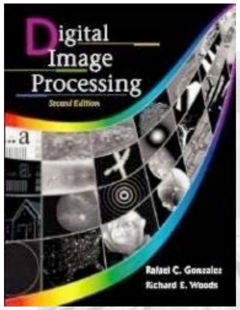
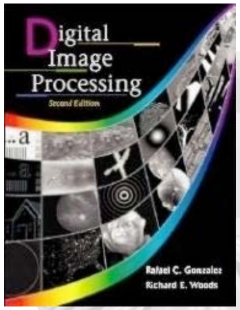


Lecture #2

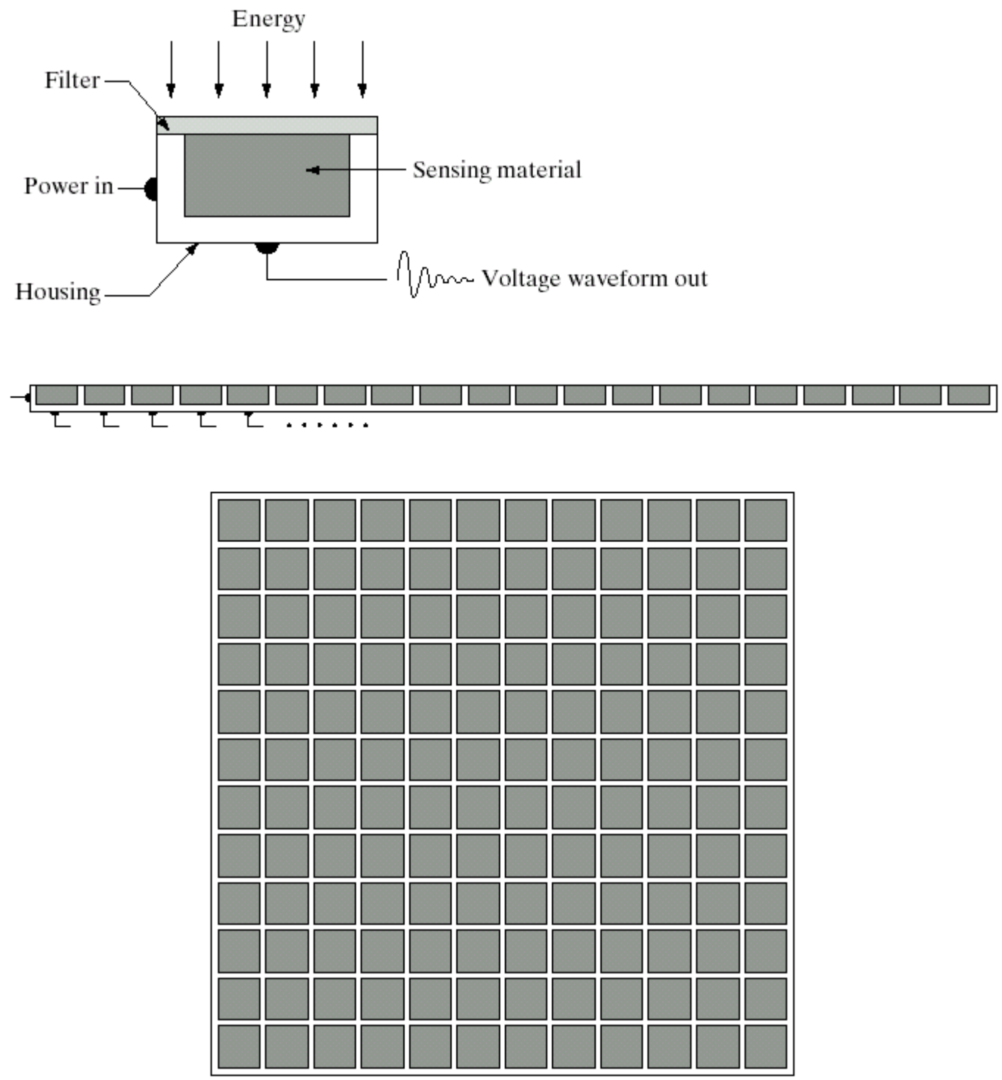
- Image acquisition
- Images in the spatial domain
  - Digital representation
  - Sampling
  - Quantization
  - Spatial resolution
  - Gray scale resolution
  - Resampling
- MATLAB® image processing
  - Reading and writing images
  - MATLAB® classes: uint8 and double
  - Adding and multiplying images

Image acquisition

- vidicons and other “tube”sensors
- CCD arrays
- CID arrays
- photodiode arrays
- specialized sensors, i.e., infrared, document scanning



Chapter 2: Digital Image Fundamentals



a
b
c

FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

We usually idealize sensors as square but, in reality, they are not and manufacturers data needs to be checked for critical applications.

Chapter 2: Digital Image Fundamentals

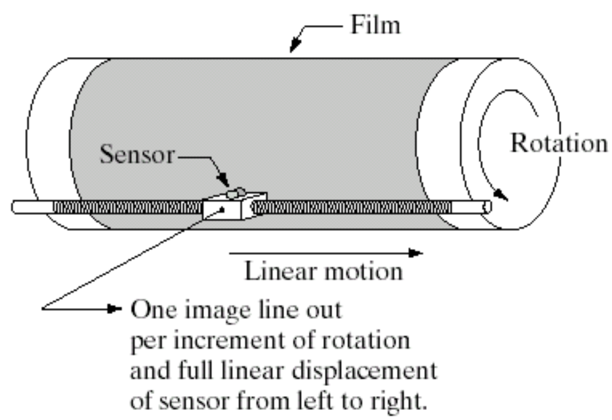


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

Chapter 2: Digital Image Fundamentals

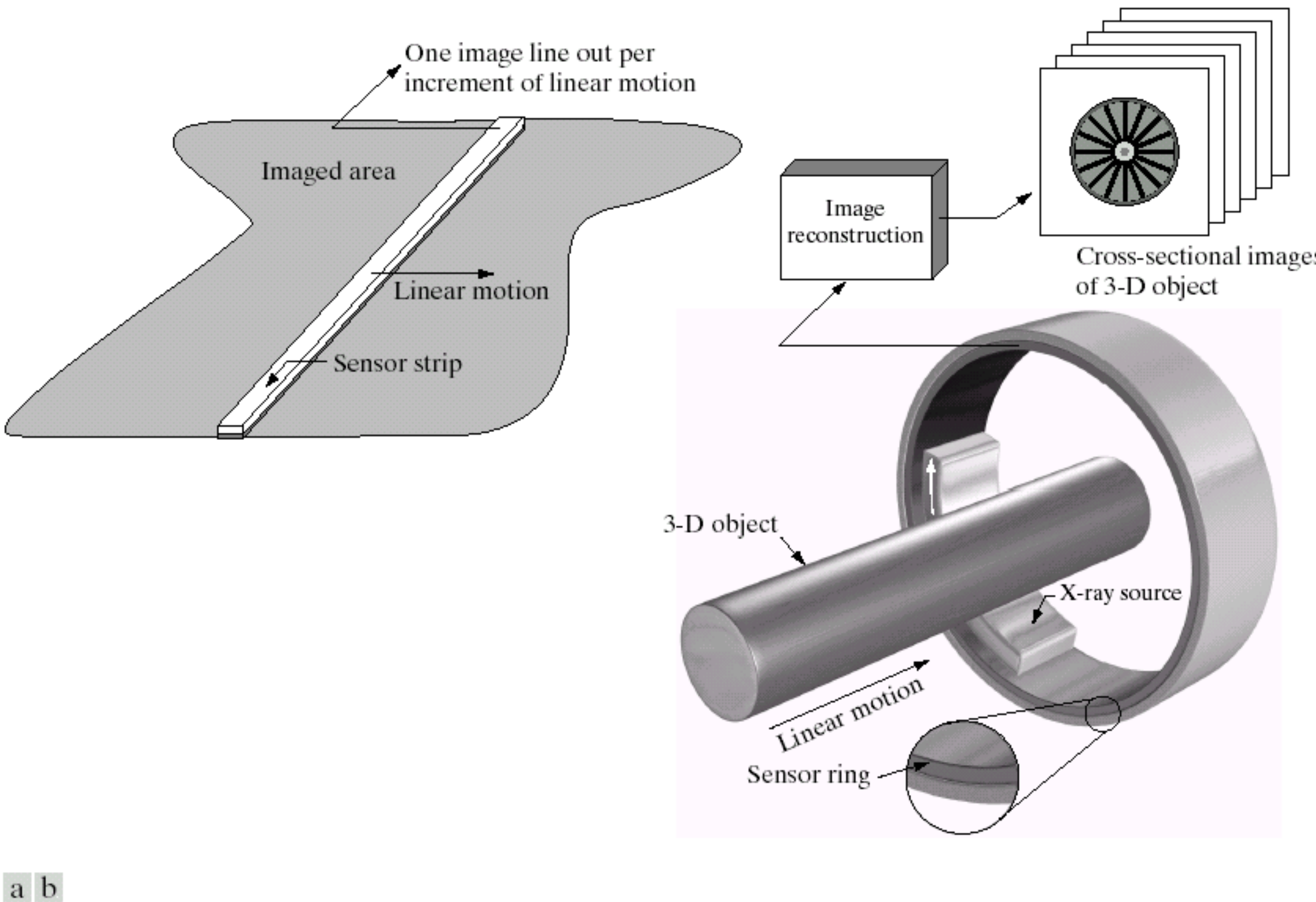
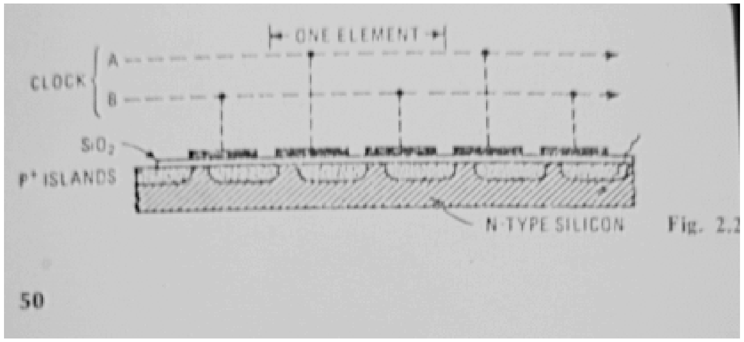


FIGURE 2.14 (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

Solid-State Image Sensors



Solid-state sensors require wiring (interconnects) to read out image data. The arrangement of the sensors and the manner and order in which they are read out can vary considerably among different sensors.

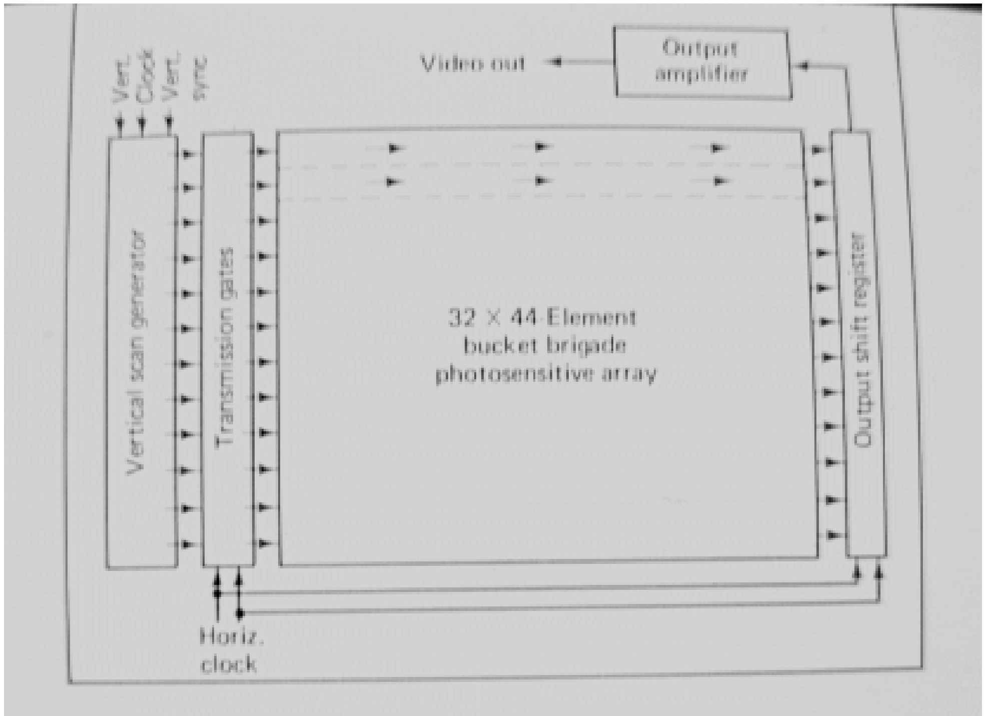
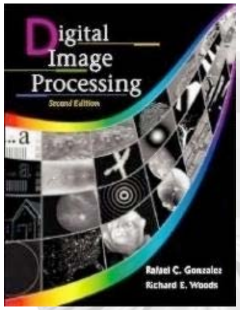
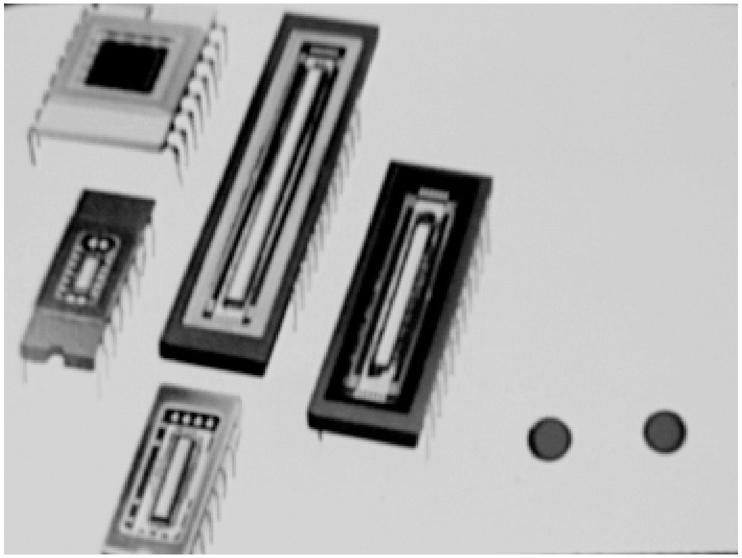


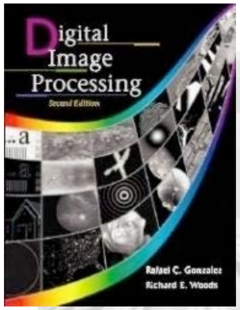
Fig. 2.23 A CCD array (line transfer).



