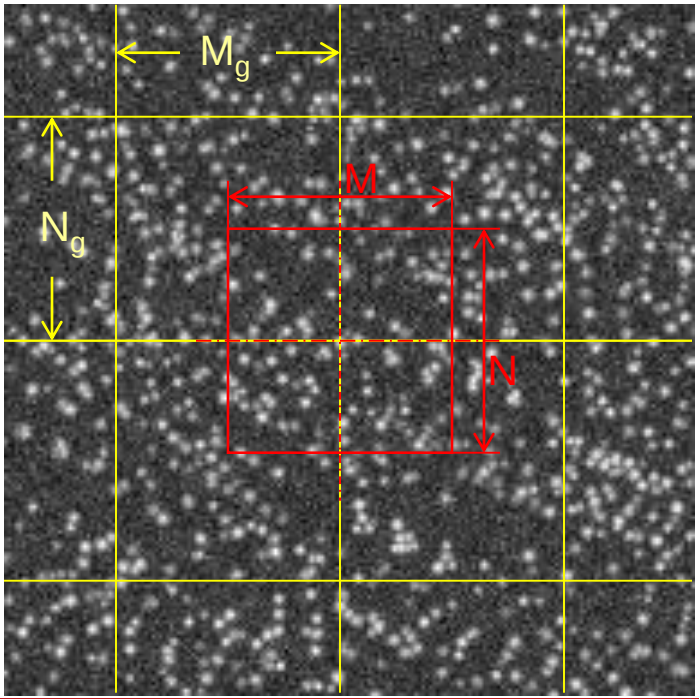
Basic principle

- Similarity of a particle image group between two frames

Procedure

- two frames are divided into small windows (e.g. 32X32 pixel)

PIV recording



Interrogation grid ($M_g \times N_g$)

Interrogation window ($M \times N$)

PIV image processing

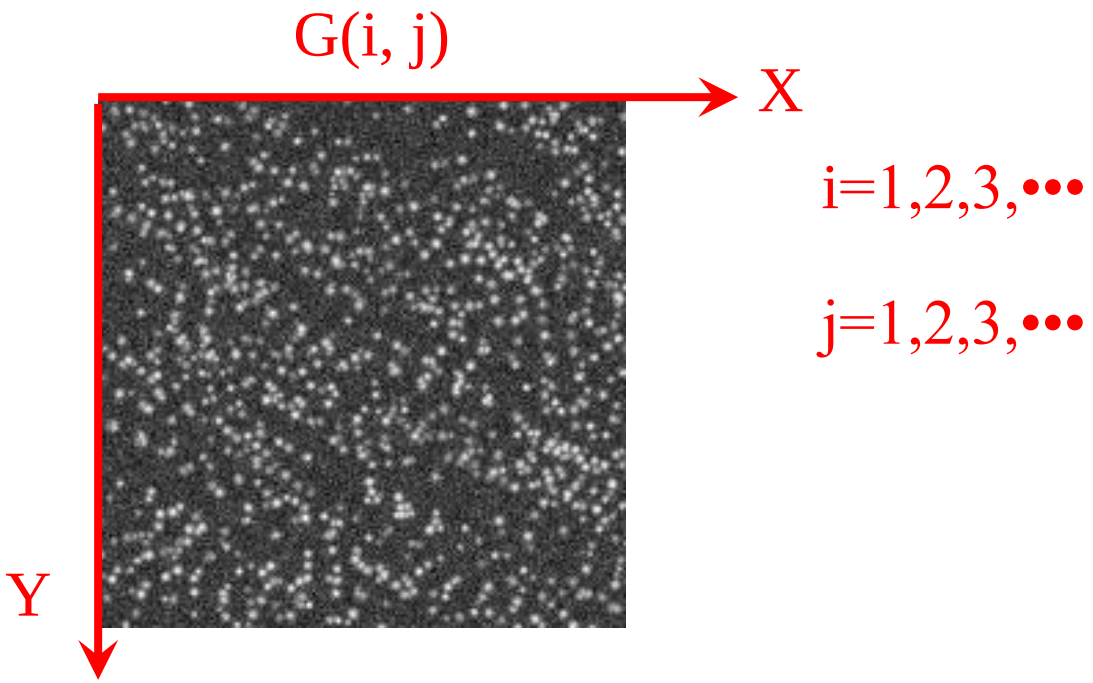
2D cross correlation

Basic principle

- Similarity of a particle image group between two frames

Procedure

- two frames are divided into small windows (e.g. 32X32 pixel)
- each interrogation windows are cross correlated



PIV image processing

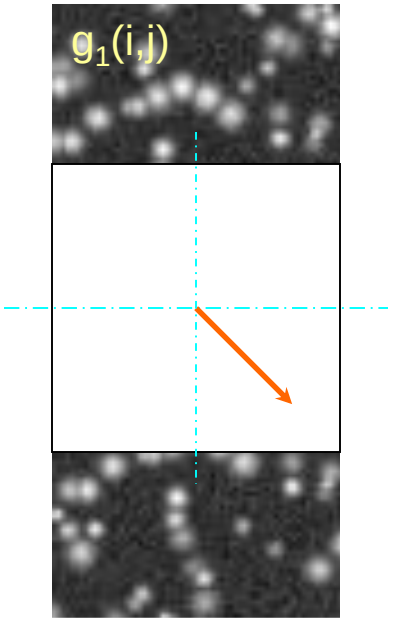
2D cross correlation

Basic principle

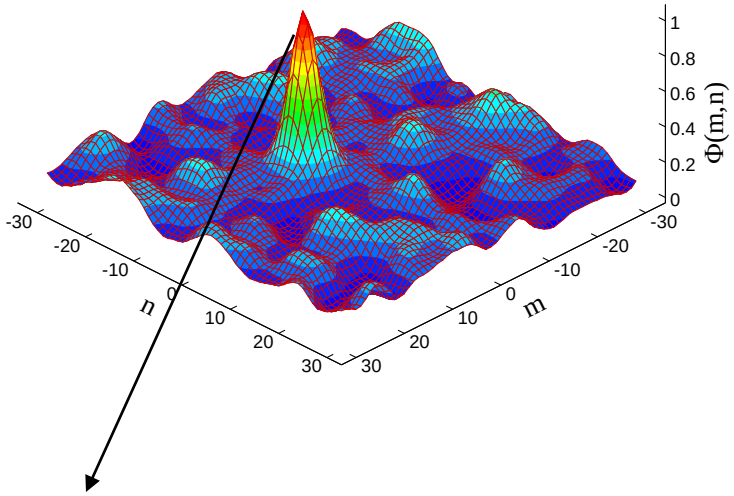
- Similarity of a particle image group between two frames

Procedure

- two frames are divided into small windows (e.g. 32X32 pixel)
- each interrogation window pair is cross correlated



$$\Phi(m,n)=\sum_{i=1}^{M}\sum_{j=1}^{N} g_1(i,j)\cdot g_2(i+m,\, j+n)$$



(m*=3, n*=5)

PIV image processing

2D cross correlation

Basic principle

- Similarity of a particle image group between two frames

Procedure

- two frames are divided into small windows (e.g. 32X32 pixel)
- each interrogation window pair is cross correlated
- a velocity vector is calculated from the interrogation window pair
- whole image is looped to obtain a velocity filed

