EECE/CS 253 Image Processing

Lecture Notes: Digital Images and Matlab

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

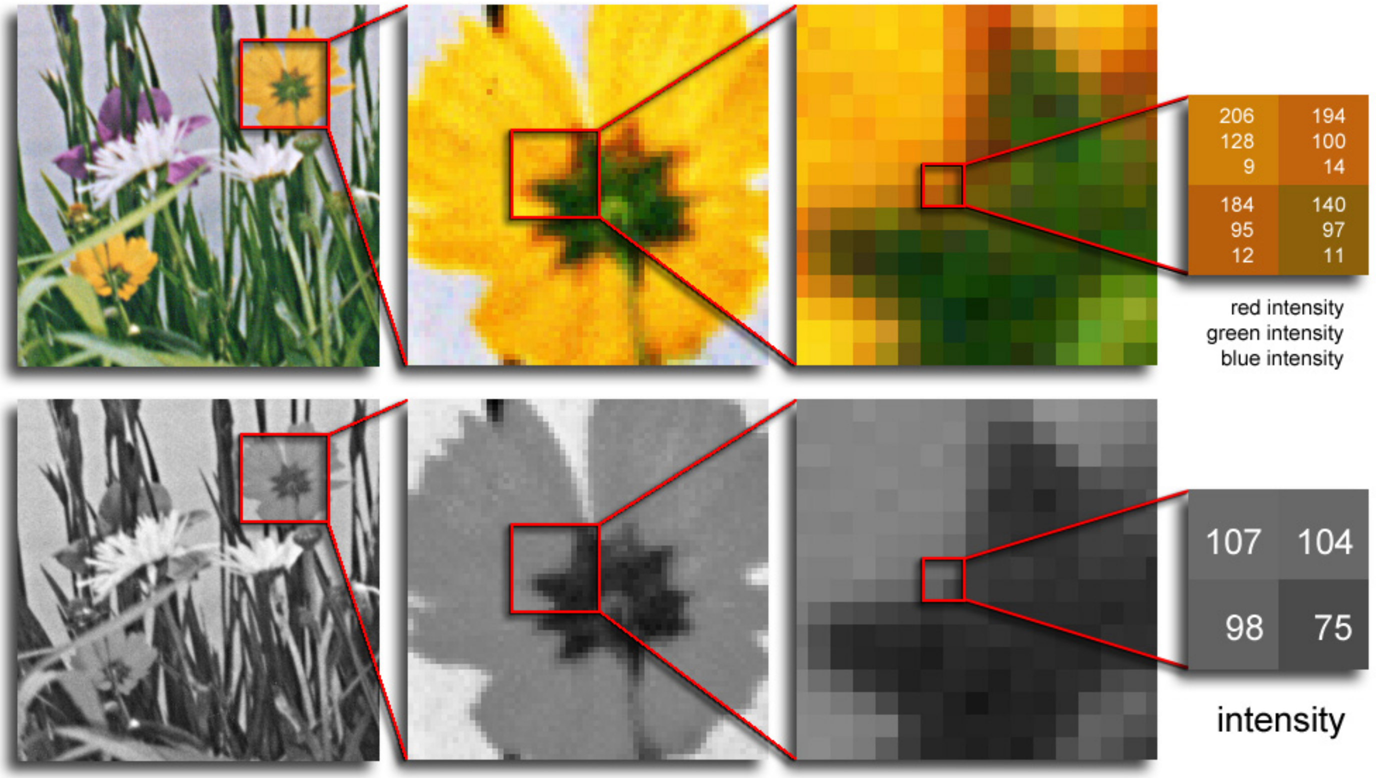
Fall Semester 2006

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*)