Digitial arrays come in many arrangements and sizes

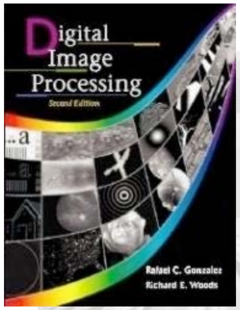


You can get imaging sensors in many sizes and shapes for specialized applications.

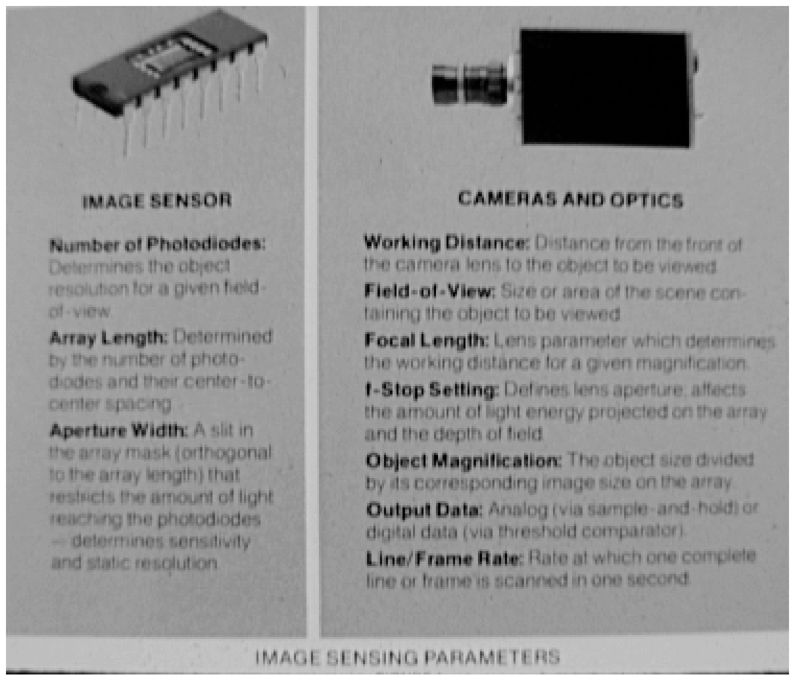


Comparing CCD and CID image sensors

- They are sensitive over a wide spectral range, from 450 to 1,600 nanometers (corresponding to the range from blue light through the visible spectrum to the near infrared region)
- They operate on low voltages and consume only a small amount of power.
- They do not exhibit lag or memory, so that the traces of moving objects are not smeared.
- They are not damaged by intense light. Present devices will oversaturate and “bloom” under intense light but are not permanently damaged (as a vidicon tube might be, for example).
- Their positioning accuracy and therefore measurement accuracy are very good because of the accurate photolithography process used to form them.



Camera/image sensor



Analog RS-170 video signals

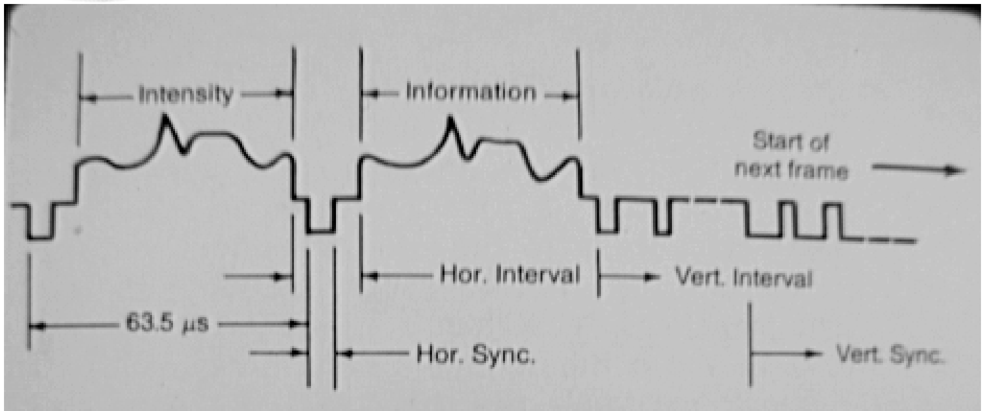


Figure 9.2 Composite video signal to the EIA RS-170 standard

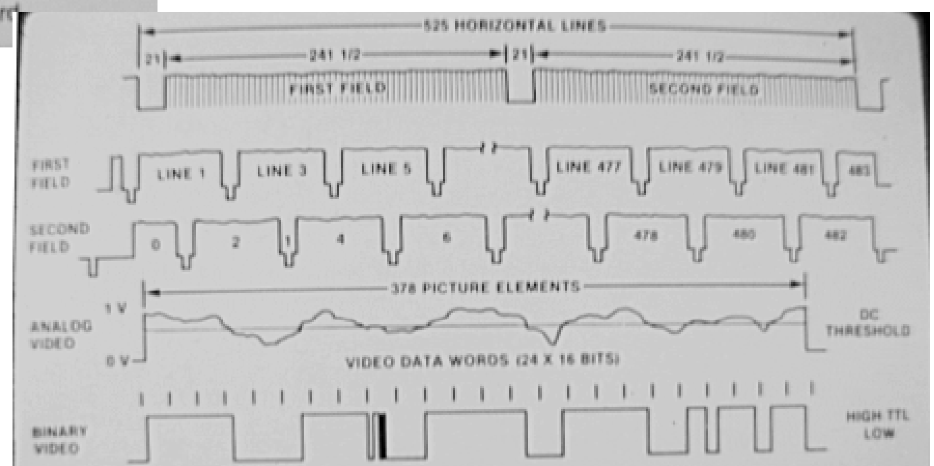


Figure 9.6 Video conversion of image data from CCD3000 into binary words.

Film

- Once dominating image recording, digital techniques have replaced film for most applications

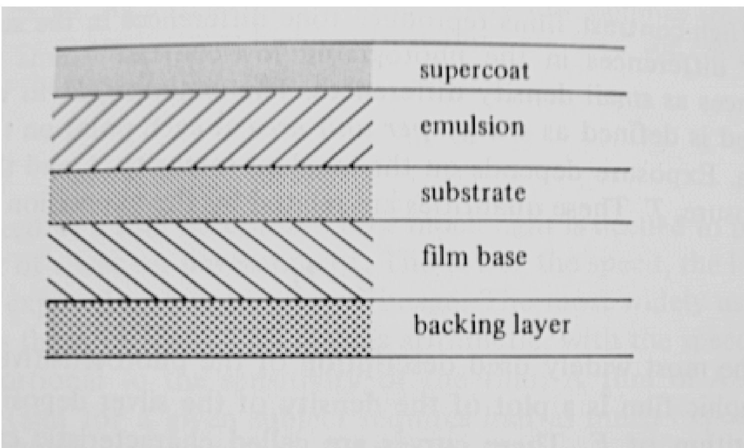


Figure 2.11. Structure of a modern black-and-white film.

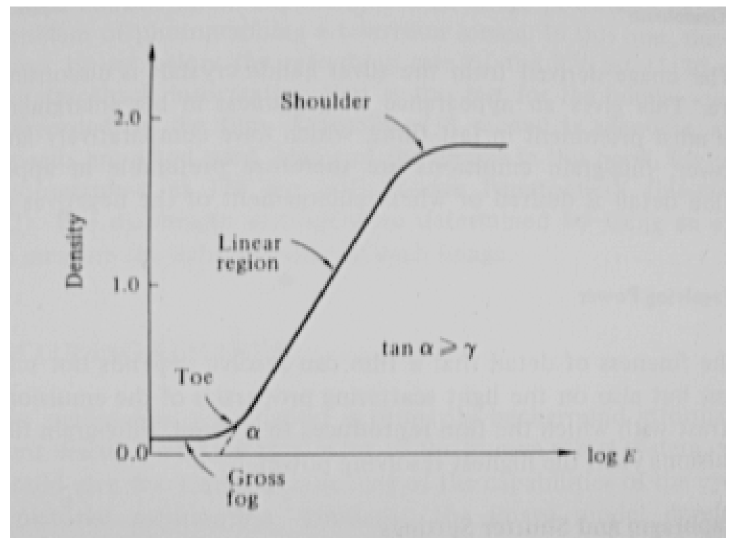
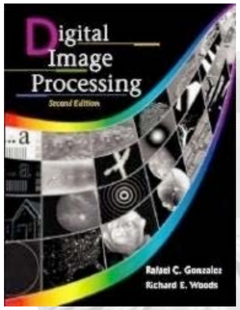


Figure 2.12. A typical H & D curve.

Images in the spatial domain

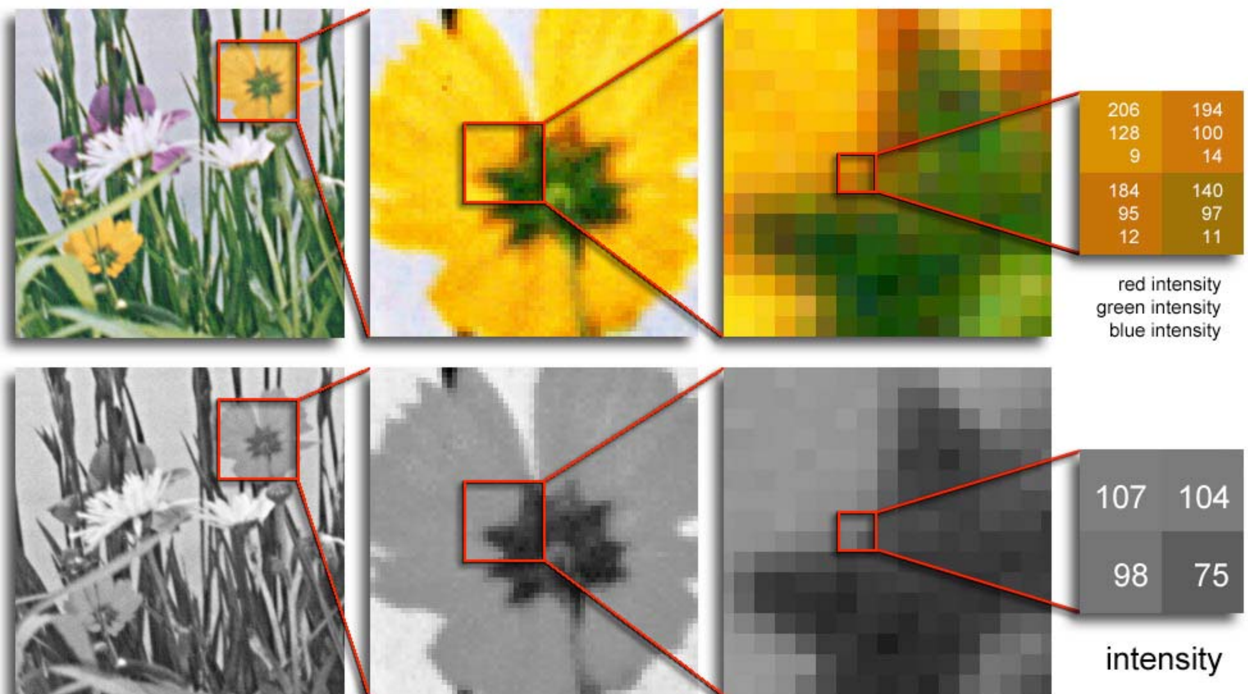


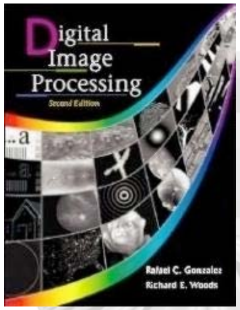
Digital Image

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*)



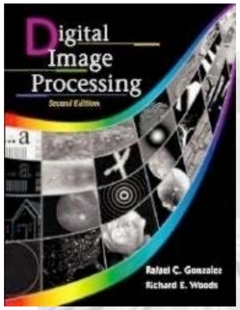


Pixels

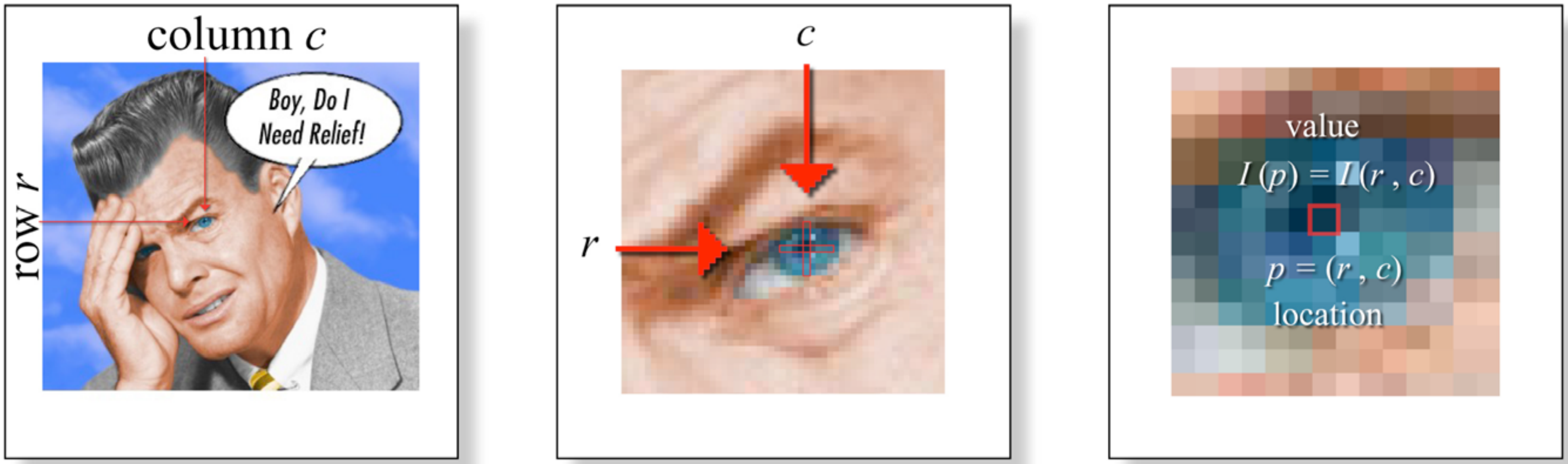
- A digital image, I, is a mapping from a 2D grid of uniformly spaced discrete points, $\{p = (r,c)\}$, into a set of positive integer values, $\{I(p)\}$, or a set of vector values, *e.g.*, $\{[R\ G\ B]^T(p)\}$.
- At each column location in each row of I there is a value.
- The pair (p, $I(p)$) is called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

Digital Image

- $p = (r,c)$ 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).