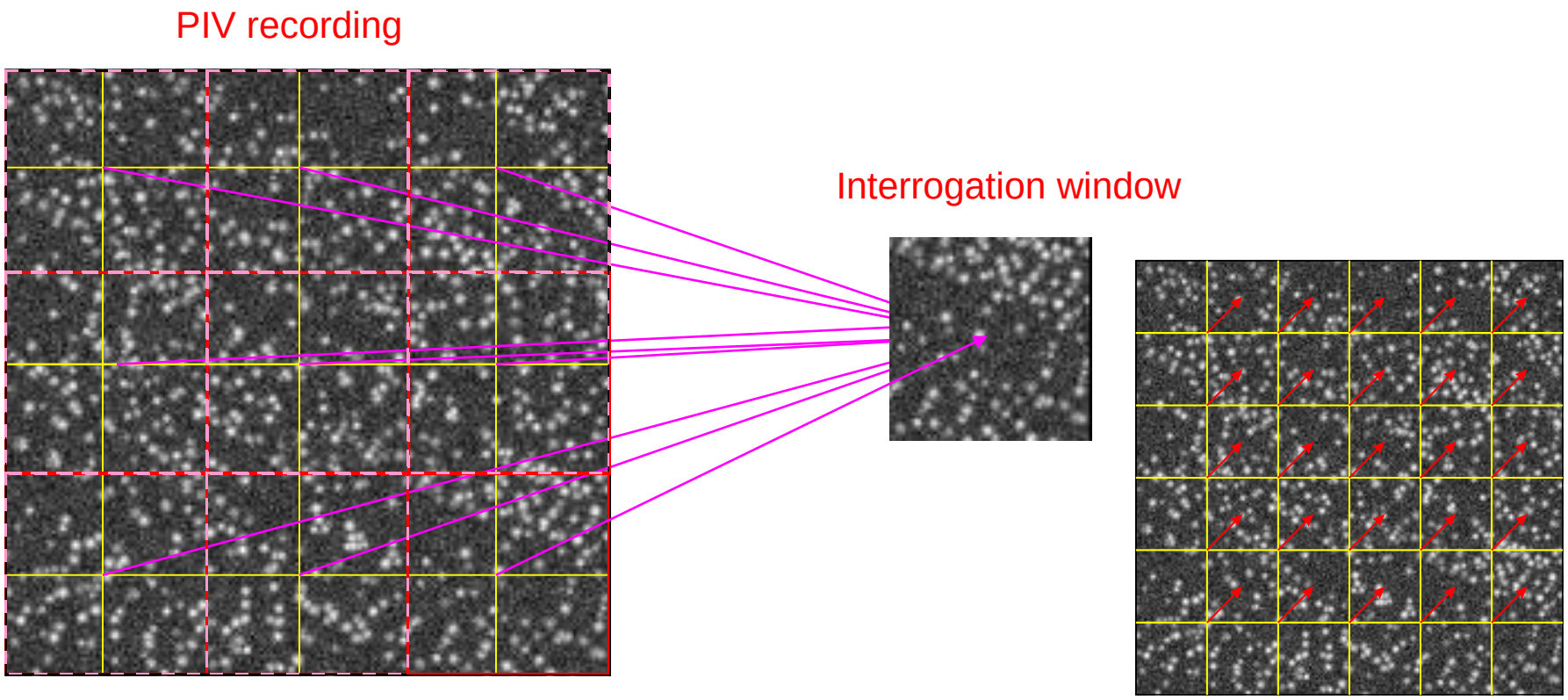
PIV image processing

2D cross correlation

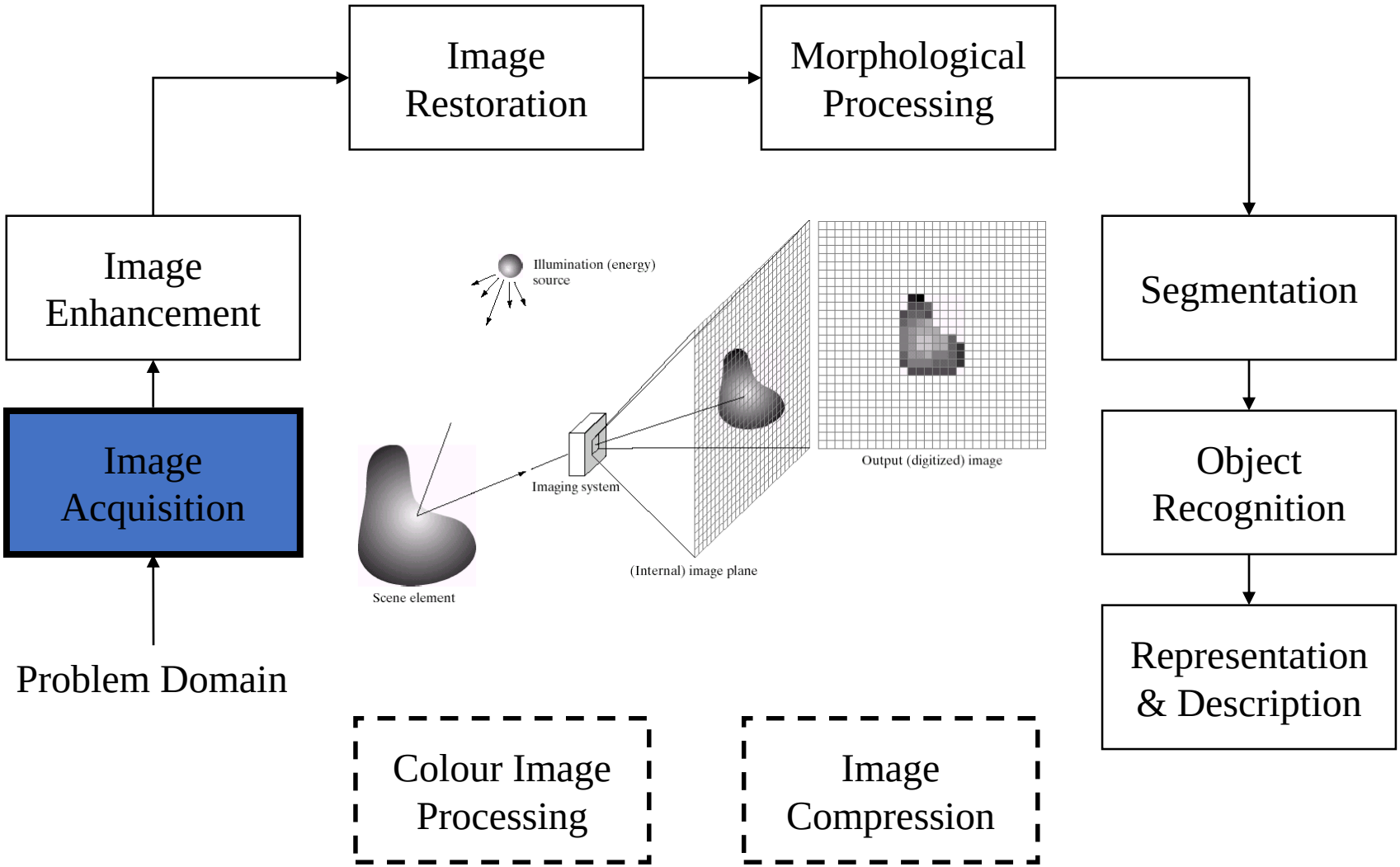
Basic principle

- Similarity of a particle image group between two frames

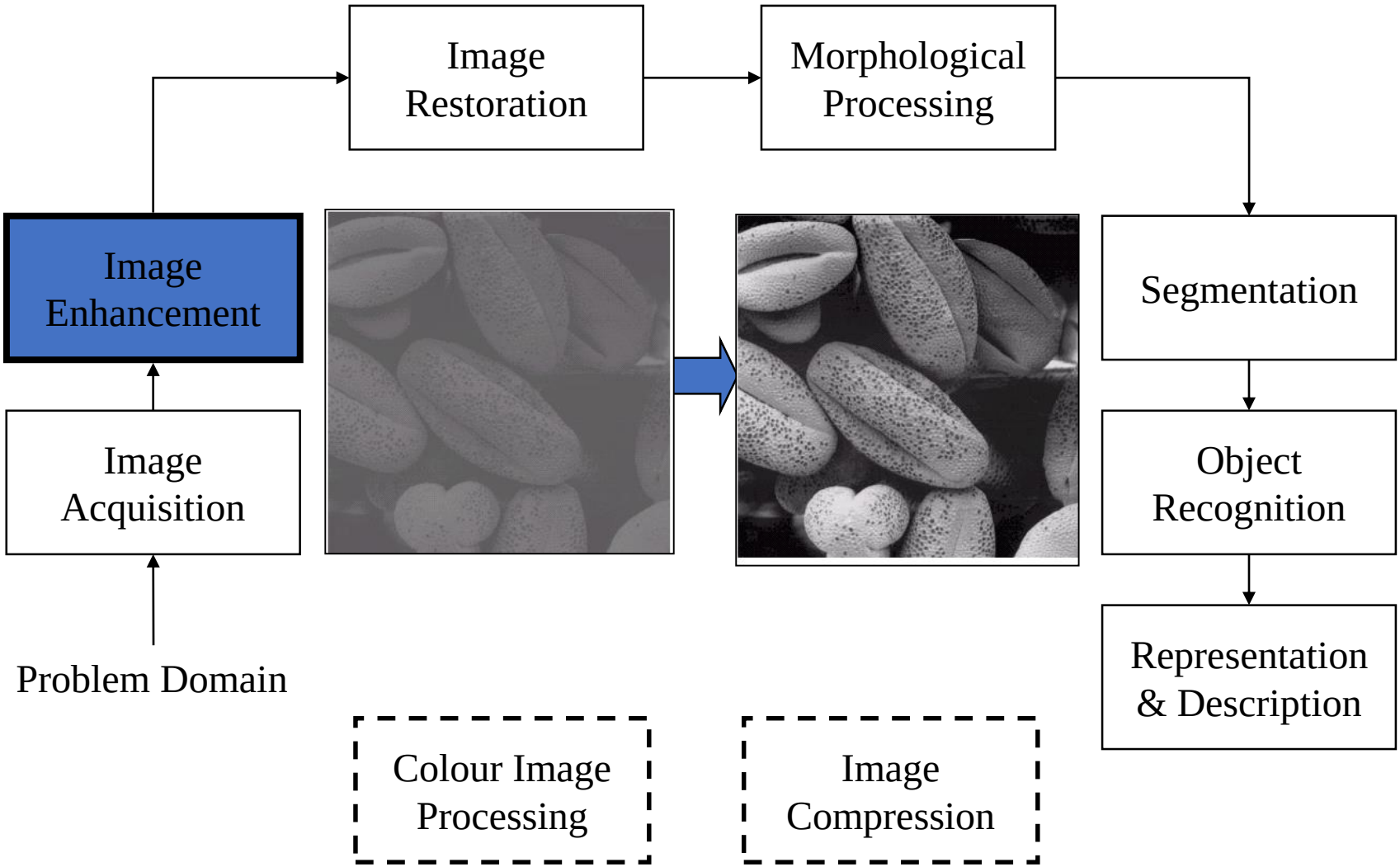
Procedure



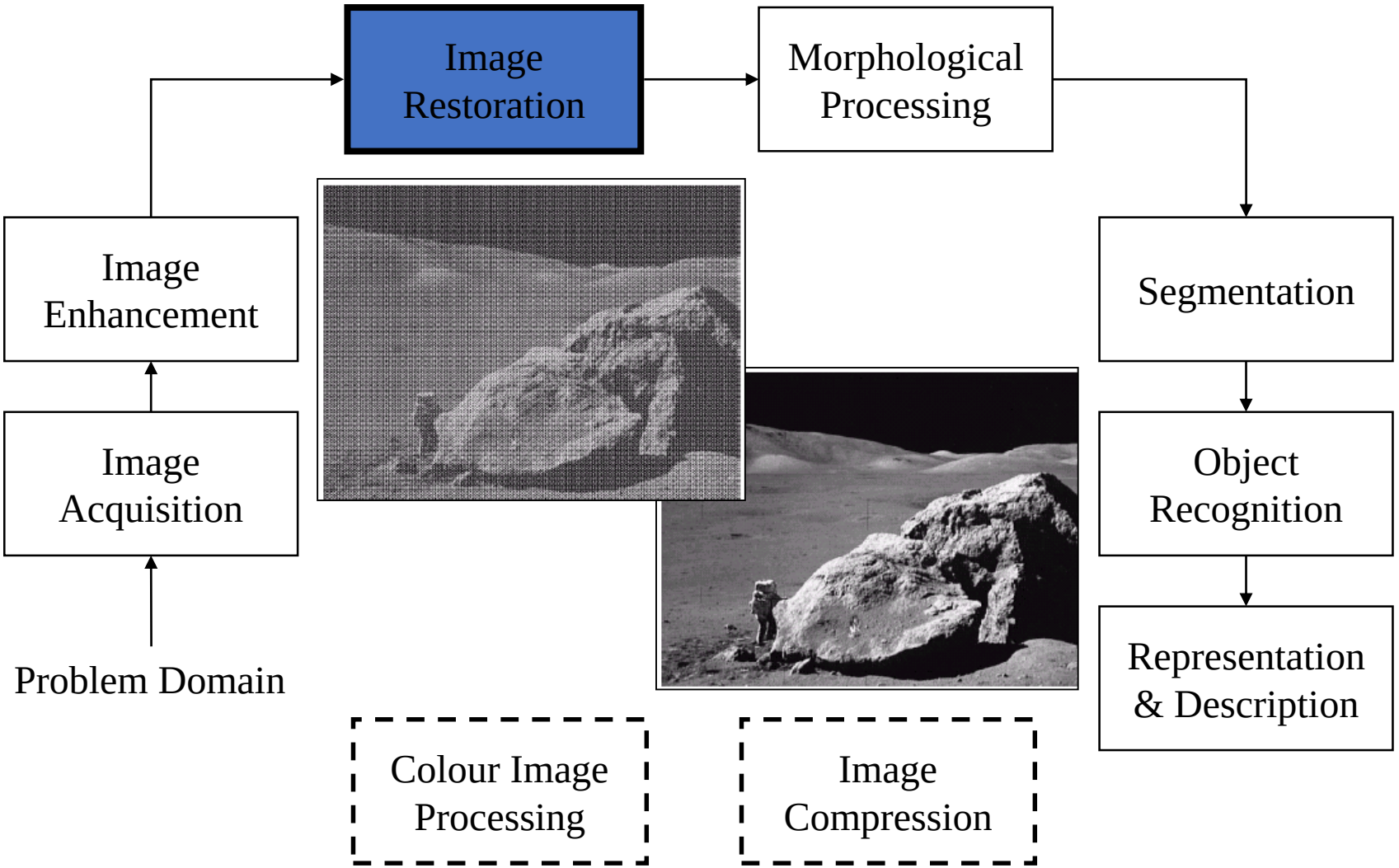
Key steps in image processing



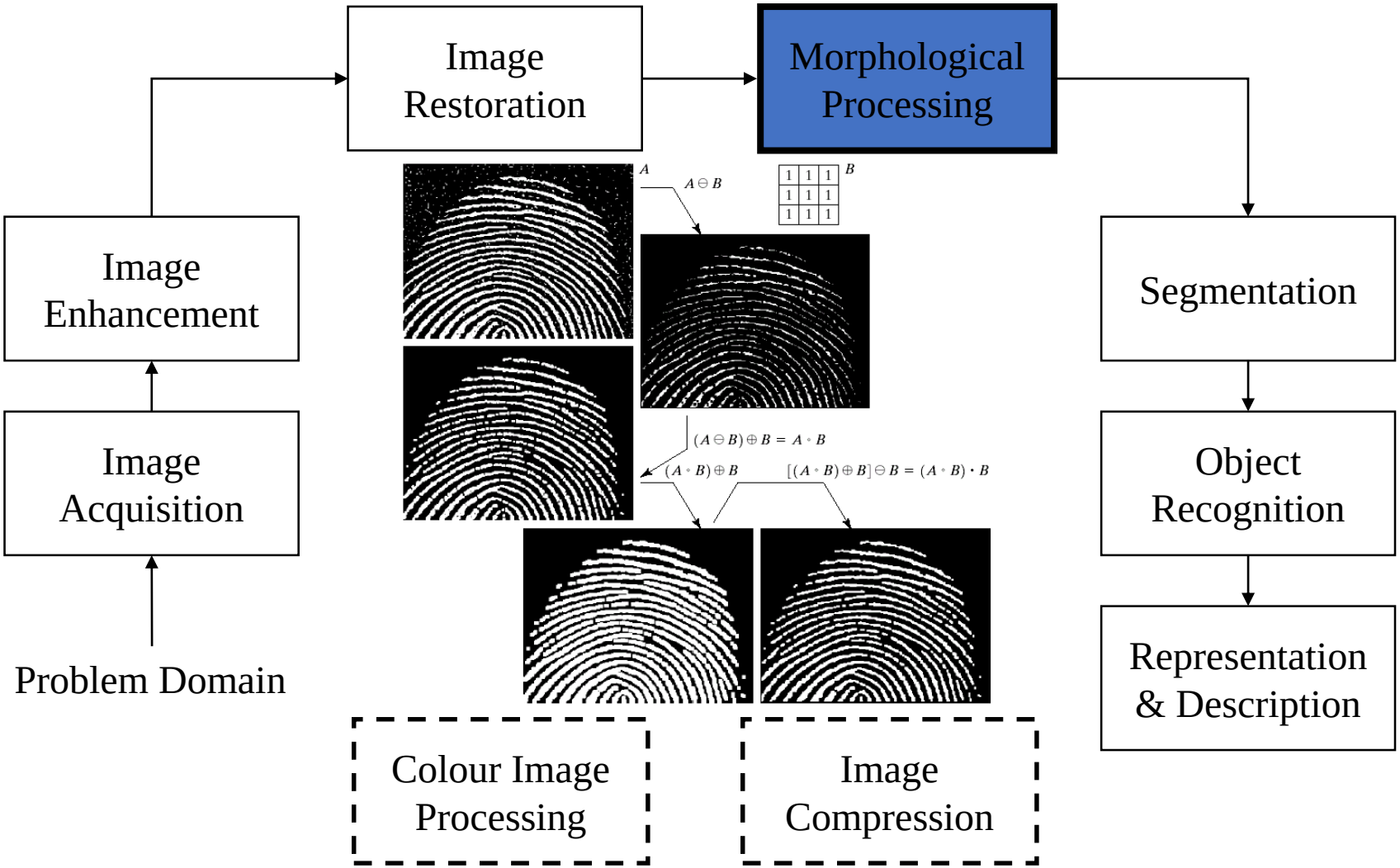
Key steps in image processing



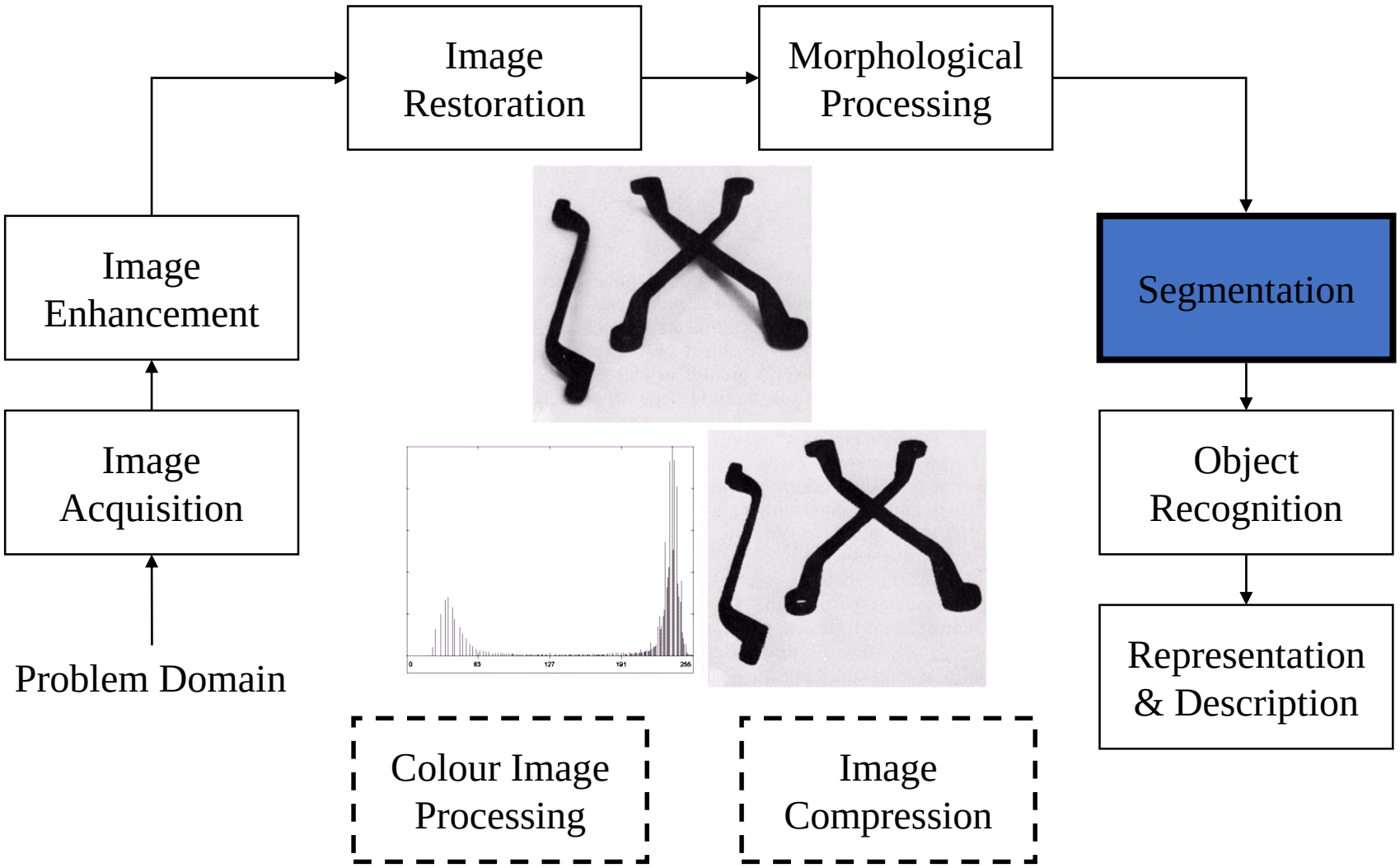
Key steps in image processing



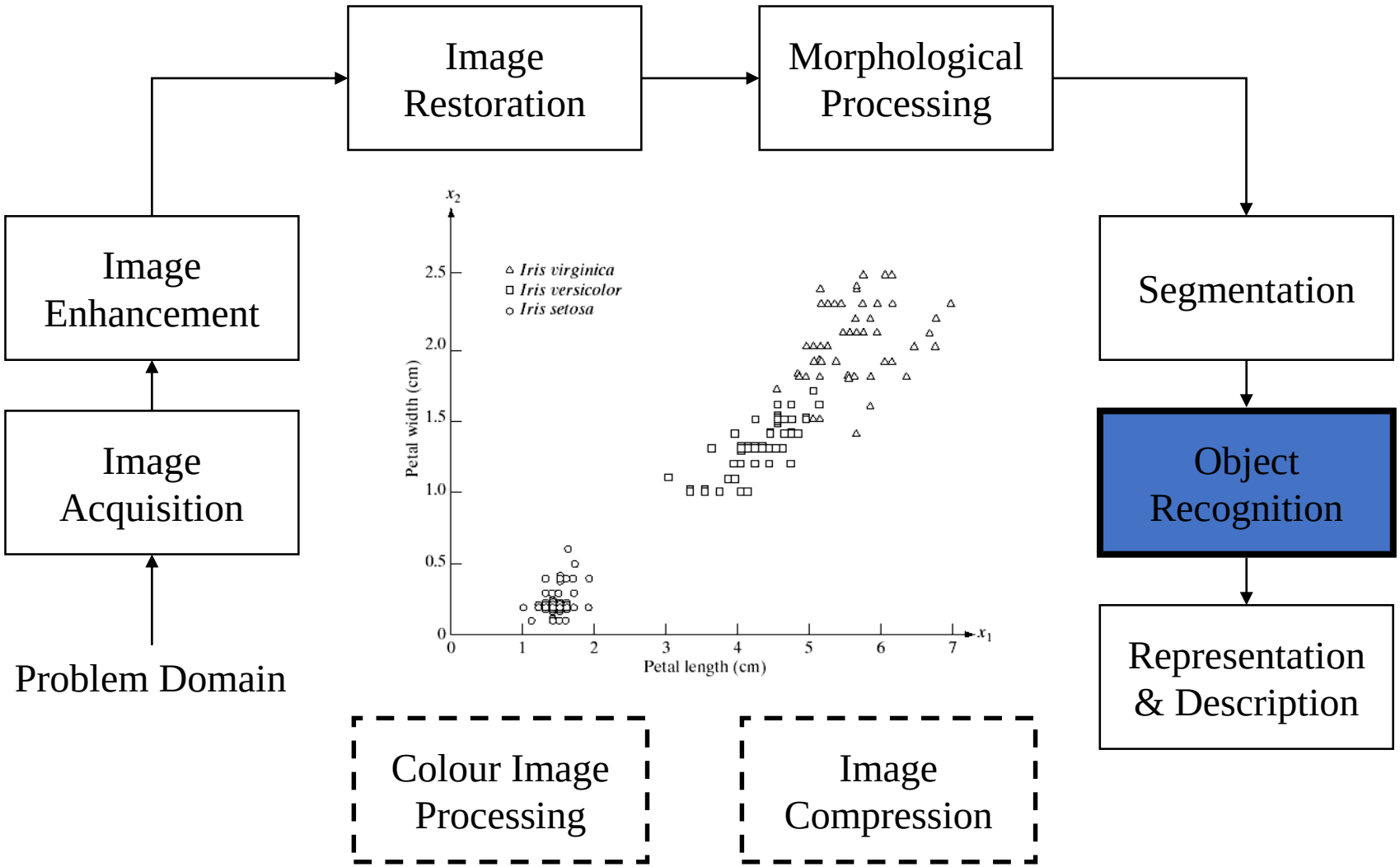
Key steps in image processing



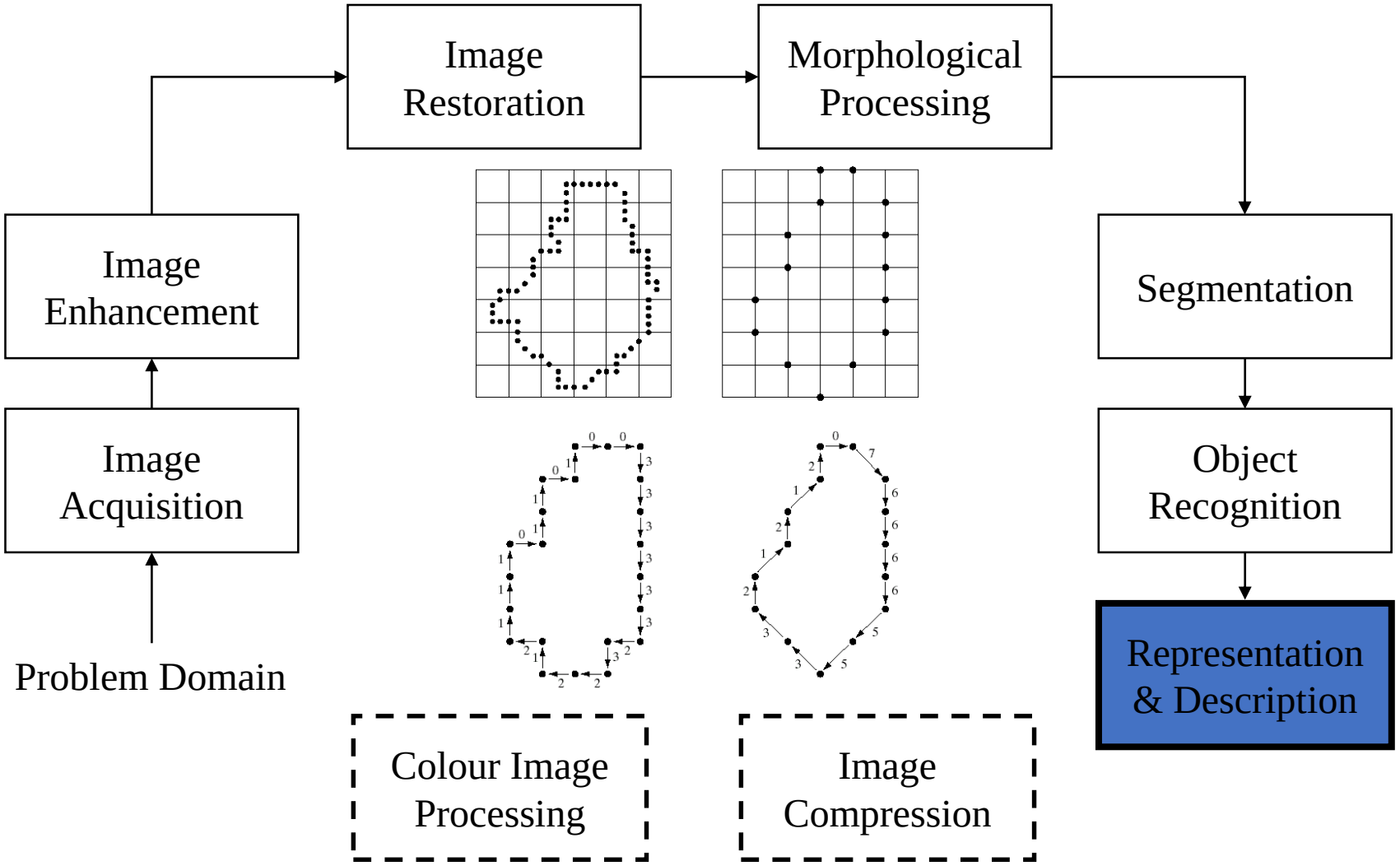
Key steps in image processing



Key steps in image processing



Key steps in image processing



Key steps in image processing

Image Restoration

Morphological Processing

Segmentation

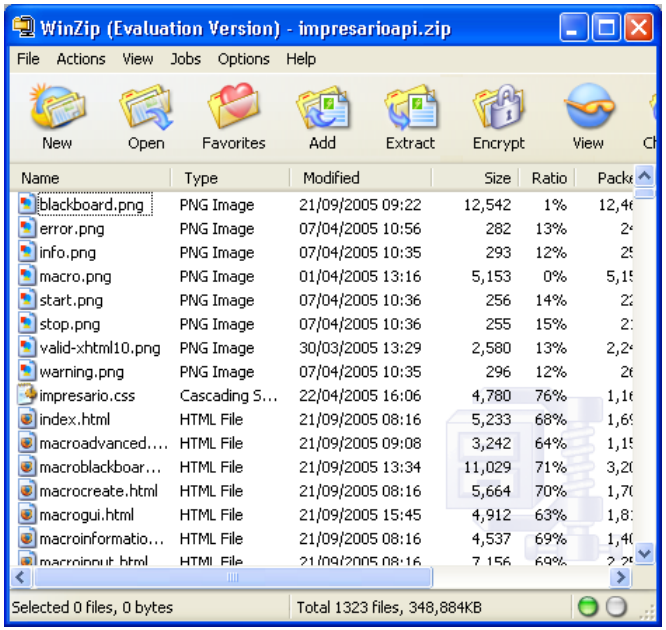
Object Recognition

Representation & Description

Image Enhancement

Image Acquisition

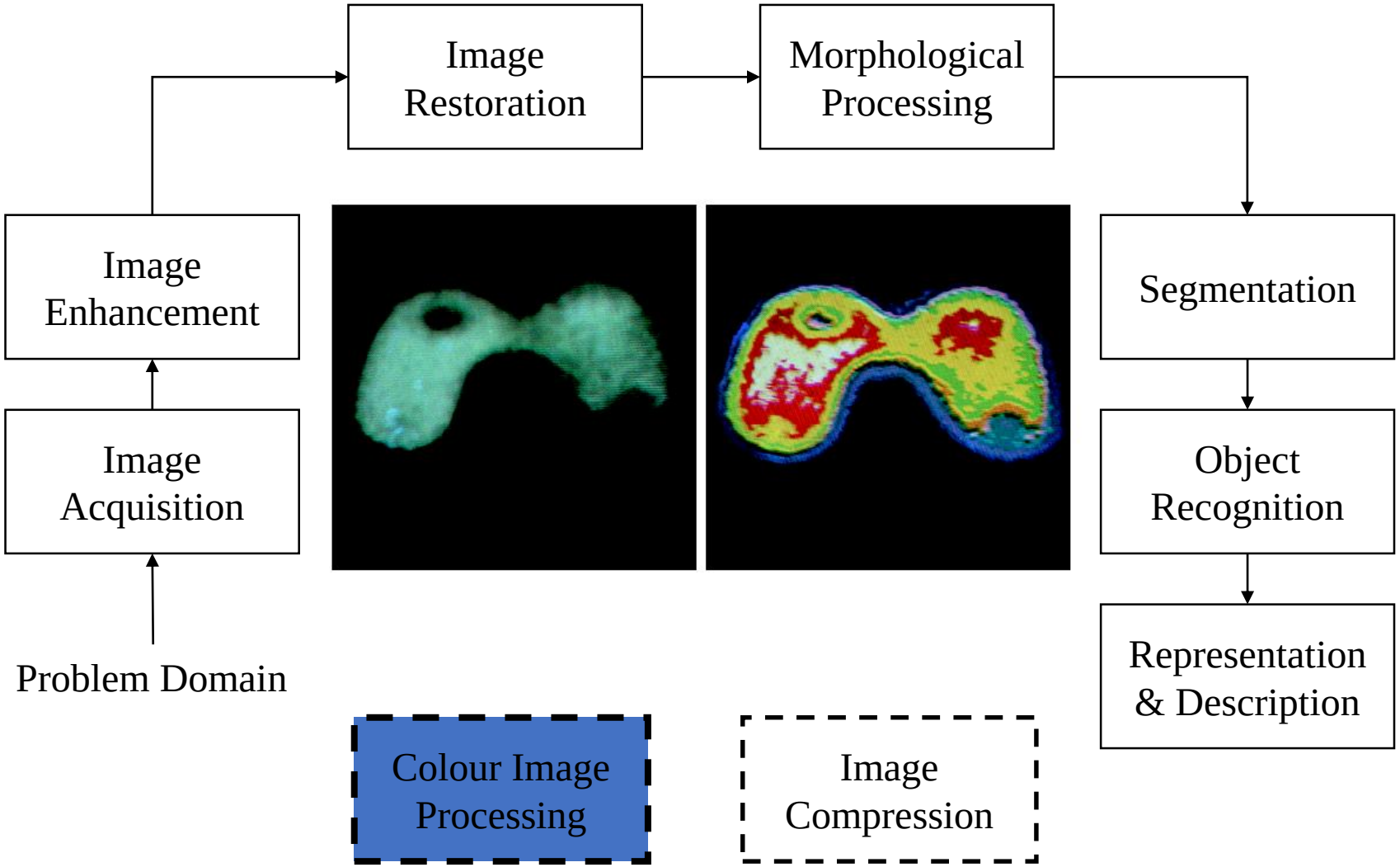
Problem Domain



Colour Image Processing

Image Compression

Key steps in image processing



Read/Display/Write an image