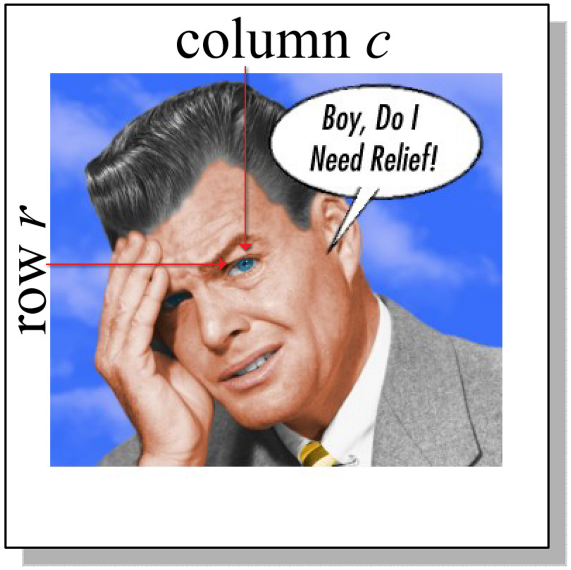
Pixels

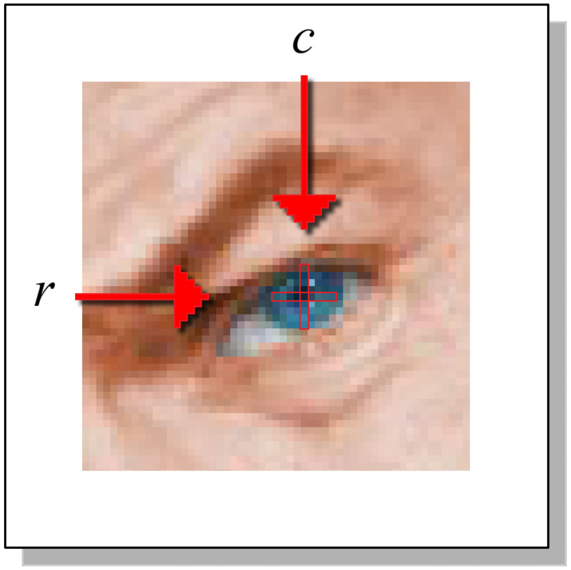
- A digital image, *I*, is a mapping from a 2D grid of uniformly spaced discrete points, $\{p = (r,c)\}$, into a set of positive integer values, $\{I(\,p)\}$, or a set of vector values, *e.g.*, $\{[R\ G\ B]^{\mathsf{T}}(p)\}$.
- At each column location in each row of *I* there is a value.
- The pair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