Pixels



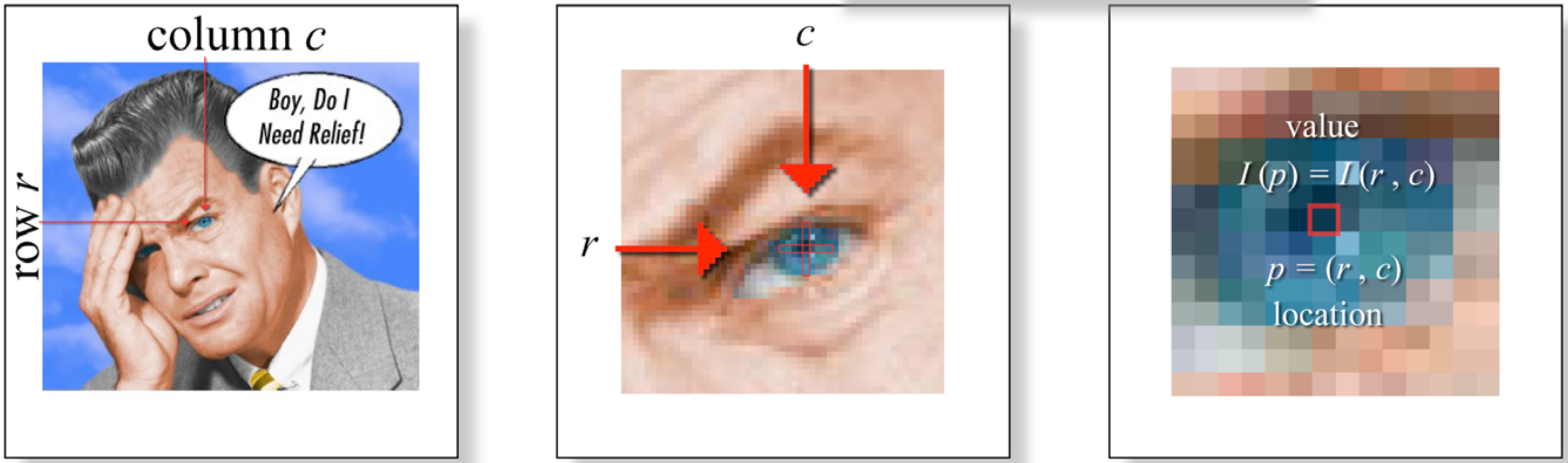
Pixel Location: $p = (r\ ,\ c)$

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

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

Digital Image

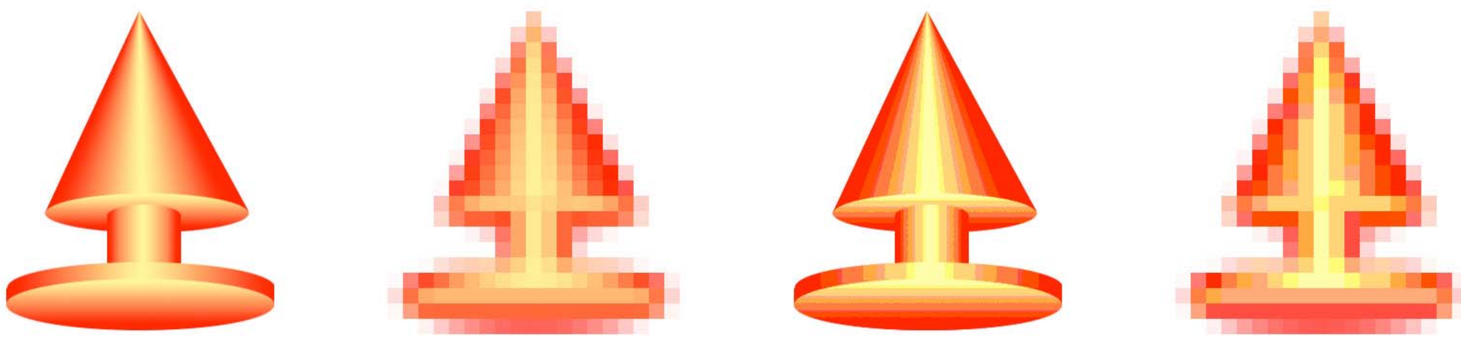
Pixel : [p, $I(p)$]



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

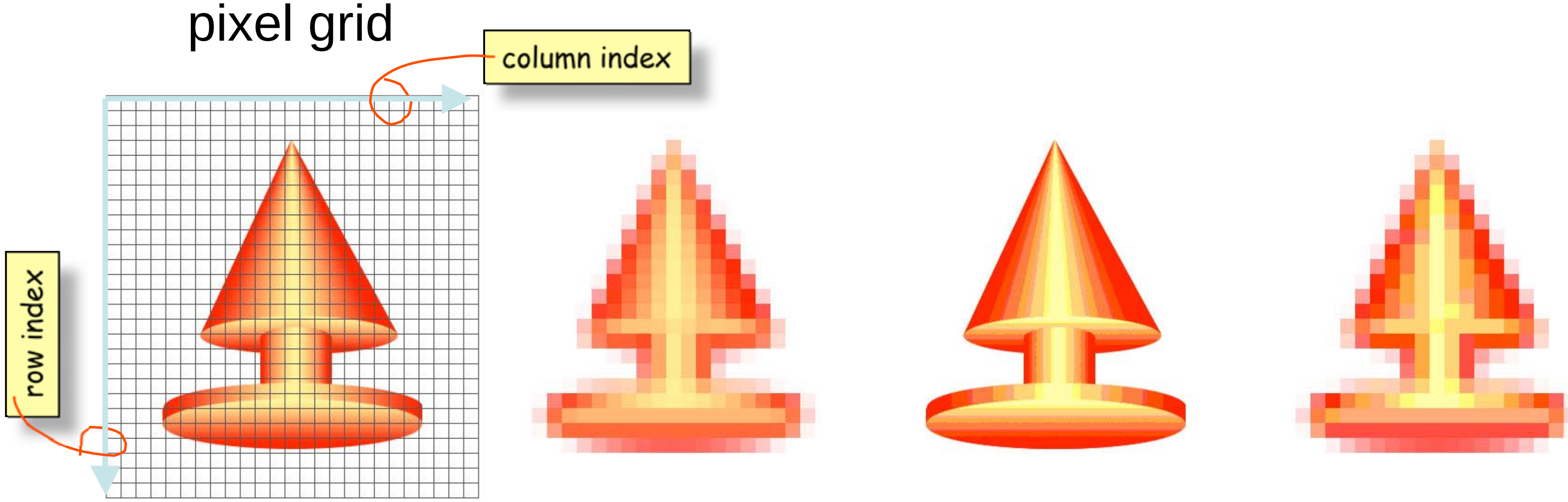
$$I(p) = \begin{bmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{bmatrix} = \begin{bmatrix} 12 \\ 43 \\ 61 \end{bmatrix}$$

Sampling & Quantization



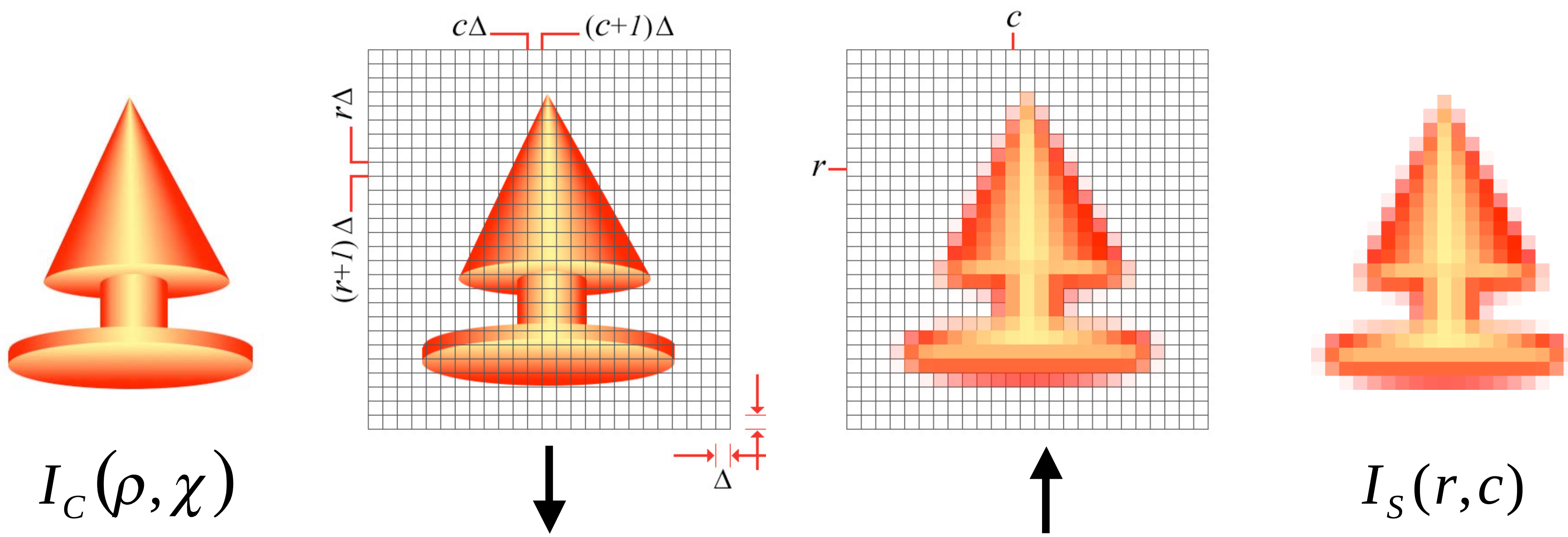
real image	sampled	quantized	sampled & quantized
	p, space	*I(p), space*	

Sampling & Quantization



real image | sampled | quantized | sampled & quantized

Sampling



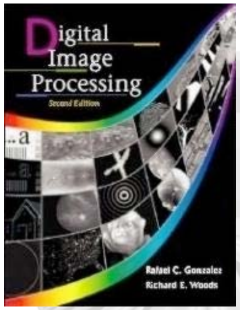
$I_C(\rho,\chi)$

continuous image

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

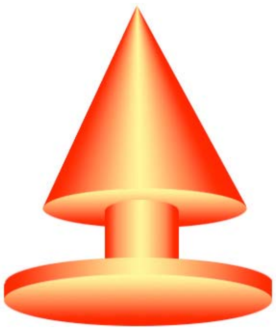
$I_S(r,c)$

sampled image



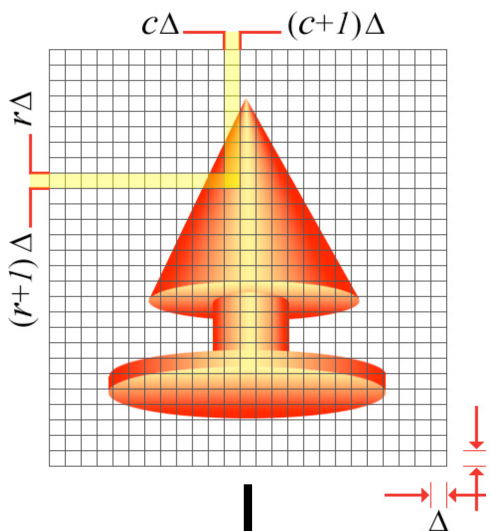
Sampling

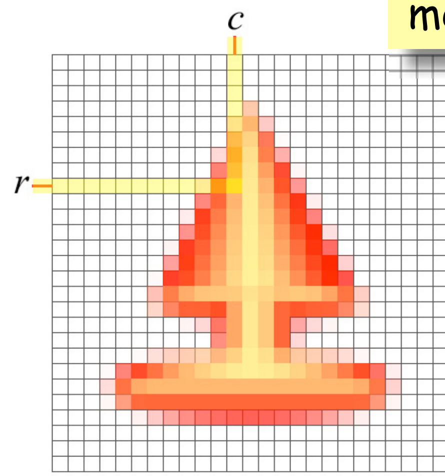
Take the average within each square is the most common method of sampling.