- MATLAB can import/export several image formats:
  - BMP (Microsoft Windows Bitmap)
  - GIF (Graphics Interchange Files)
  - HDF (Hierarchical Data Format)
  - JPEG (Joint Photographic Experts Group)
  - PCX (Paintbrush)
  - PNG (Portable Network Graphics)
  - TIFF (Tagged Image File Format)
  - XWD (X Window Dump)
  - raw-data and other types of image data

- Data types in MATLAB
  - Double (64-bit double-precision floating point)
  - Single (32-bit single-precision floating point)
  - Int32 (32-bit signed integer)
  - Int16 (16-bit signed integer)
  - Int8 (8-bit signed integer)
  - Uint32 (32-bit unsigned integer)
  - Uint16 (16-bit unsigned integer)
  - Uint8 (8-bit unsigned integer)

Display/Write an image

A = imread(filename, fmt)

imwrite(A,filename,fmt)

imshow(A)
imshow(A,[low high])
imshow(RGB)
imshow(BW)
imshow(X,map)
imshow(filename)

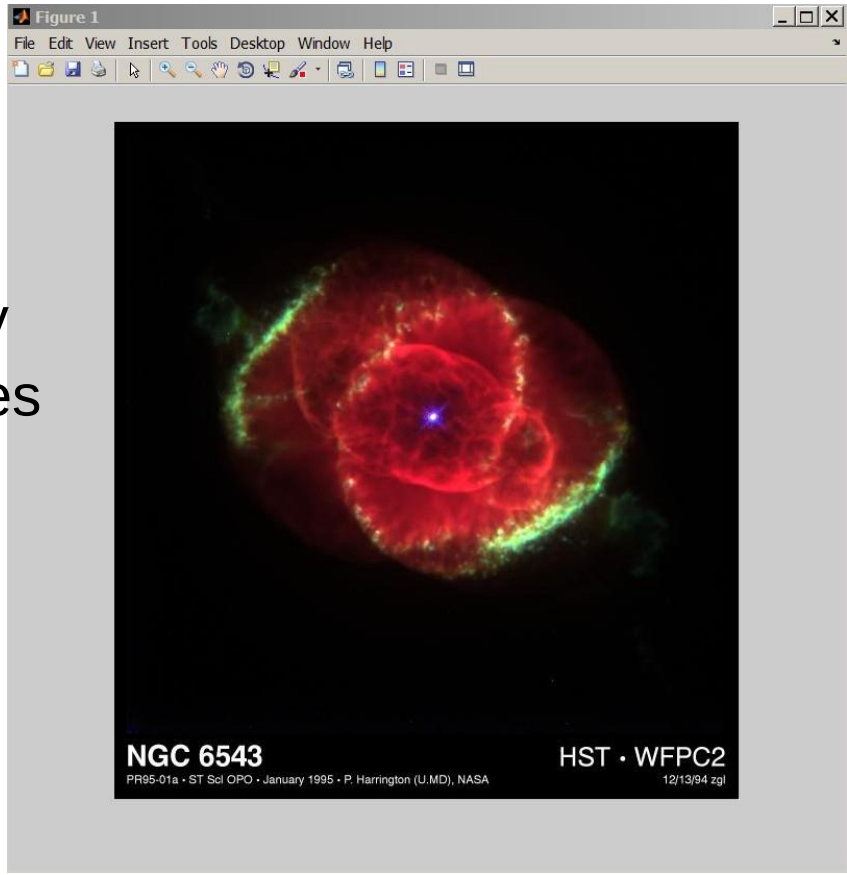
imagesc(A)

imtool(A)

image(A)

RGB image

```
rgb = imread('ngc6543a.jpg');
image(rgb);
```



Intensity of each pixel is represented by value in the red, green and blue matrices

```
>> I(1,1,:)
ans(:,:,1) =
   135
ans(:,:,2) =
    97
ans(:,:,3) =
    33
```

Red

Green

Blue

Images where the pixel value in the image represents the intensity of the pixel are called intensity images.

Index

An
col

The
inte

loa
ima
col

>>
ans

>>
ans
0

Try
An

Converting between image types

- `gray2ind` - intensity image to index image
- `im2bw` - image to binary
- `im2double` - image to double precision
- `im2uint8` - image to 8-bit unsigned integers
- `im2uint16` - image to 16-bit unsigned integers
- `ind2gray` - indexed image to intensity image
- `mat2gray` - matrix to intensity image
- `rgb2gray` - RGB image to grayscale
- `rgb2ind` - RGB image to indexed image

Converting between image types

```
f =
   -0.5000    0.5000
    0.7500    1.5000
```

```
g=im2uint8(f)
ans =
    0  128
  191  255
```

```
im2double(g)
ans =
         0    0.5020
    0.7490    1.0000
```

Try:

```
I = imread('cameraman.tif');
[X, map] = gray2ind(I, 16);
imshow(X, map);
```

```
I = imread('cameraman.tif');
[X, map] = gray2ind(I, 256);
imshow(X, map);
```

```
I = imread('cameraman.tif');
I=im2bw(I,0.5);
imshow(I);
```