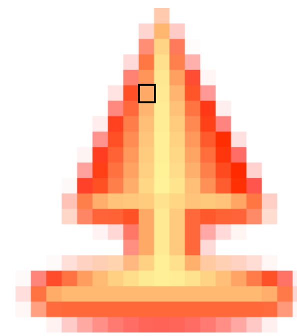


$I_C(\rho,\chi)$

continuous image



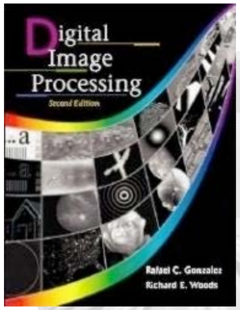




$I_S(r,c)$

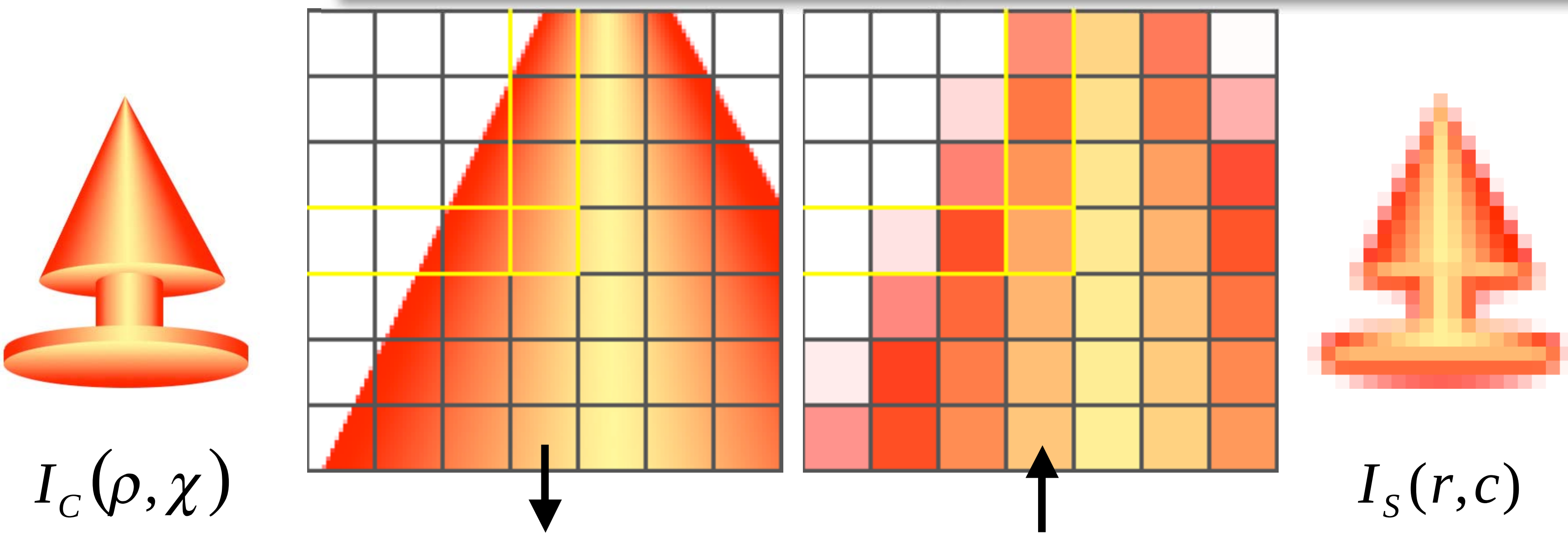
sampled image

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



Sampling

A common assumption is that Δ is the same for r and c. This is not always true.



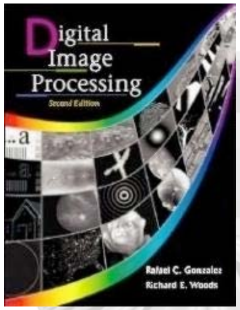
$I_C(\rho,\chi)$

continuous image

$I_S(r,c)$

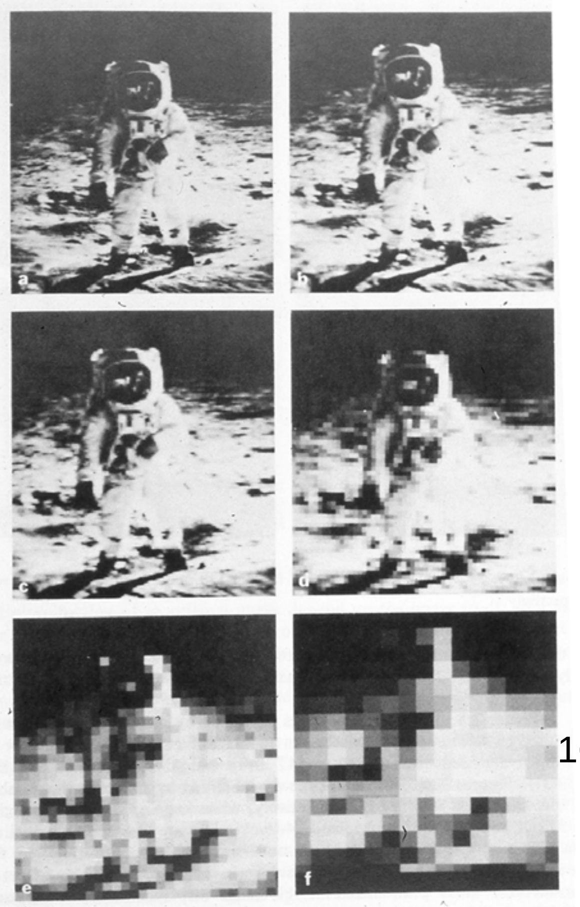
sampled image

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



Spatial Resolution

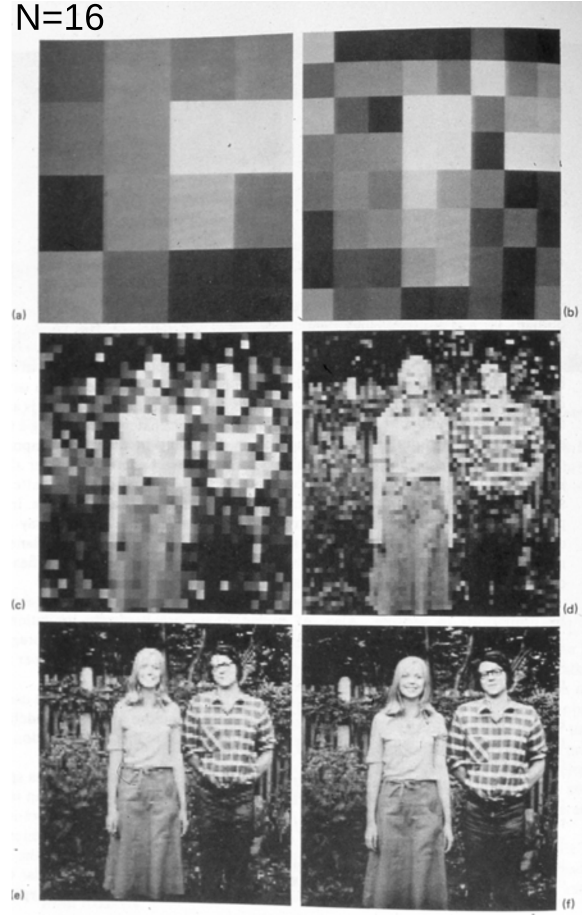
1024x1024



16x16

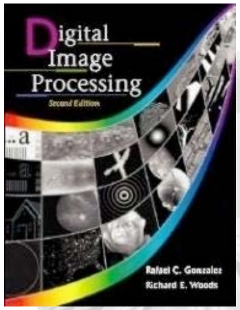
Figure 2.7. Effects of reducing sampling-grid size.

N=16



N=512

Fig. 2.9 Using different numbers of samples. (a) $N = 16$; (b) $N = 32$; (c) $N = 64$; (d) $N = 128$; (e) $N = 256$; (f) $N = 512$.



Gray Scale Resolution

N=256

N=2

Figure 2.8. A 512 X 512 image displayed in 256, 128, 64, 32, 16, 8, 4, and 2 levels, respectively.

Gray Scale Resolution

m=1 bit

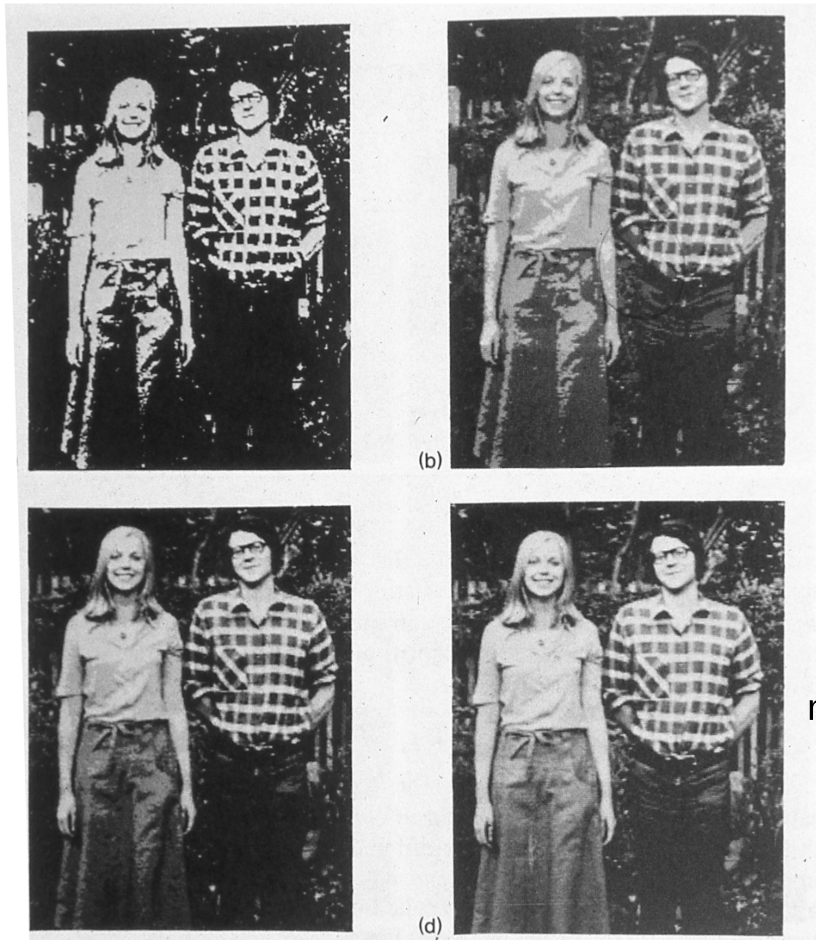


Fig. 2.10 Using different numbers of bits per sample. (a) $m = 1$; (b) $m = 2$; (c) $m = 4$; (d) $m = 8$.

m=8 bits

Subsampling

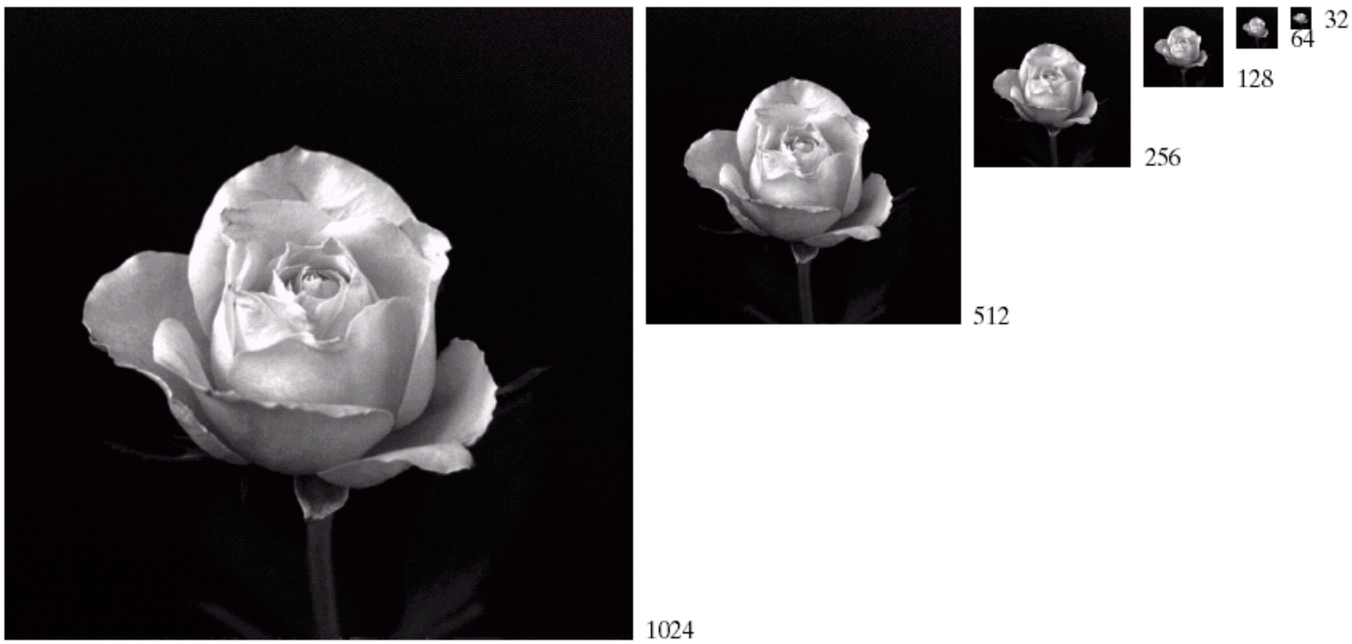
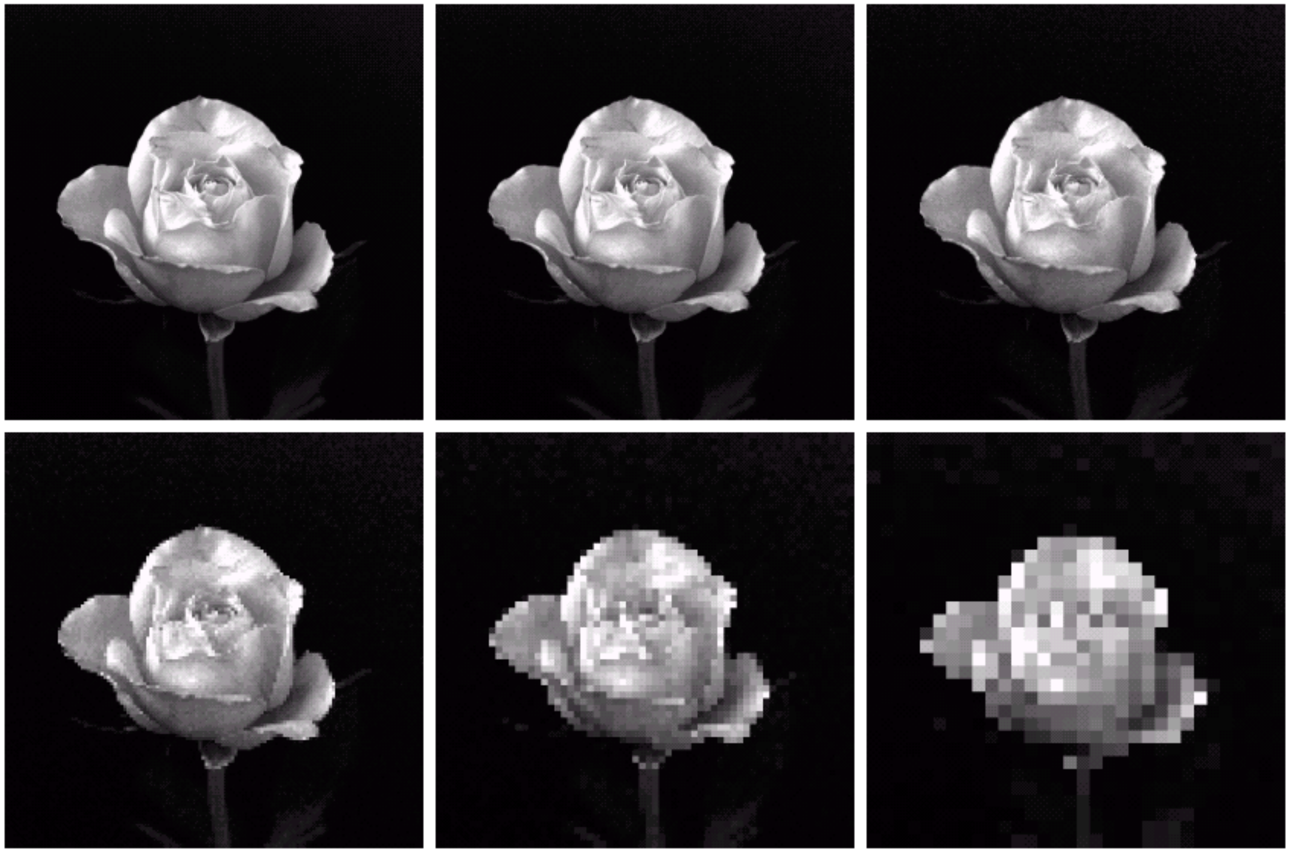


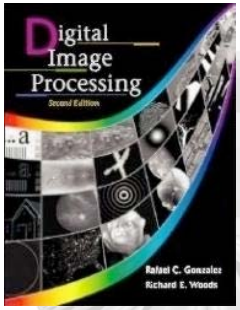
FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

Resampling



a b c
d e f

FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.



Subsampling

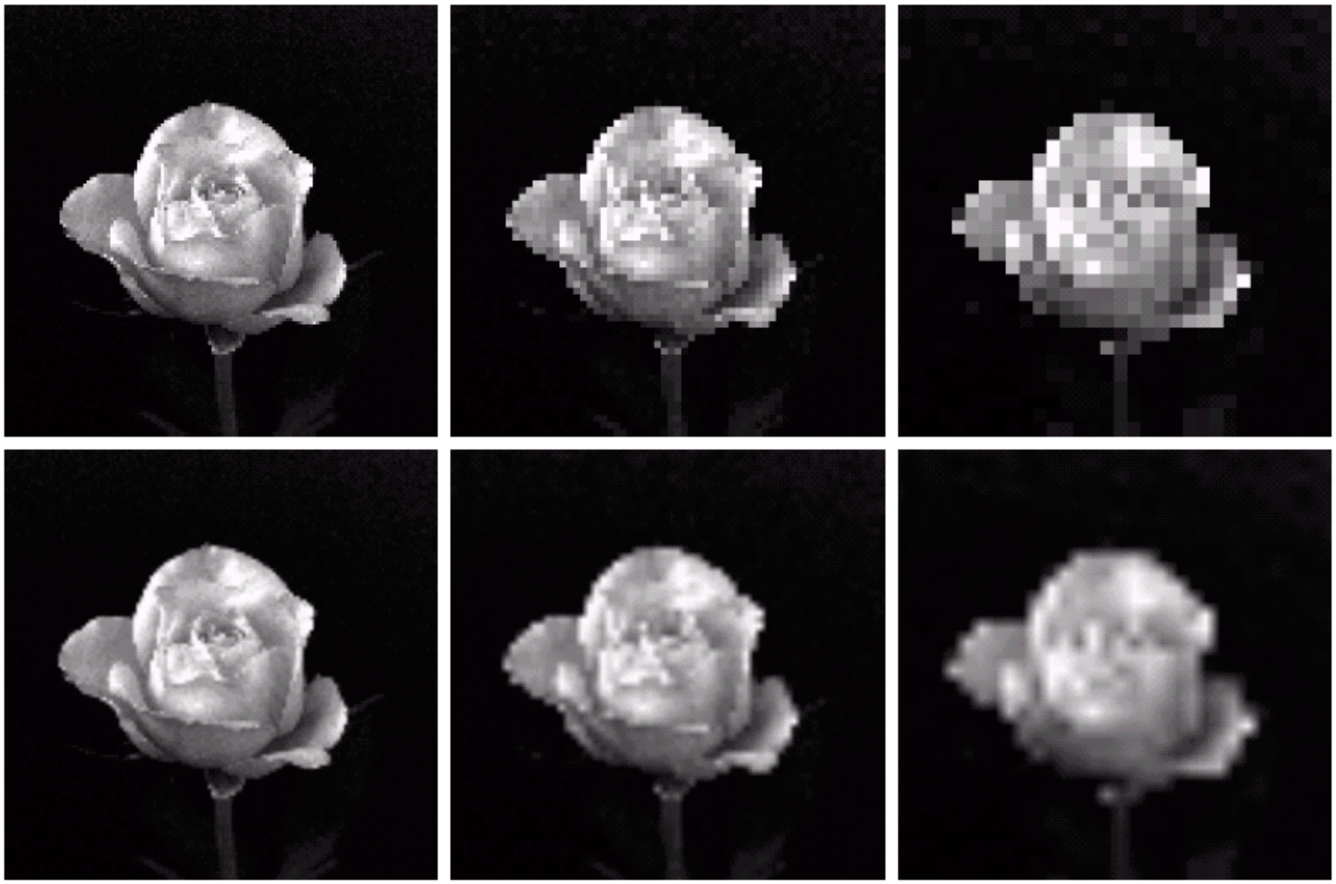
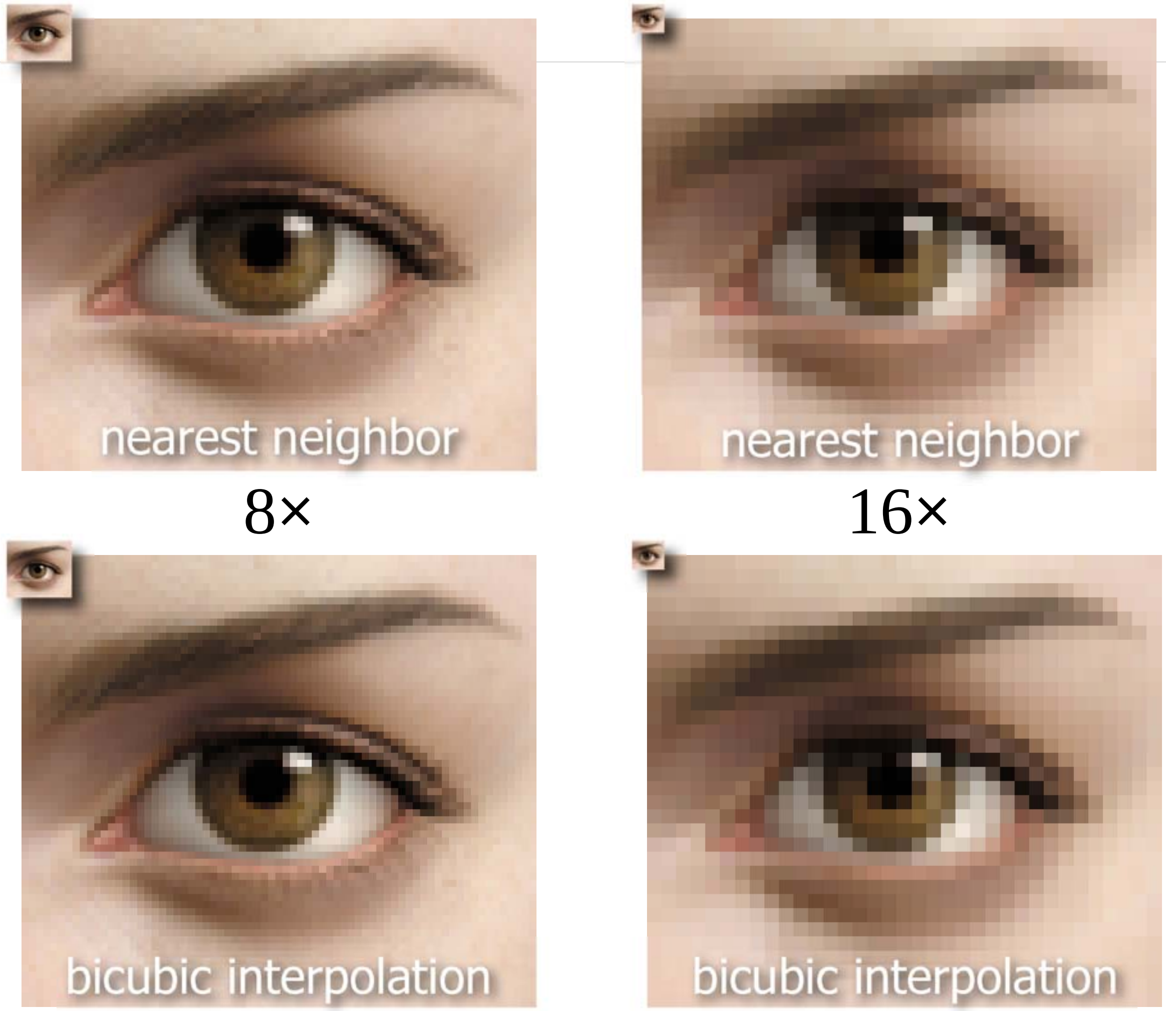
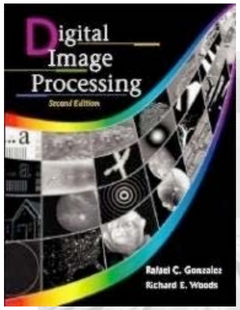


FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

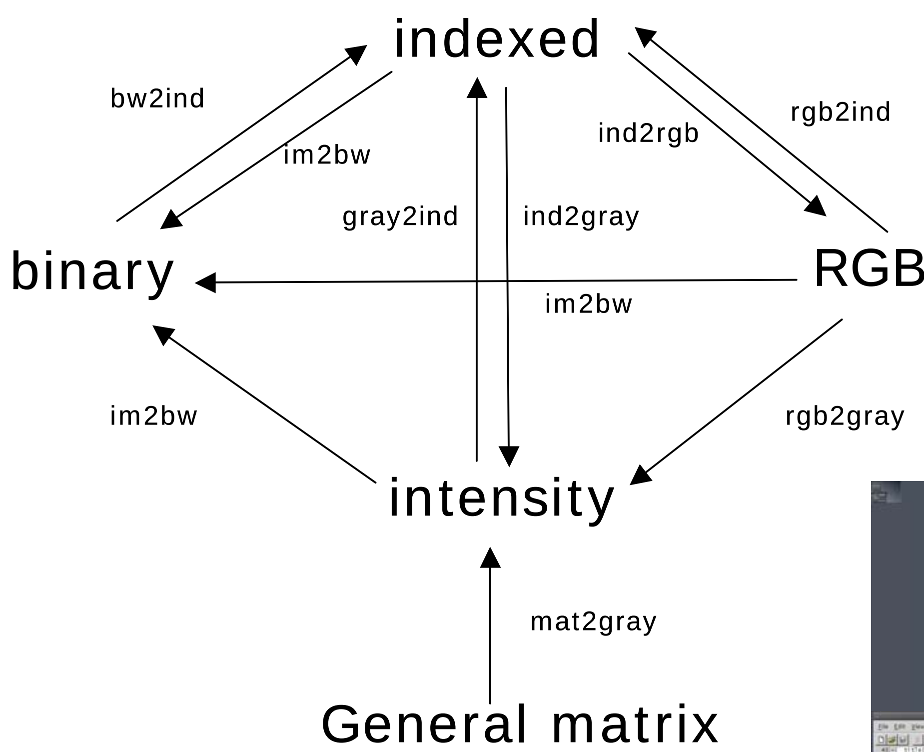
Resampling

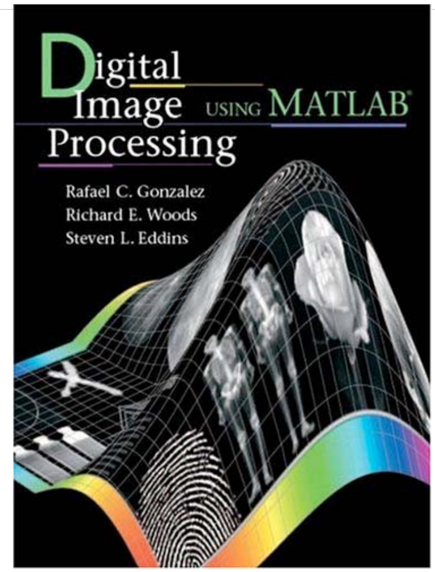
(resizing)