Read a Truecolor Image into Matlab

Read a Truecolor Image into Matlab

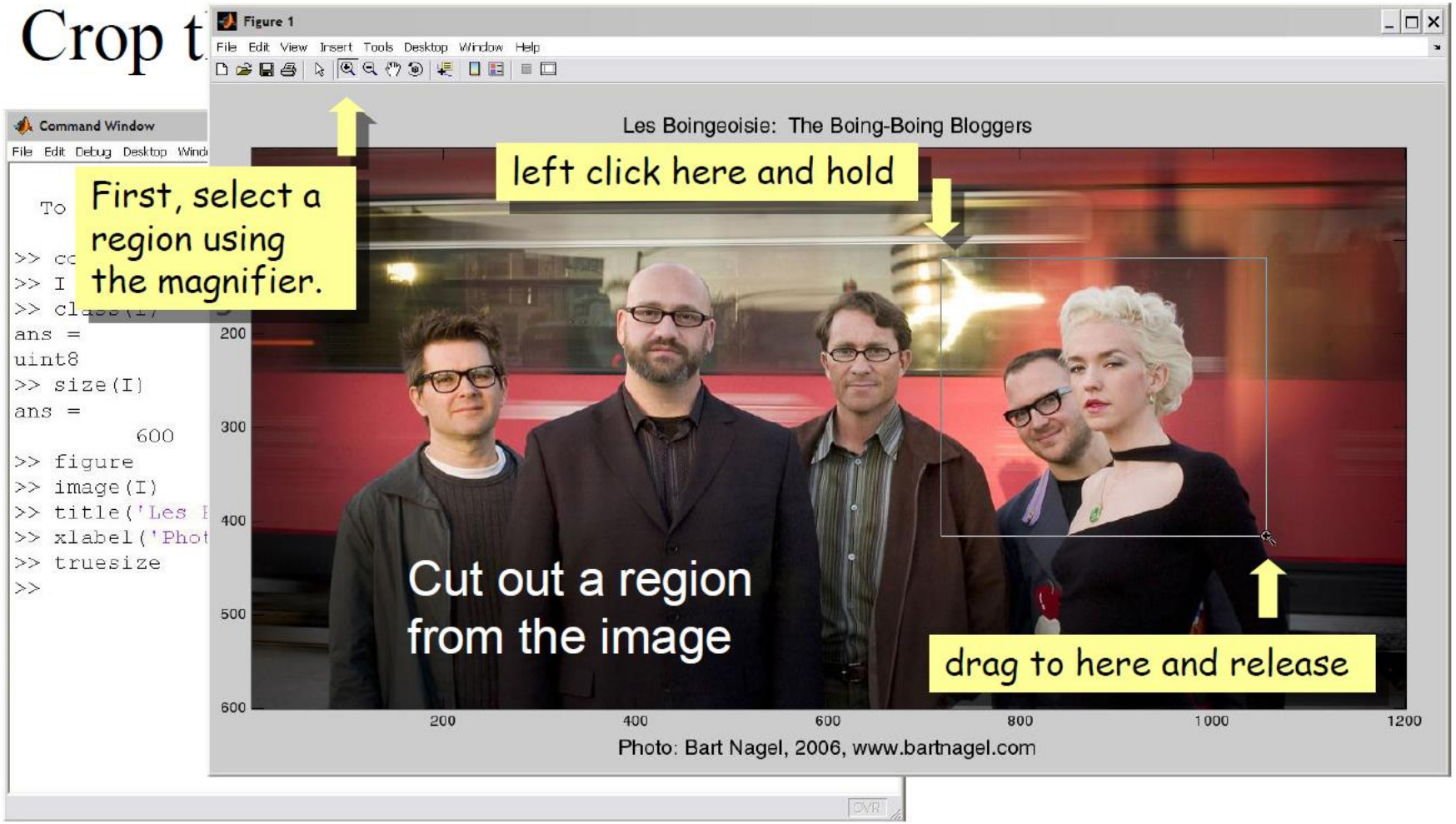
Command Window

File Edit Debug Desktop Window Help

```
  To get started, select MATLAB Help or Demos from the Help menu.

>> cd 'E:\images\Animals\People\Famous'
>> I = imread('Les_Boingeoisie.jpg','jpg');
>> class(I)
ans =
uint8
>> size(I)
ans =
         600        1200           3
>> figure
>> image(I)
>> title('Les Boingeoisie:  The Boing-Boing Bloggers')
>> xlabel('Photo: Bart Nagel, 2006, www.bartnagel.com')
>>
```

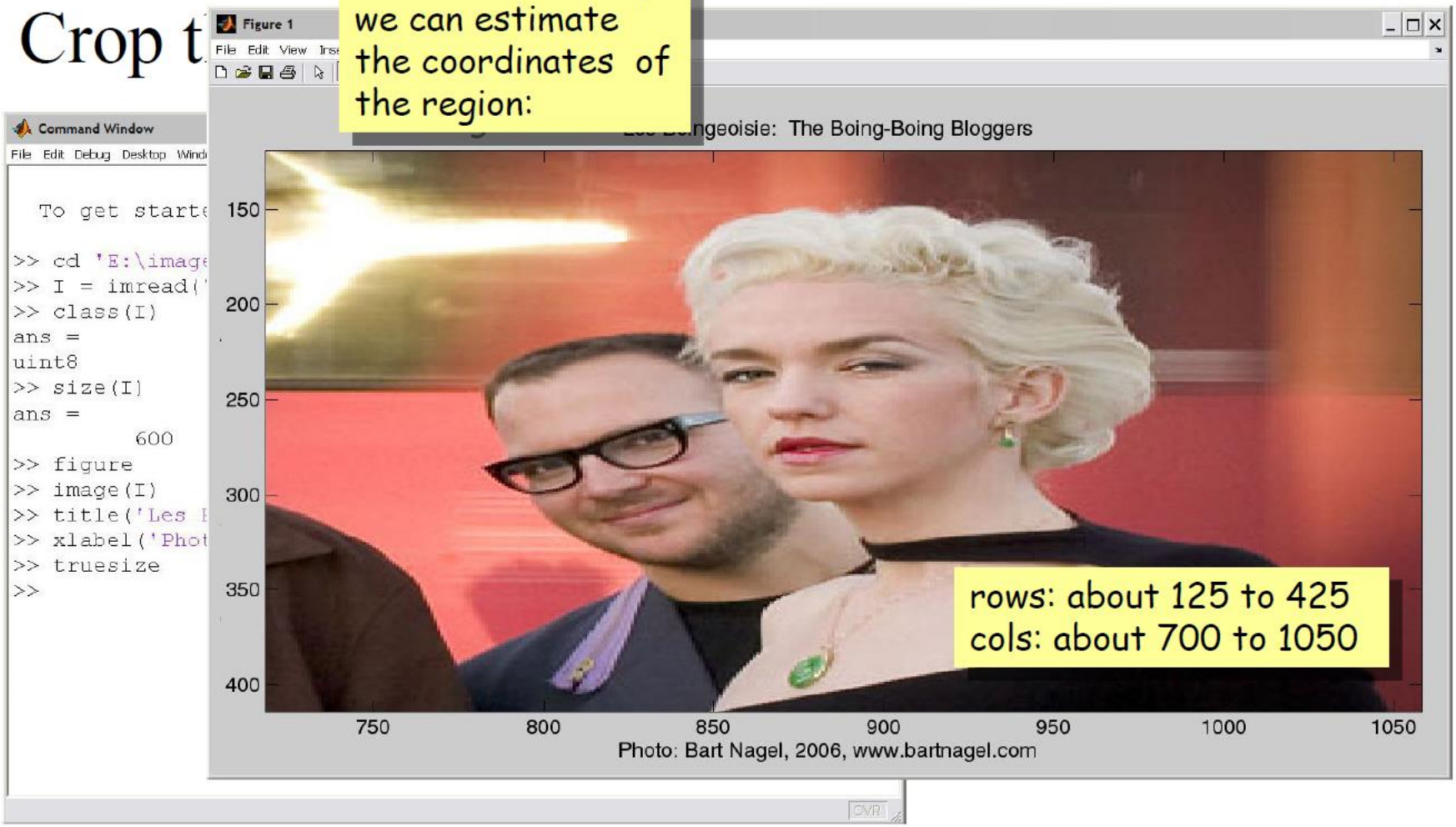
Read a Truecolor Image into Matlab



Crop an image

Crop t

From this close-up we can estimate the coordinates of the region:



Crop an image

Crop t

Figure 1

File Edit View Insert Tools Desktop Window Help

Command Window

File Edit Debug Desktop Window Help

```
  To get started, select MATLAB Help or Demos from the Help menu.

>> cd 'E:\images\Animals\People\Famous'
>> I = imread('Les_Boingeoisie.jpg','jpg');
>> class(I)
ans =
uint8
>> size(I)
ans =
         600        1200           3
>> figure
>> image(I)
>> title('Les Boingeoisie:  The Boing-Boing Bloggers')
>> xlabel('Photo: Bart Nagel, 2006, www.bartnagel.com')
>> truesize
>> J = I(125:425,700:1050,:);
>> figure
>> image(J)
>> truesize
>>
```

Here it is:

Figure 2

File Edit View Insert Tools Desktop Window Help

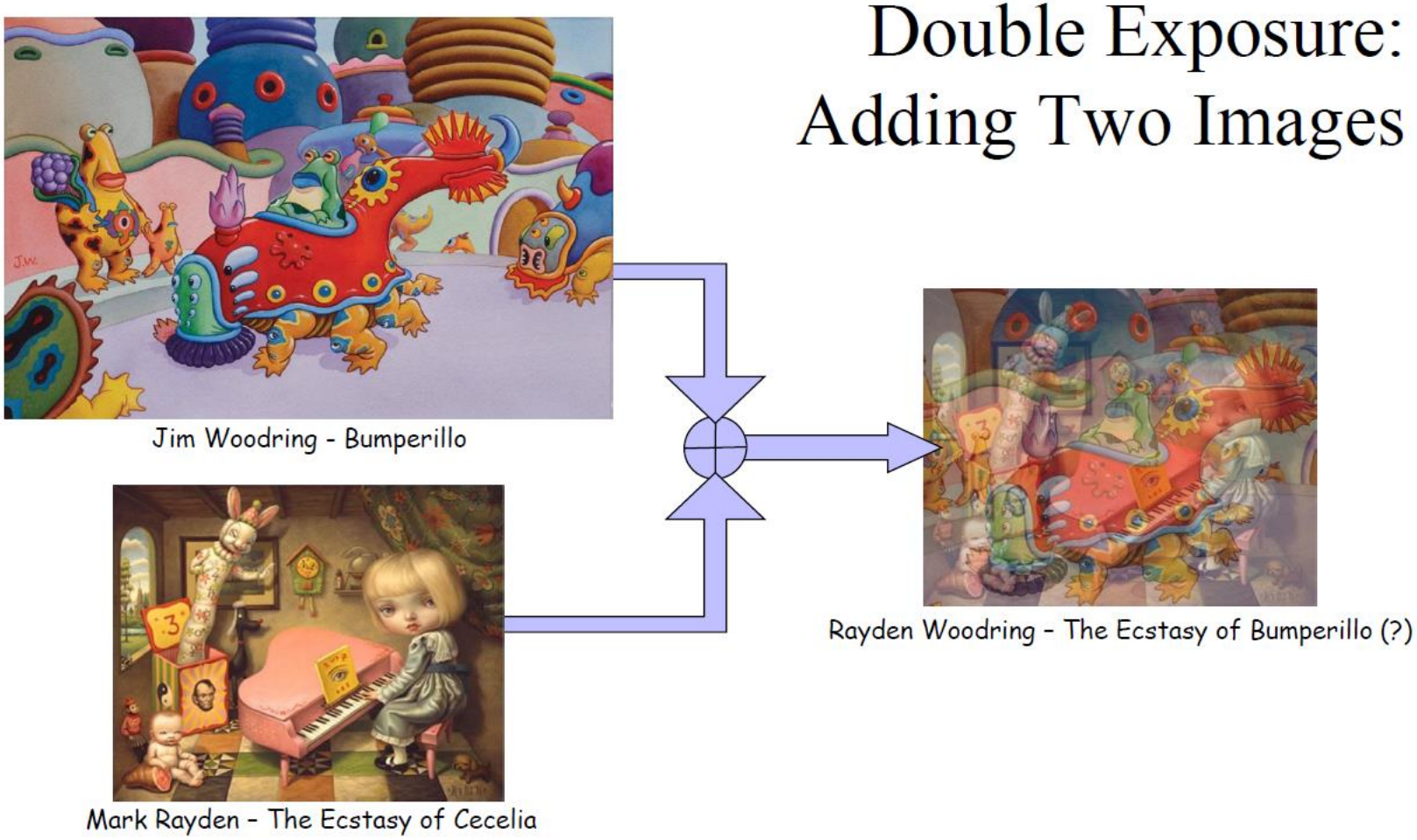
oing Bloggers

950

bartnagel.com

Now close the other image

Add two images



Add two images

Example Matlab Code

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```

Add two images

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```

Example Matlab Code

Cut a section out of the middle of the larger image the same size as the smaller image.

Add two images

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```

Example Matlab Code

Note that the images are averaged, pixelwise.