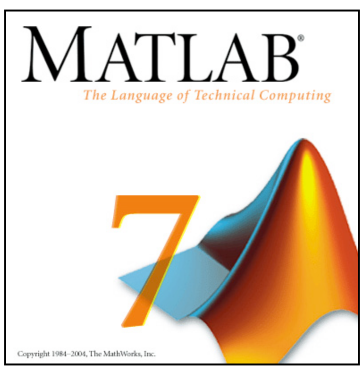


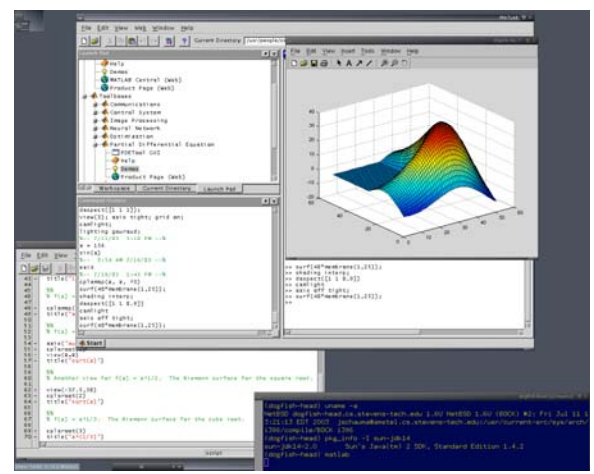


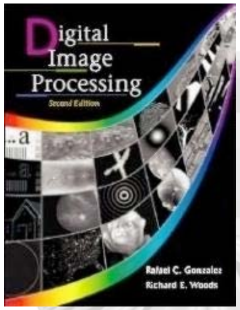
MATLAB® Image Types



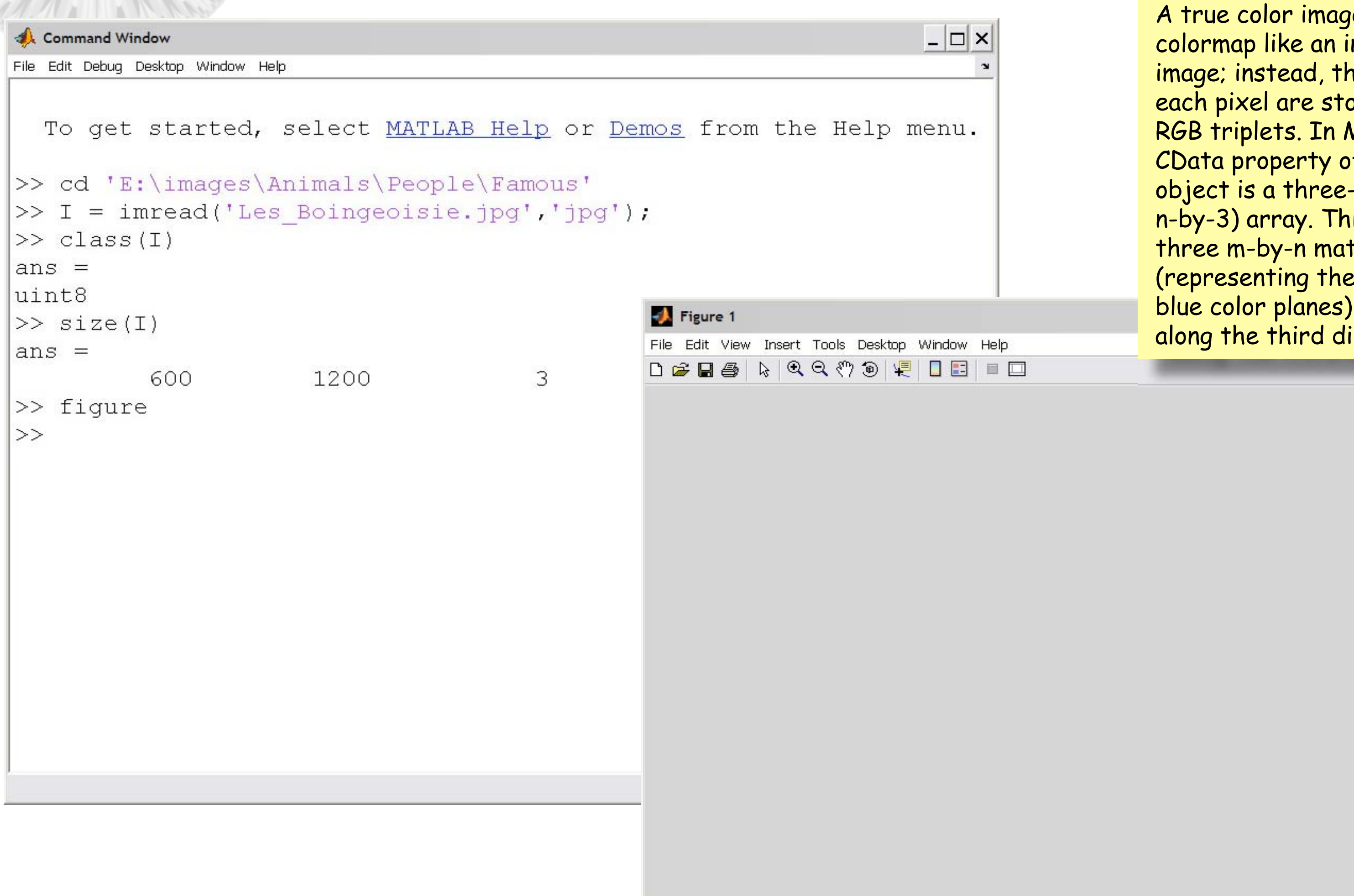




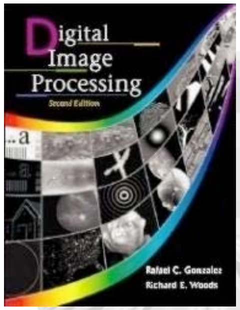




Read a Truecolor Image into Matlab

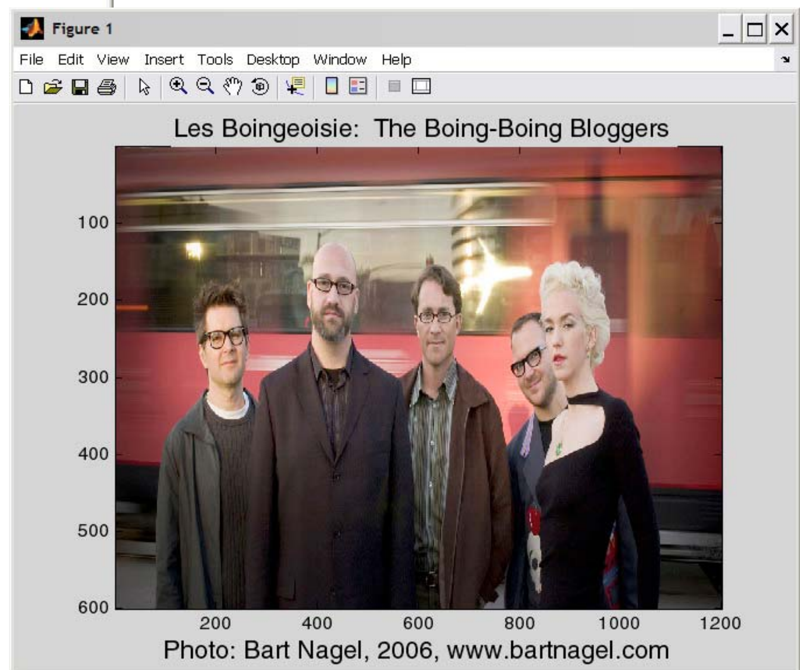


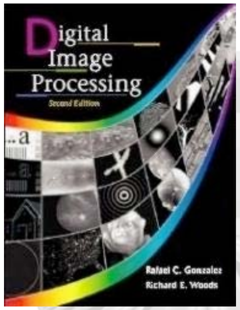
A true color image does not use a colormap like an indexed color image; instead, the color values for each pixel are stored directly as RGB triplets. In MATLAB , the CData property of a truecolor image object is a three-dimensional (m-by-n-by-3) array. This array consists of three m-by-n matrices (representing the red, green, and blue color planes) concatenated along the third dimension.



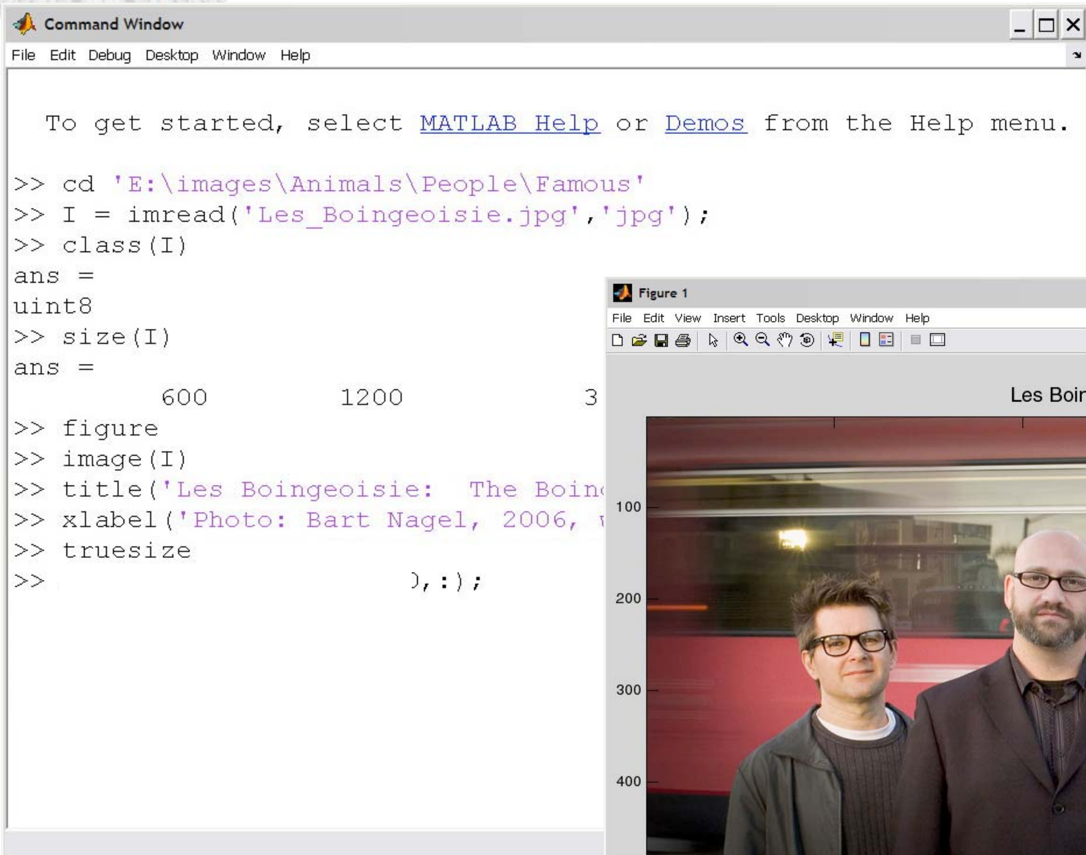
Read a Truecolor Image into Matlab

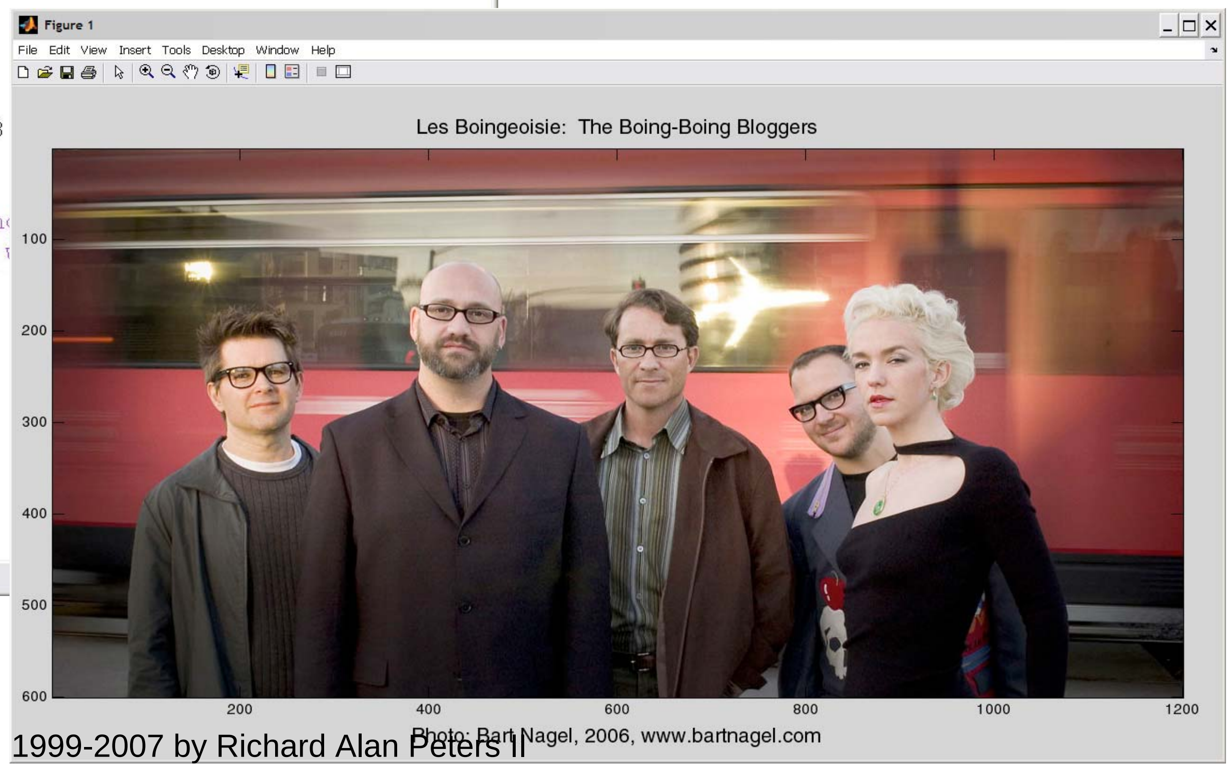
```
Command Window
File  Edit  Debug  Desktop  Window  Help

  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')
>>
```