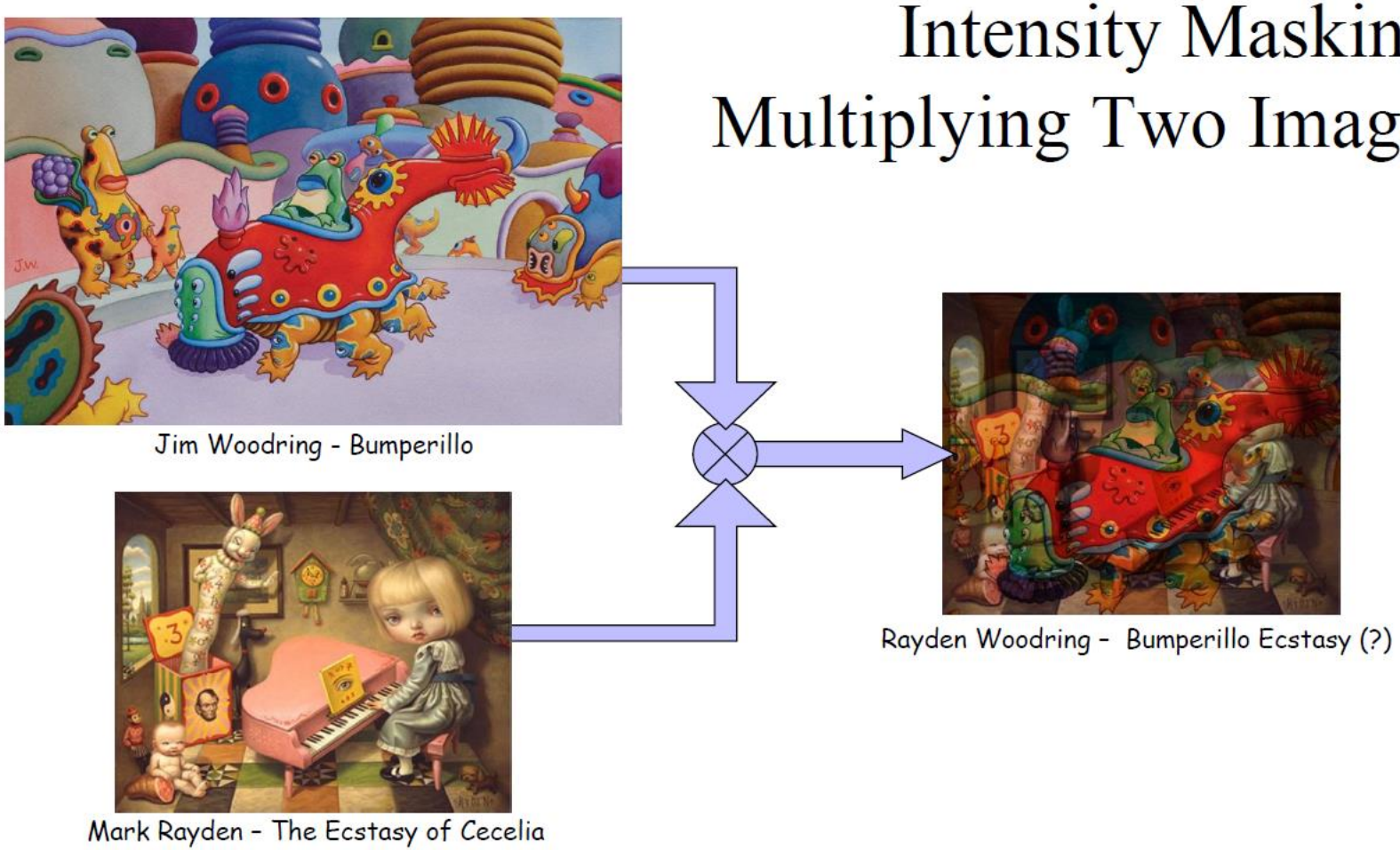
Add two images

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> title('The Ecstasy of Cecelia')
>> xlabel('Mark Ryden')
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8(double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Ecstasy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
```

Example Matlab Code

Note the data class conversions.

Multiply two images



Multiply two images

Example Matlab Code

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Extacsy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
>> JWtimesMR = double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:)).*double(MR);
>> M = max(JWtimesMR(:));
>> m = min(JWtimesMR(:));
>> JWtimesMR = uint8(255*(double(JWtimesMR)-m)/(M-m));
>> figure
>> image(JWtimesMR)
>> truesize
>> title('EcstasyBumperillo')
```

Multiply two images

```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);
>> figure
>> image(JWplusMR)
>> truesize
>> title('The Extacsy of Bumperillo')
>> xlabel('Jim Woodring + Mark Ryden')
>> JWtimesMR = double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:)).*double(MR);
>> M = max(JWtimesMR(:));
>> m = min(JWtimesMR(:));
>> JWtimesMR = uint8(255*(double(JWtimesMR)-m)/(M-m));
>> figure
>> image(JWtimesMR)
>> truesize
>> title('EcstasyBumperillo')
```

Example Matlab Code

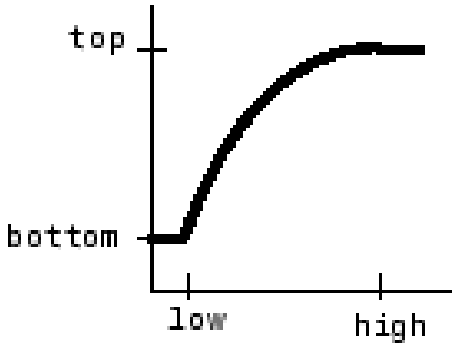
Note that the images are multiplied, pixelwise.

Note how the image intensities are scaled back into the range 0-255.

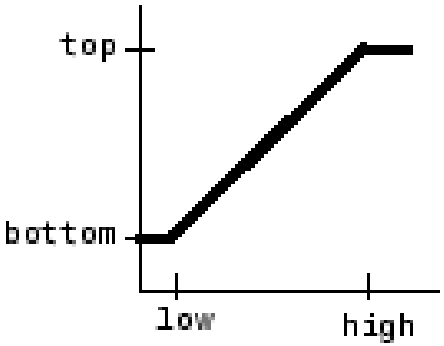
Intensity transformation

J = imadjust(I,[low_in; high_in],[low_out; high_out],gamma)

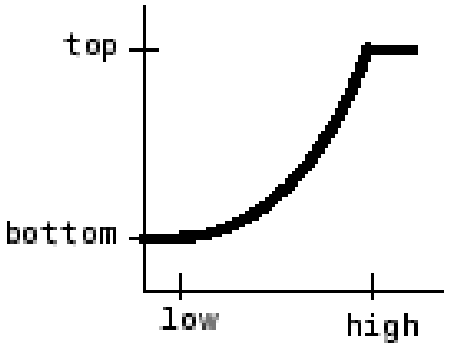
$\gamma < 1$



$\gamma = 1$



$\gamma > 1$



Intensity transformation

J = imadjust(I,[low_in; high_in],[low_out; high_out],gamma)

Try
I = imread('pout.tif');
J = imadjust(I,[0,1],[1,0]);
imshow(I)
Figure
imshow(J)

I = imread('pout.tif');
J = imadjust(I);
imshow(I)
Figure
imshow(J)

I = imread('pout.tif');
g=imcomplement(I);
imshow(g)

RGB1= imadjust(WallaceAndGromit,[],[],2);
imshow(RGB1)
RGB2 = imadjust(WallaceAndGromit,[],[],0.5);
figure;
imshow(RGB2)

The Histogram of a Grayscale Image

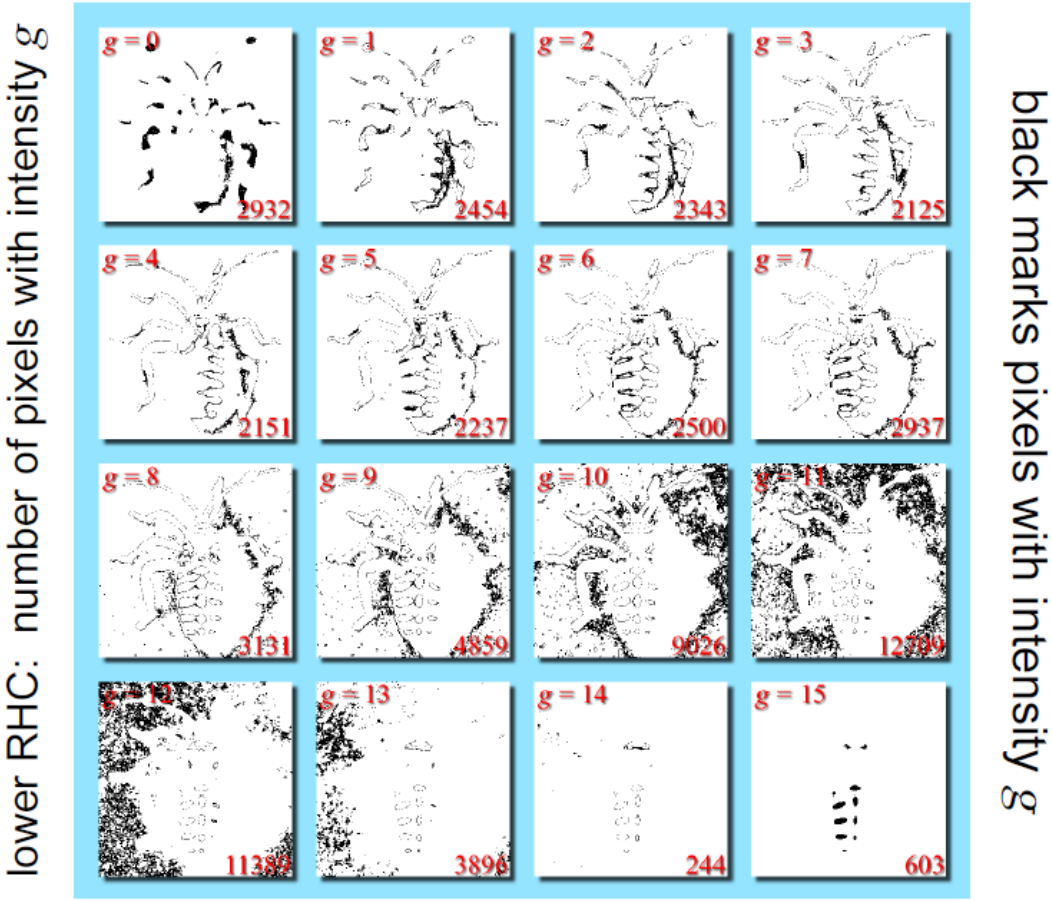
- Let **I** be a 1-band (grayscale) image.
- **I**(r,c) is an 8-bit integer between 0 and 255.
- Histogram, $h_{\mathbf{I}}$, of **I**:
  - a 256-element array, $h_{\mathbf{I}}$
  - $h_{\mathbf{I}}(g)$, for $g = 1, 2, 3, \ldots, 256$, is an integer
  - $h_{\mathbf{I}}(g)$ = number of pixels in **I** that have value g-1.

h=imhist(I)

Intensity transformation

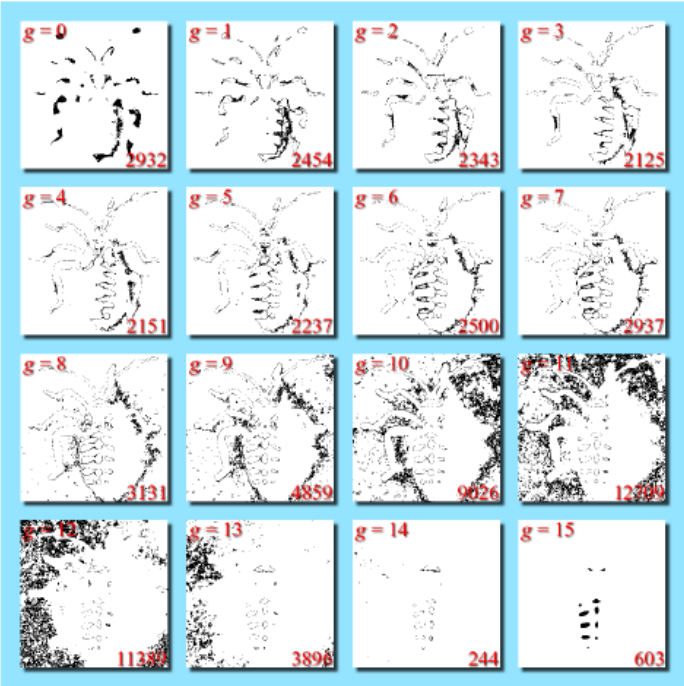
The Histogram of a Grayscale Image

16-level (4-bit) image

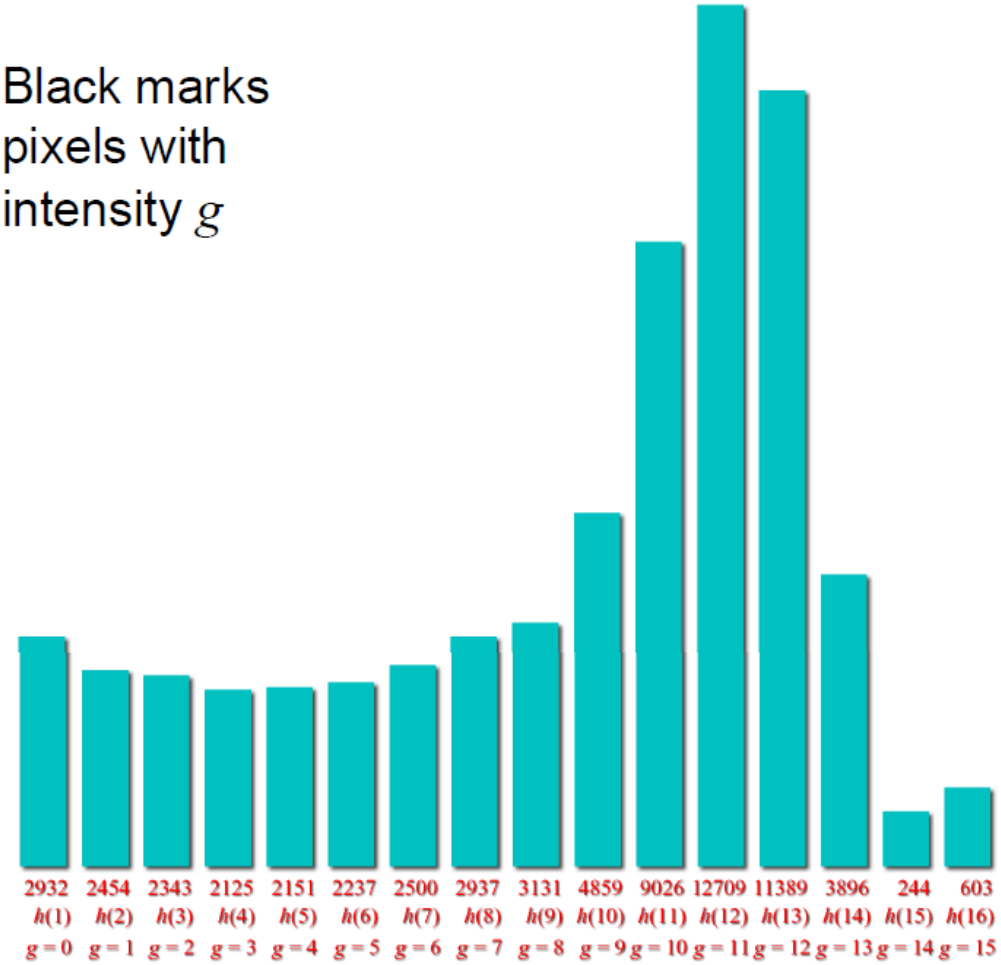


Intensity transformation

The Histogram of a Grayscale Image



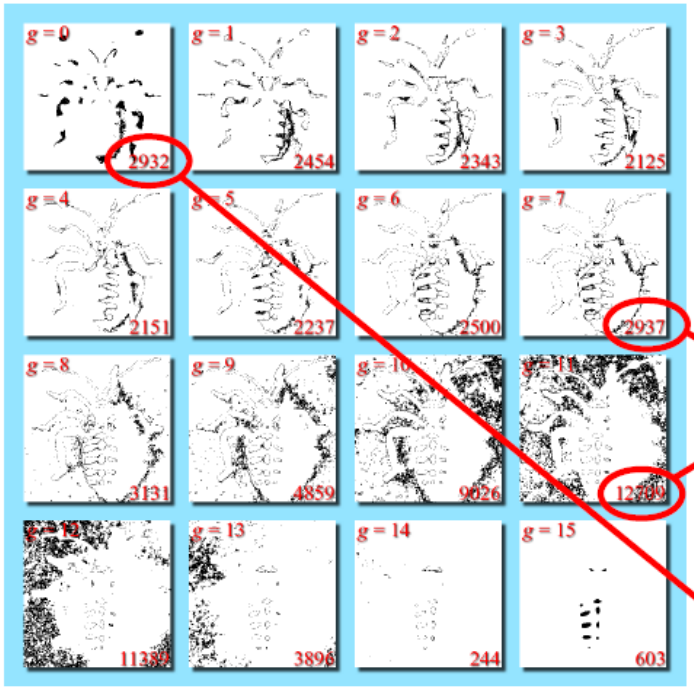
Black marks pixels with intensity g



Plot of histogram:
number of pixels with intensity g

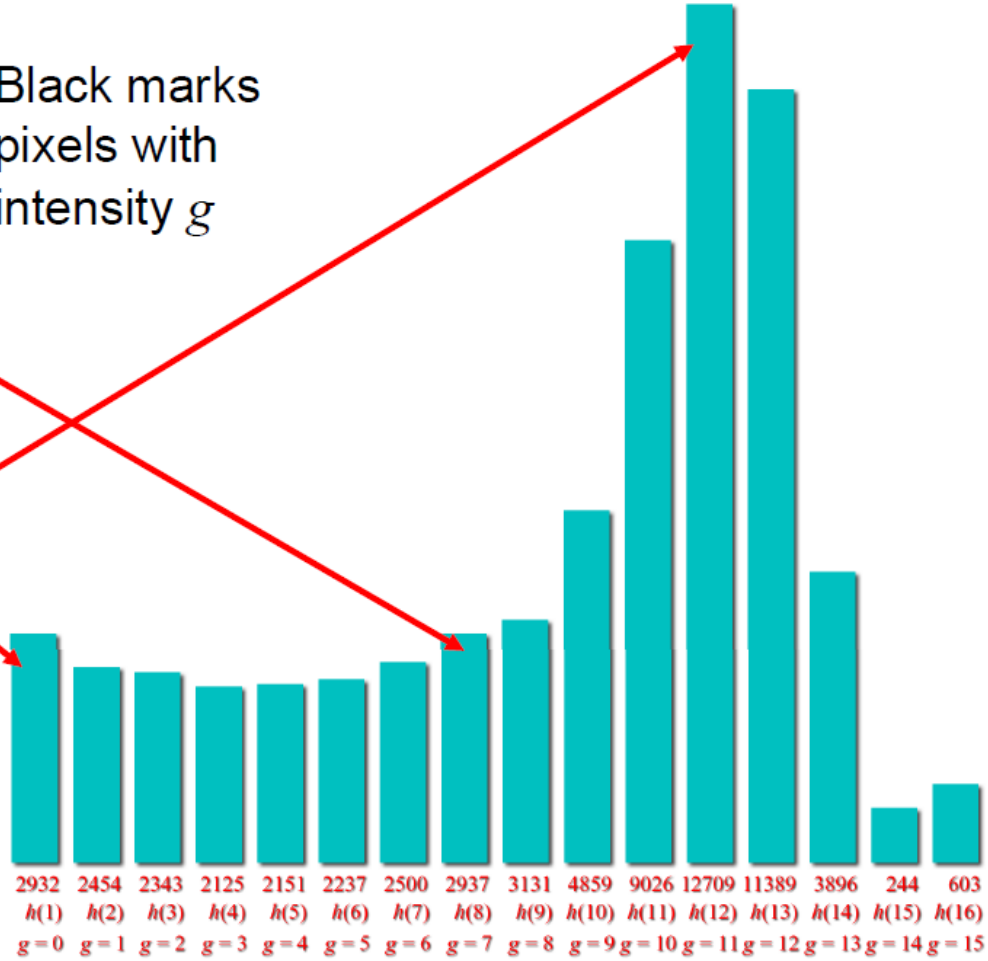
Intensity transformation

The Histogram of a Grayscale Image



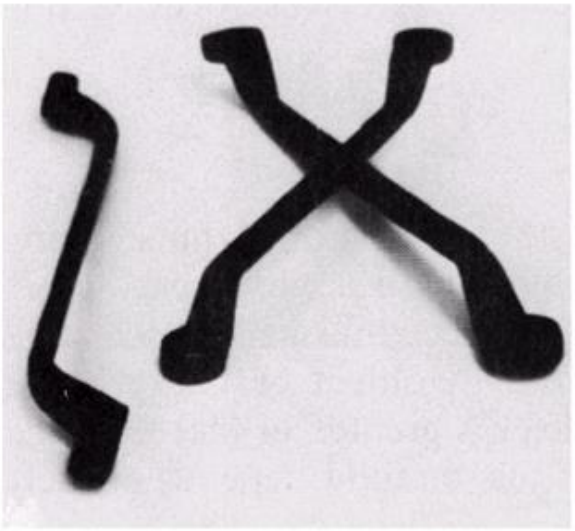
Black marks pixels with intensity g

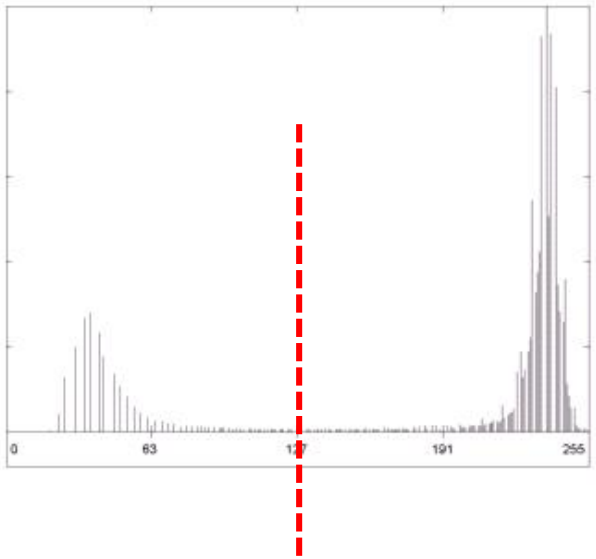
Plot of histogram: number of pixels with intensity g

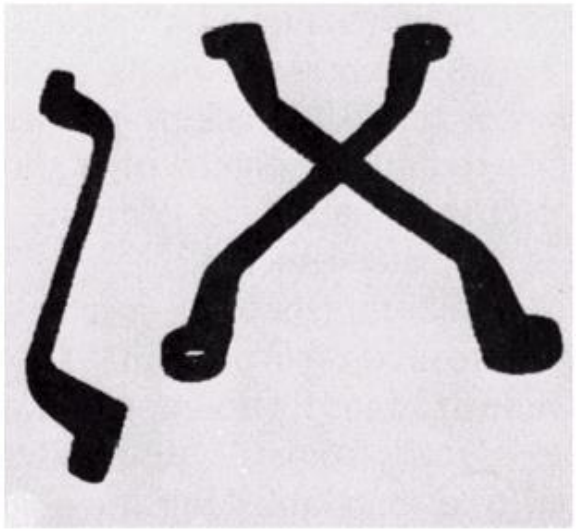


Intensity transformation

Use info from imhist for image segmentation



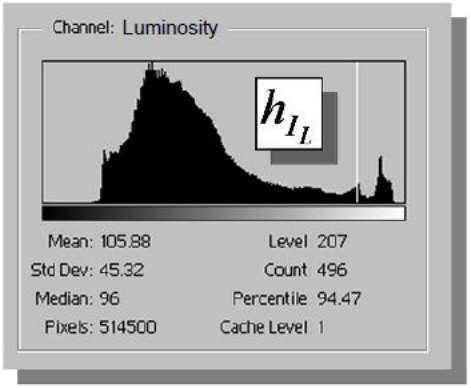


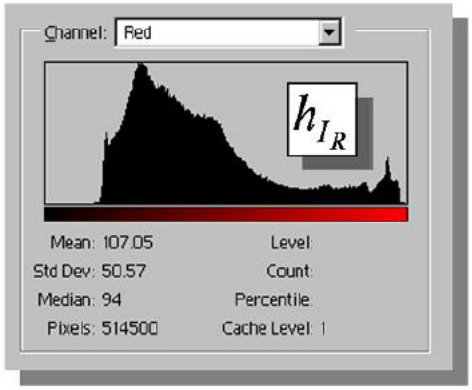


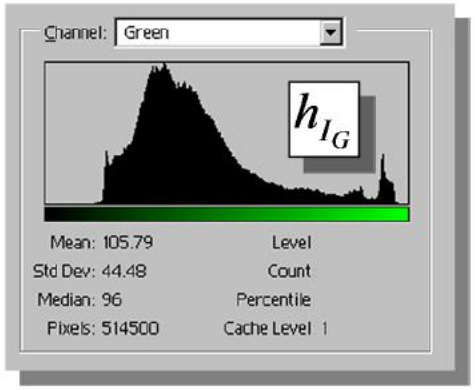
Intensity transformation

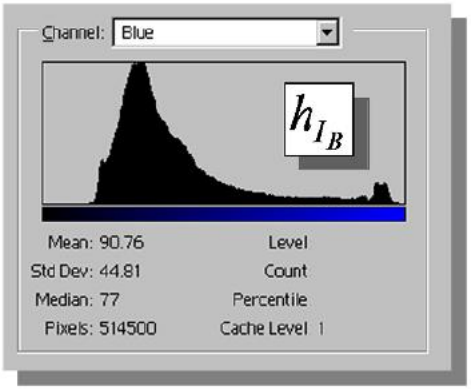
The Histogram of a Color Image

There is one histo-gram per color band R, G, & B. Luminosity histogram is from 1 band = (R+G+B)/3