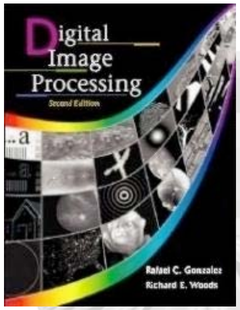




Read a Truecolor Image into Matlab

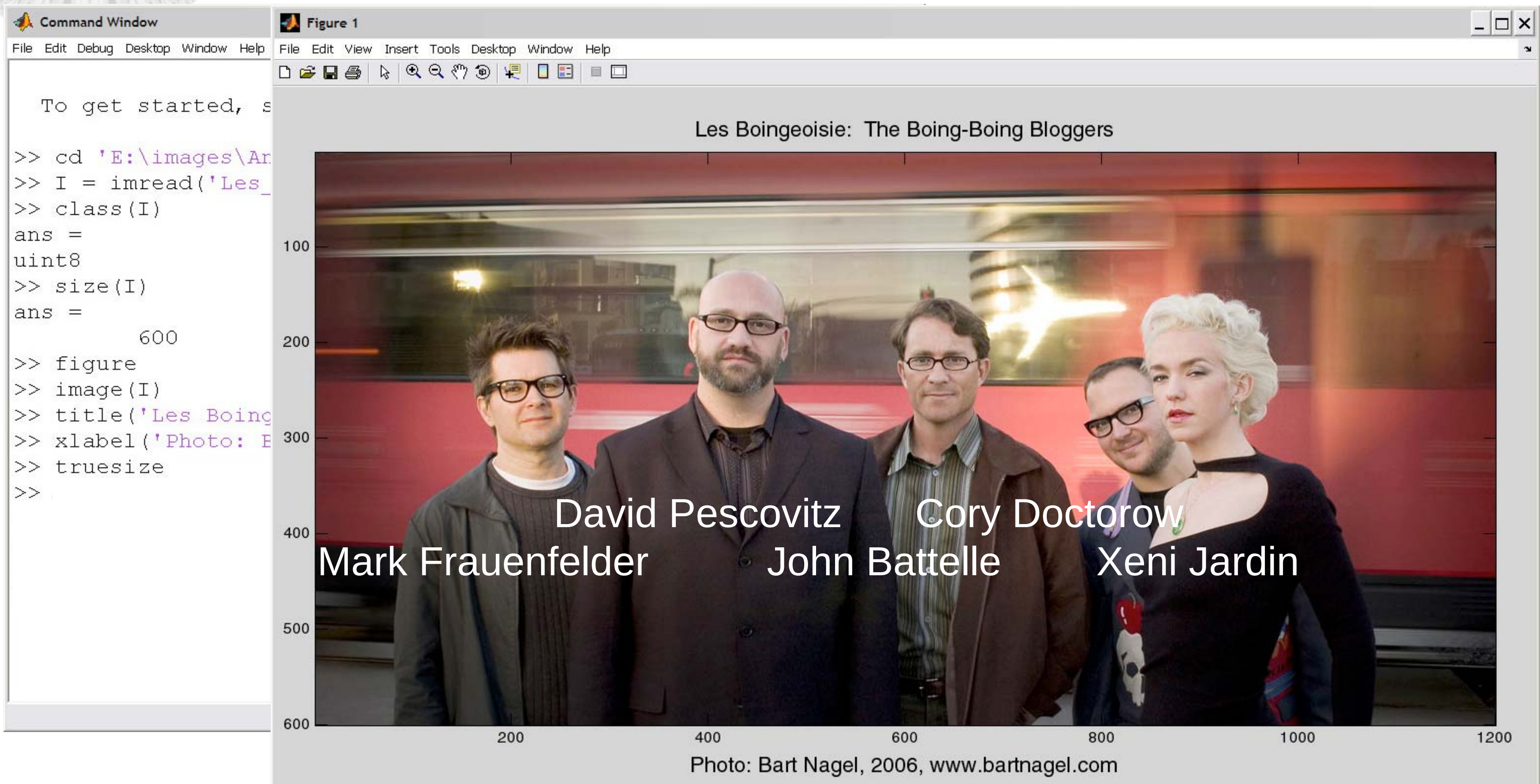




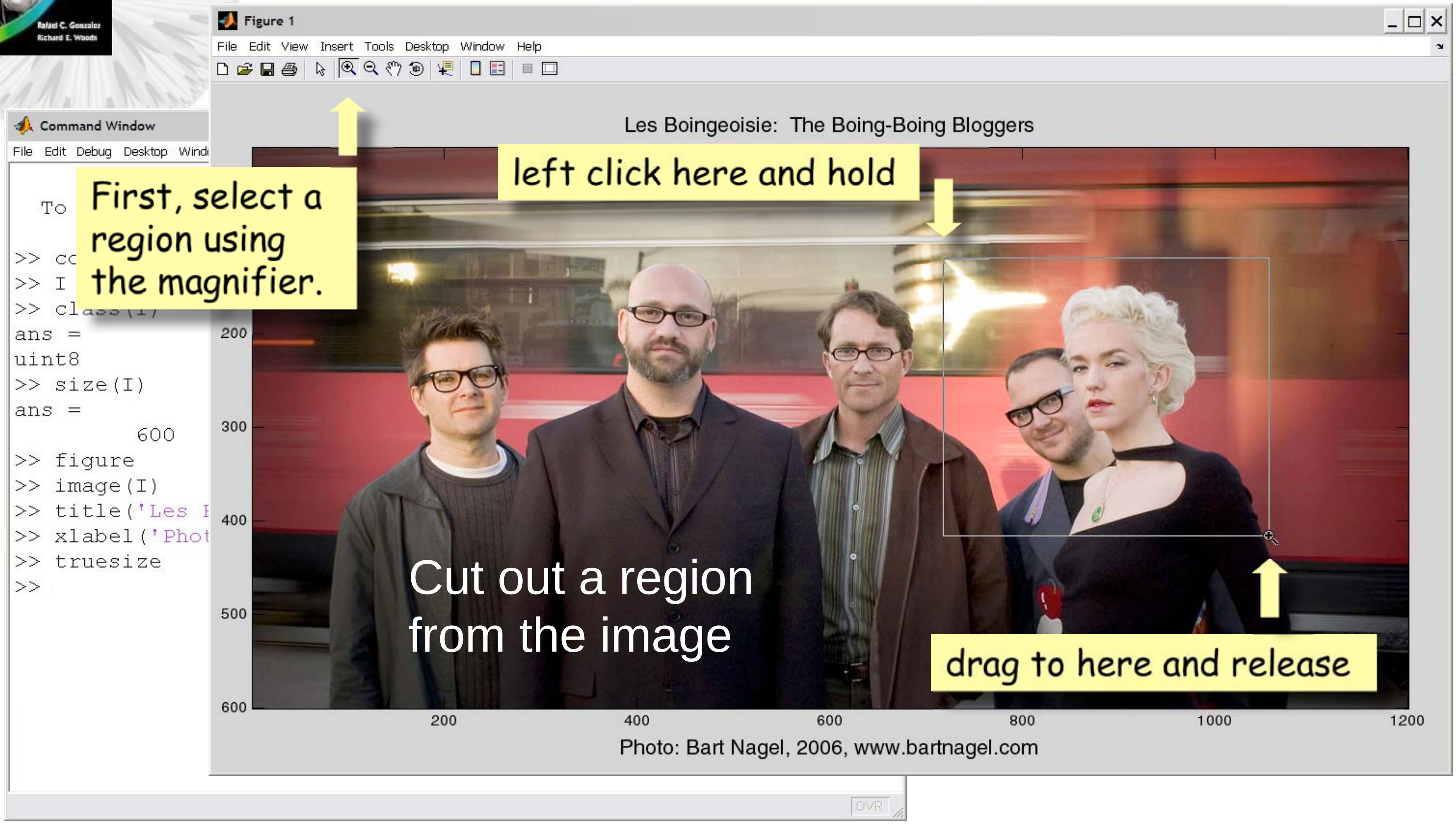


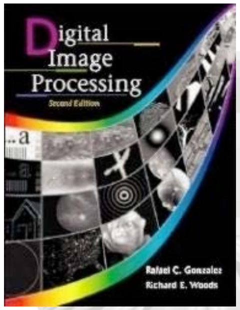
Read a Truecolor Image into Matlab

http://boingboing.net/

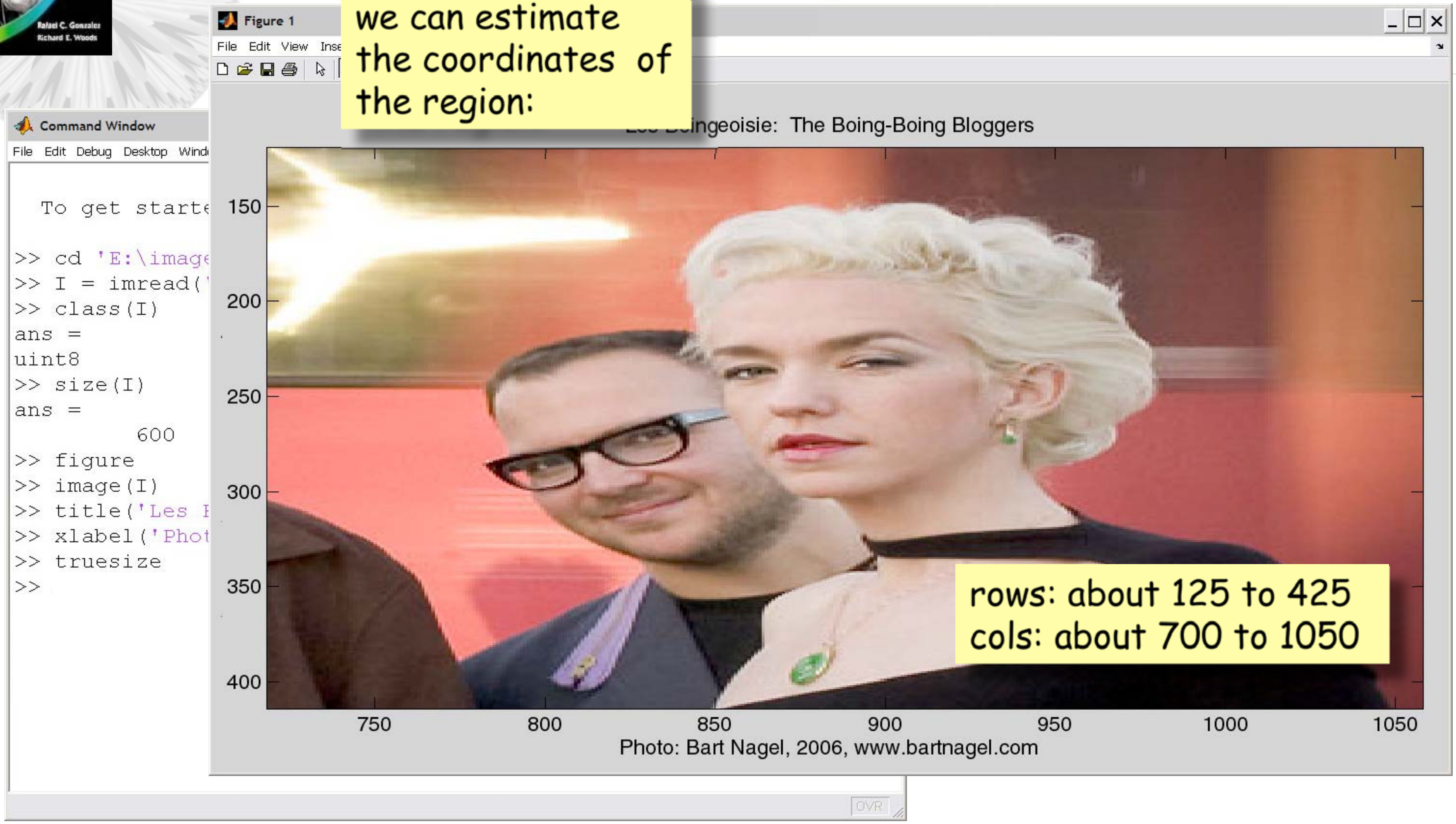


Read a Truecolor Image into Matlab



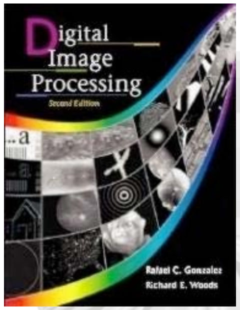


Read a Truecolor Image into Matlab

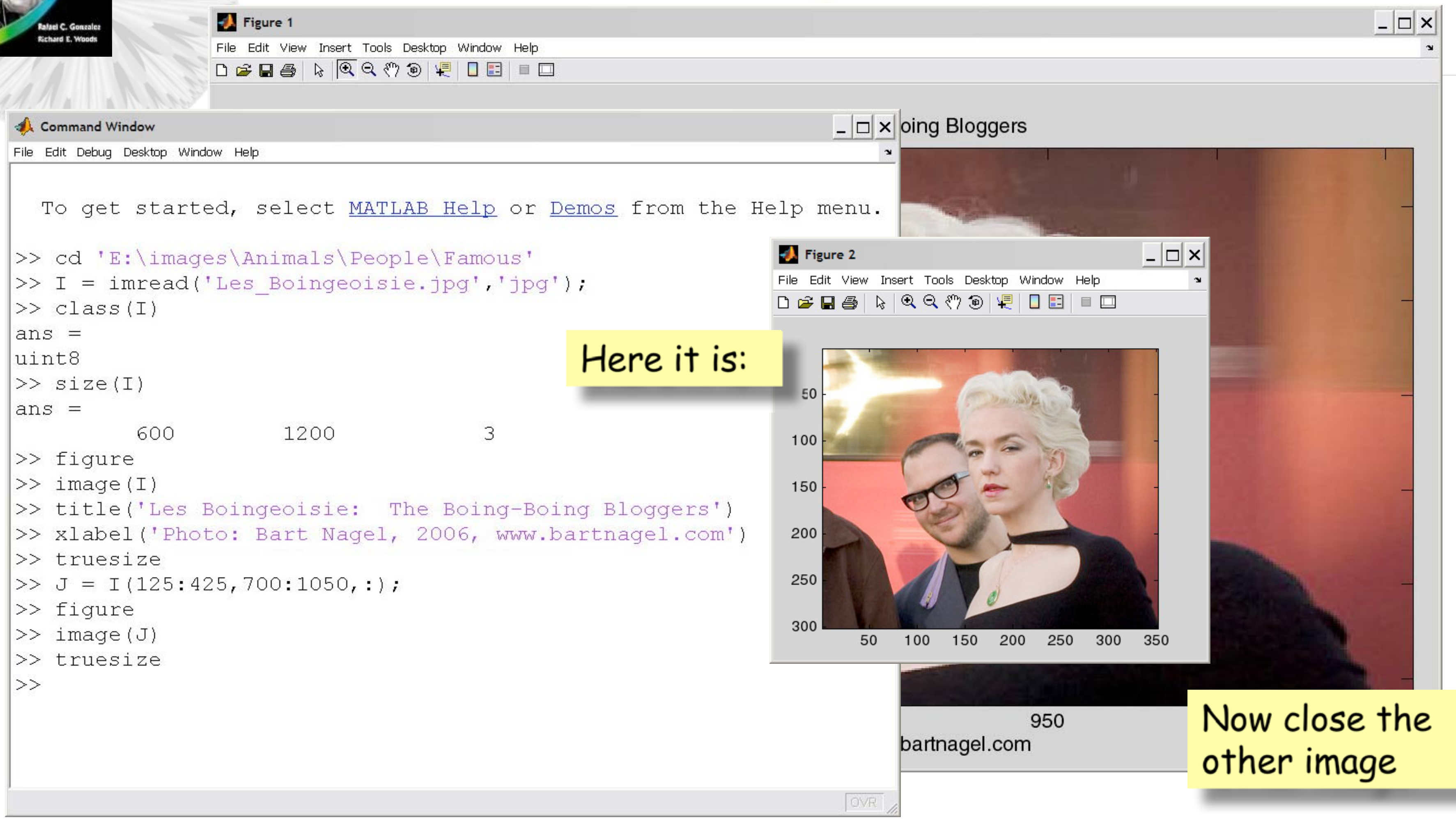


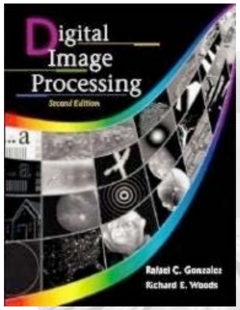
From this close-up we can estimate the coordinates of the region:

rows: about 125 to 425
cols: about 700 to 1050

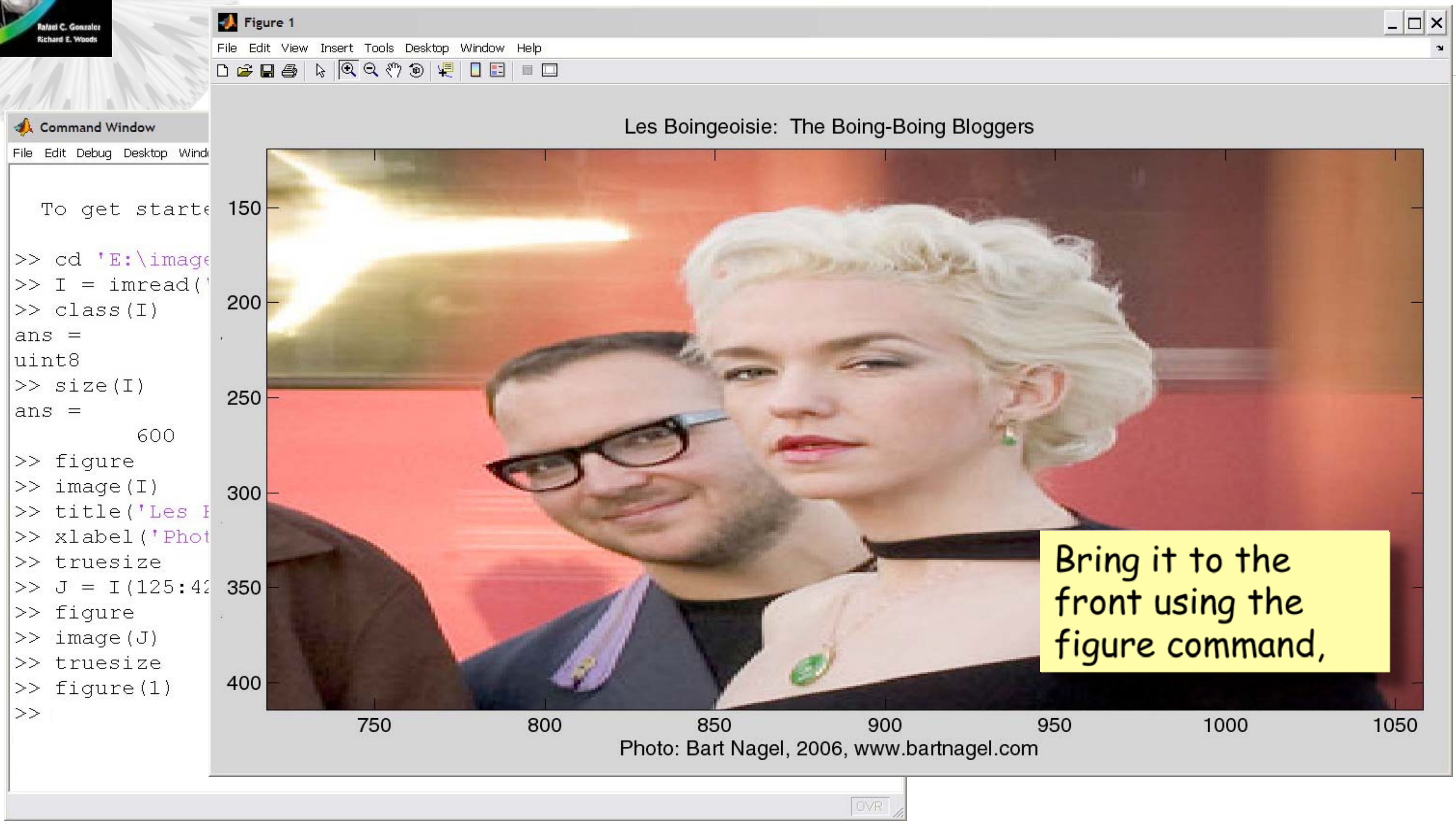


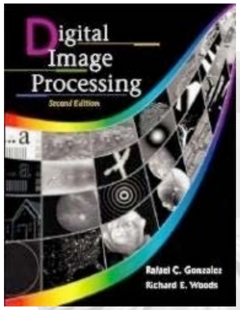
Read a Truecolor Image into Matlab





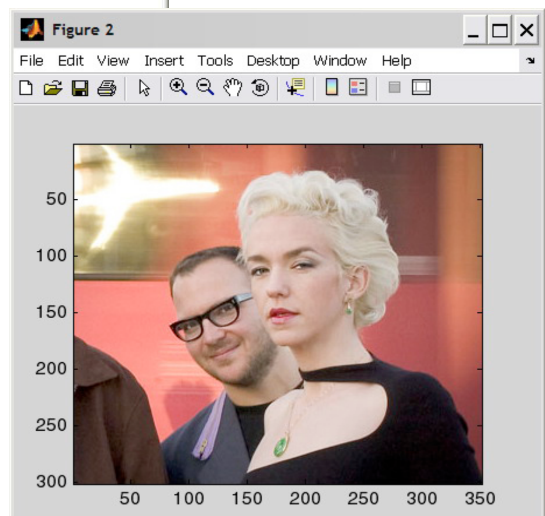
Read a Truecolor Image into Matlab



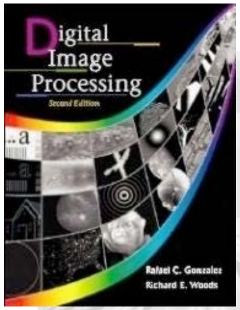


Crop the image

```
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
>> figure(1)
>> close
>>
```

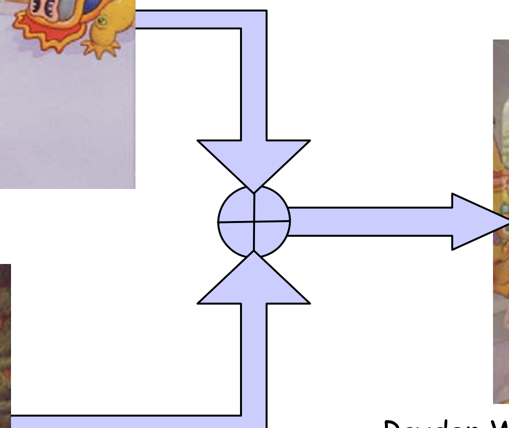


then type 'close' at the prompt.



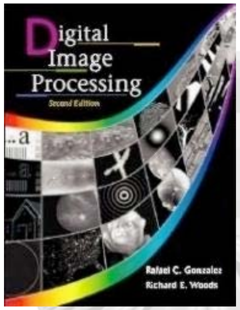
Double exposure: adding two images

Jim Woodring - Bumperillo



Rayden Woodring – The Ecstasy of Bumperillo (?)

Mark Rayden – The Ecstasy of Cecelia



Double exposure: adding two images

Example Matlab Code

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> 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')
```

Double exposure: adding two images

Example Matlab Code