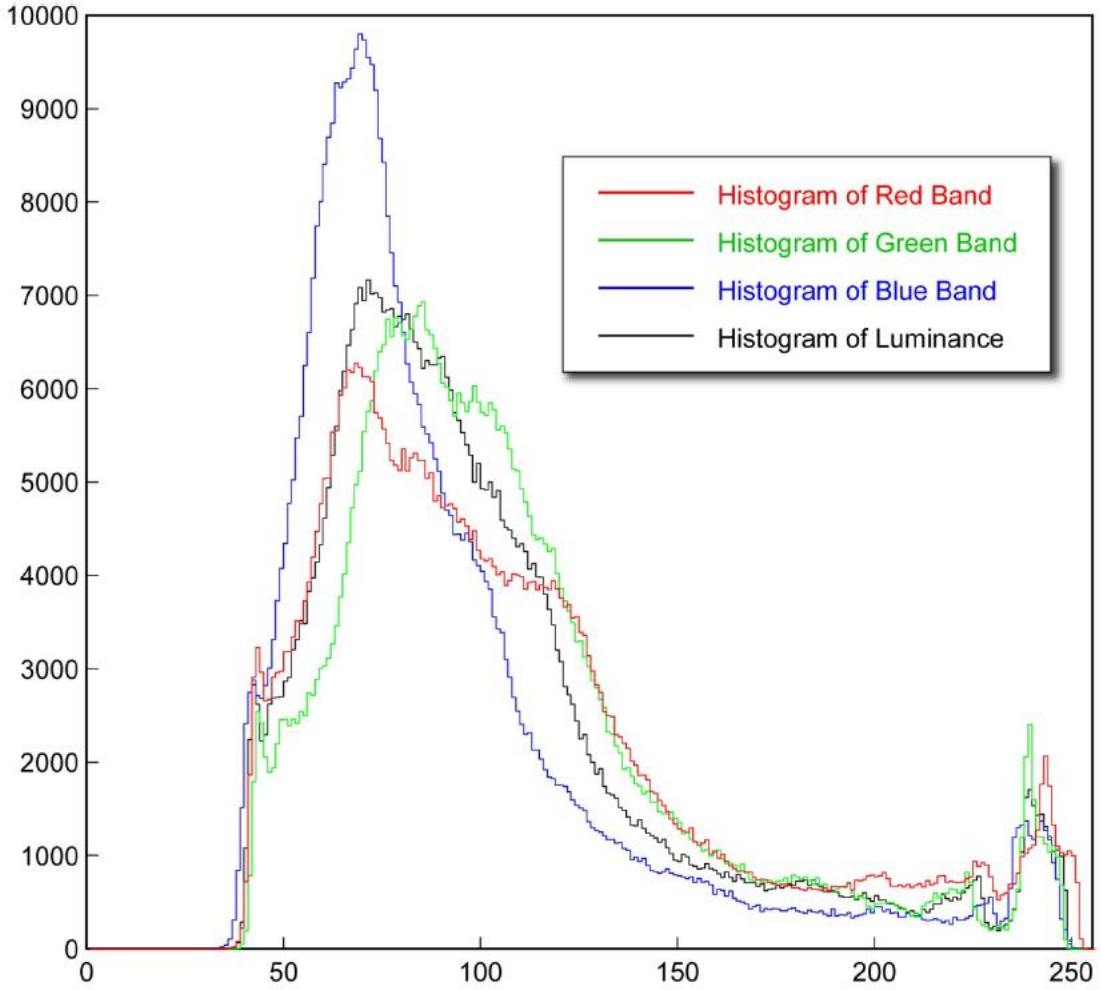






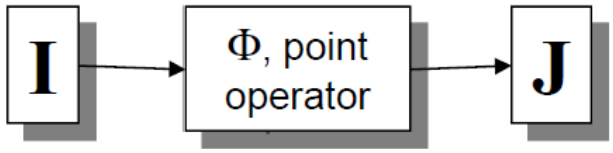
Intensity transformation

The Histogram of a Color Image



Point Ops via Functional Mappings

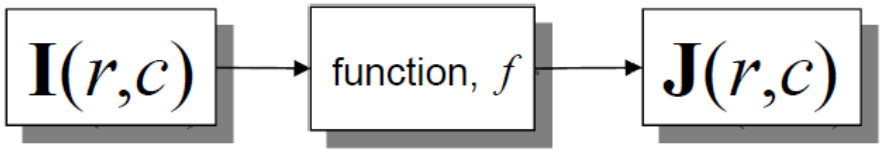
Image:



Input Output

$$\mathbf{J} = \Phi[\mathbf{I}]$$

Pixel:



If $\mathbf{I}(r,c) = g$
and $f(g) = k$
then $\mathbf{J}(r,c) = k$.

The transformation of image **I** into image **J** is accomplished by replacing each input intensity, g, with a specific output intensity, k, at every location (r,c) where $\mathbf{I}(r,c) = g$.

The rule that associates k with g is usually specified with a function, f, so that $f(g) = k$.

Point Ops via Functional Mappings

One-band Image

$$\mathbf{J}(r,c) = f\left(\mathbf{I}(r,c)\right),$$
for all pixels locations (r,c).

Three-band Image

$$\mathbf{J}(r,c,b) = f\left(\mathbf{I}(r,c,b)\right), \text{ or}$$
$$\mathbf{J}(r,c,b) = f_b\left(\mathbf{I}(r,c,b)\right),$$
for $b = 1,2,3$ and all (r,c).

Point Ops via Functional Mappings

One-band Image

$$\mathbf{J}(r,c) = f(\mathbf{I}(r,c)),$$
for all pixels locations (r,c).

Either all 3 bands are mapped through the same function, f, or ...

Three-band Image

$$\mathbf{J}(r,c,b) = f(\mathbf{I}(r,c,b)), \text{ or}$$
$$\mathbf{J}(r,c,b) = f_b(\mathbf{I}(r,c,b)),$$
for $b = 1,2,3$ and all $(r,c$

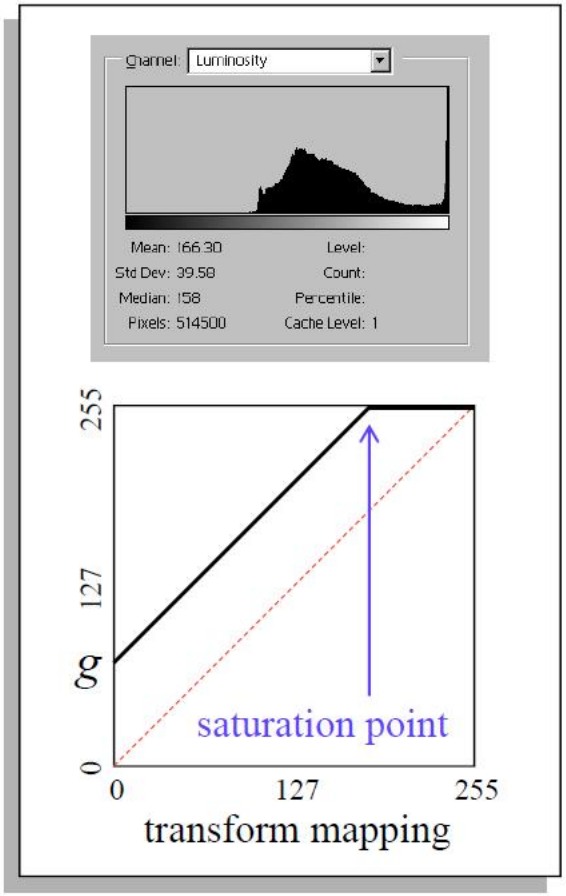
... each band is mapped through a separate function, f_b.

Intensity transformation

Point Processes: Increase Brightness

$$\mathbf{J}(r,c,b) = \begin{cases} \mathbf{I}(r,c,b) + g, & \text{if } \mathbf{I}(r,c,b) + g < 256 \\ 255, & \text{if } \mathbf{I}(r,c,b) + g > 255 \end{cases}$$

$g \geq 0$ and $b \in \{1,2,3\}$ is the band index.



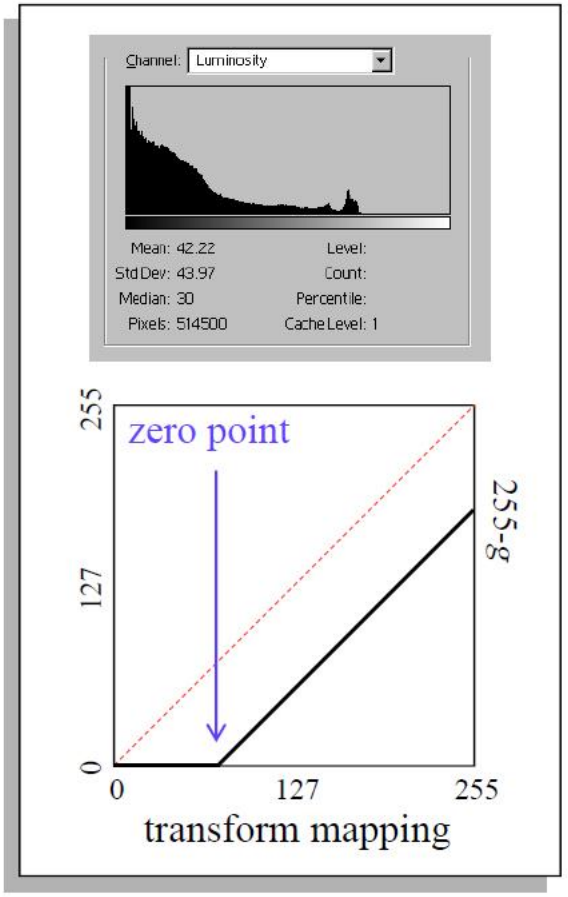
Try g=200

Intensity transformation

Point Processes: Decrease Brightness

$$\mathbf{J}(r,c,b) = \begin{cases} 0, & \text{if } \mathbf{I}(r,c,b) - g < 0 \\ \mathbf{I}(r,c,b) - g, & \text{if } \mathbf{I}(r,c,b) - g > 0 \end{cases}$$

$g \geq 0$ and $b \in \{1,2,3\}$ is the band index.



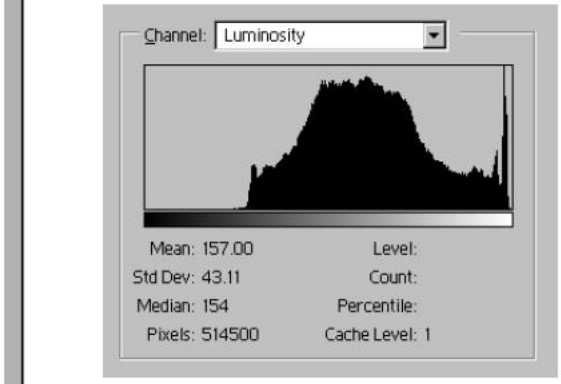
Try g=100

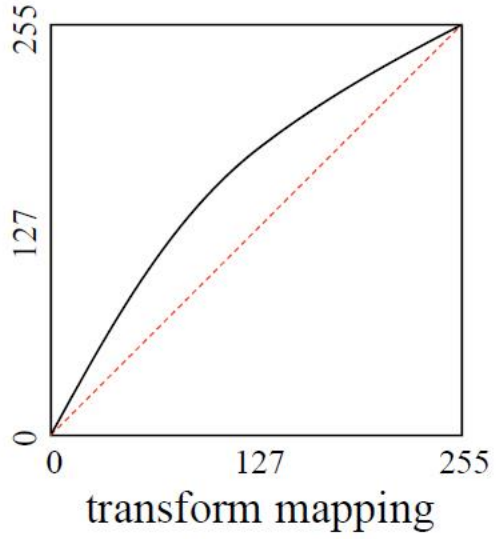
Intensity transformation

Point Processes: Increased Gamma

Try r=4

$$\mathbf{J}(r,c) = 255 \cdot \left[\frac{\mathbf{I}(r,c)}{255} \right]^{\frac{1}{\gamma}} \quad \text{for} \quad \gamma > 1.0$$





Intensity transformation

Point Processes: Decreased Gamma

Try r=0.4

$$\mathbf{J}(r,c) = 255 \cdot \left[\frac{\mathbf{I}(r,c)}{255}\right]^{\frac{1}{\gamma}} \quad \text{for} \quad \gamma < 1.0$$