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> 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')
```

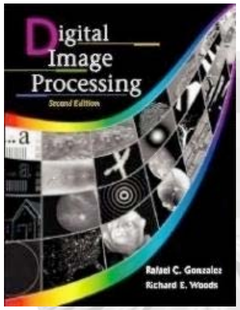
Cut a section out of the middle of the larger image the same size as the smaller image.

Double exposure: adding two images

Example Matlab Code

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> 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')
```

Note that the images are averaged, pixel-wise.



Double exposure: adding two images

Example Matlab Code

```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> 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')
```

Note the data class conversions.

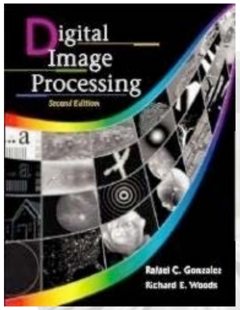
Normalize

Intensity Masking: Multiplying Two Images

Jim Woodring - Bumperillo

Mark Rayden – The Ecstasy of Cecelia

Rayden Woodring – Bumperillo Ecstasy (?)



Intensity Masking: Multiplying Two Images

Example Matlab Code

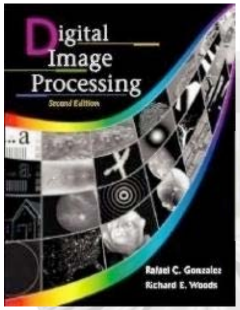
```
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> [RMR,CMR,DMR] = size(MR);
>> [RJW,CJW,DJW] = size(JW);
>> rb = round((RJW-RMR)/2);
>> cb = round((CJW-CMR)/2);
>> JWplusMR = uint8((double(JW(rb:(rb+RMR-1),cb:(cb+CMR-1),:))+double(MR))/2);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

1. Normalize to 1

2. Real number

3. Real number

4. Back to integer



Intensity Masking: Multiplying 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')
```

Note that the images are multiplied, pixelwise.

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