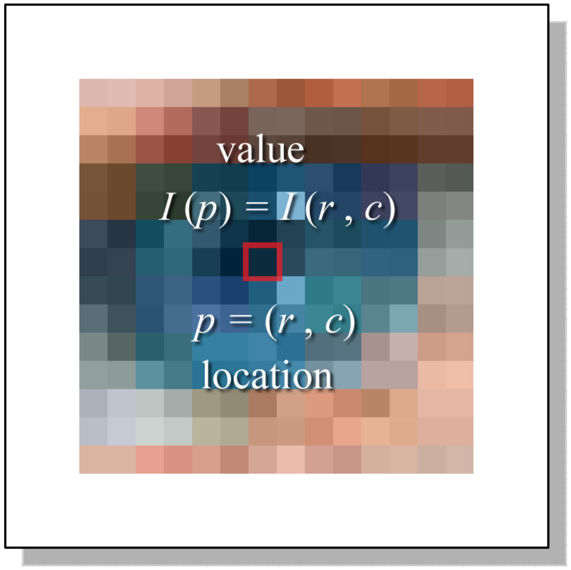
Pixels

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

Pixels







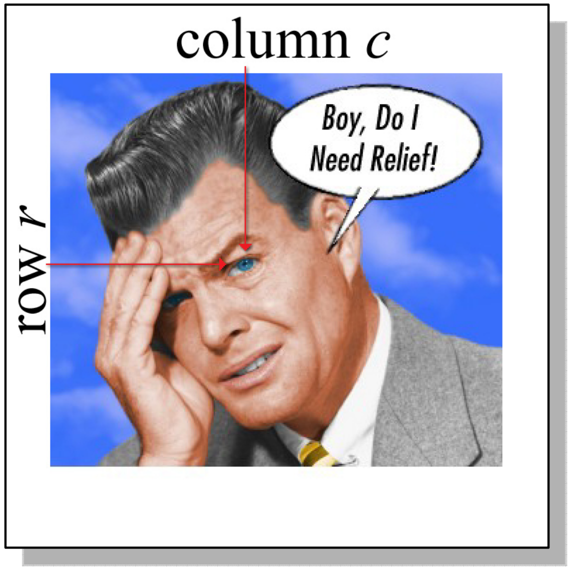
Pixel Location: $p = (r, c)$

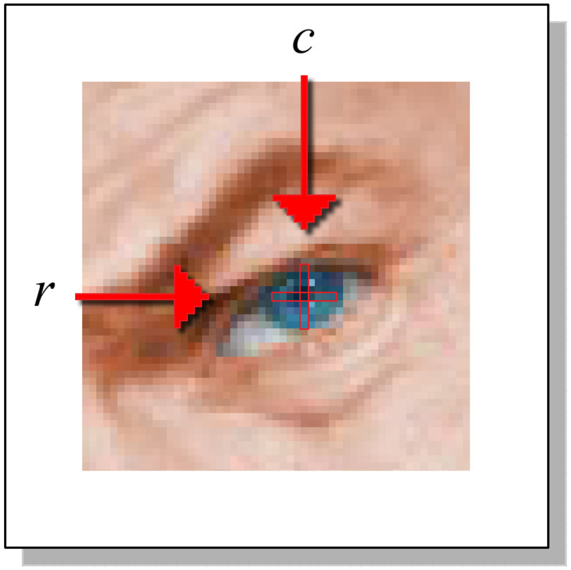
Pixel Value: $I(p) = I(r, c)$

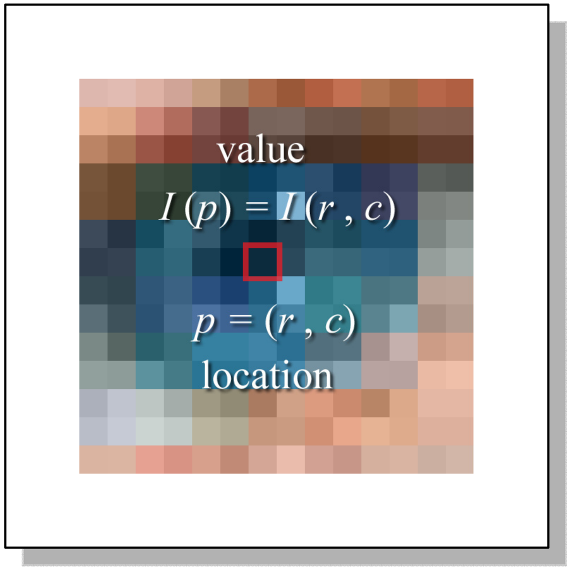
Pixel : $[p, I(p)]$

Pixels

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



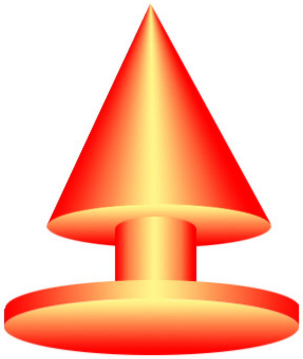




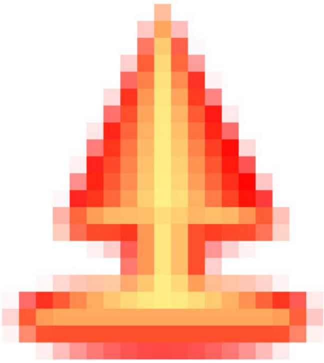
$$\begin{aligned} p &= (r,c) \\ &= (\text{row \#},\ \text{col \#}) \\ &= (272,\ 277) \end{aligned}$$

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

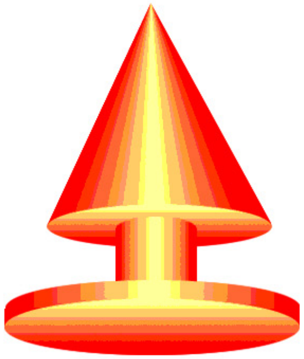
Sampling and Quantization



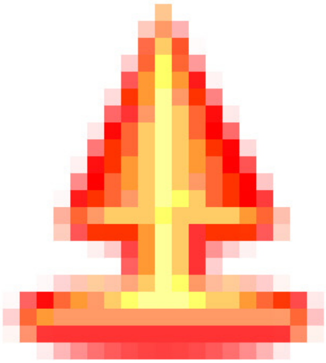
real image



sampled

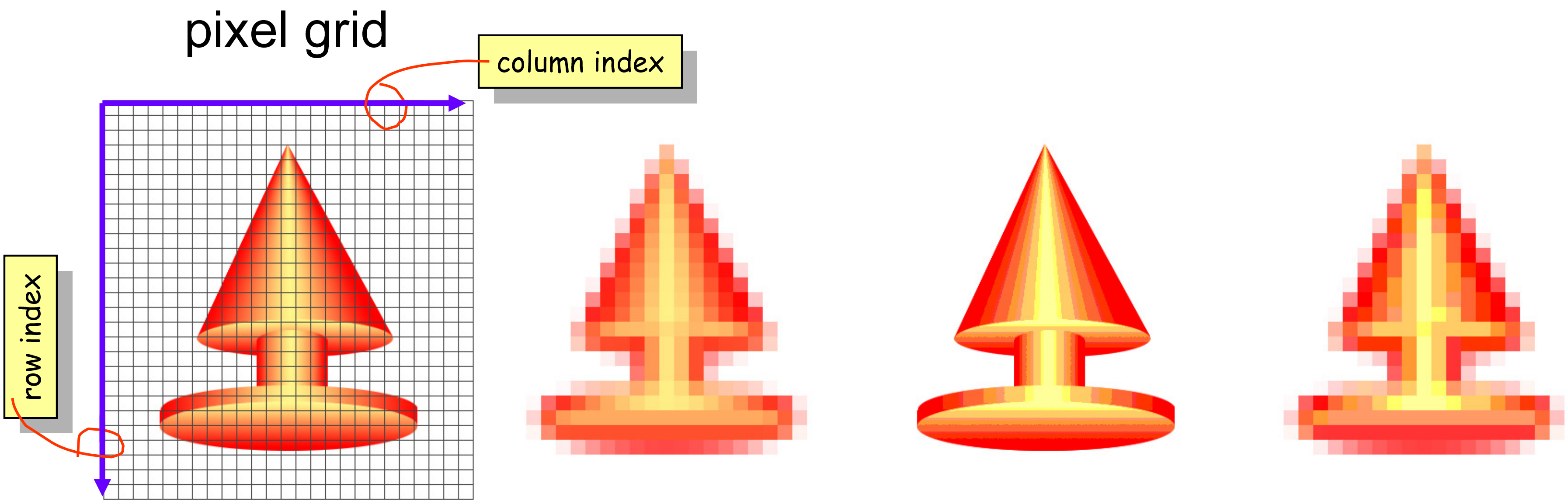


quantized



sampled & quantized

Sampling and Quantization



real image

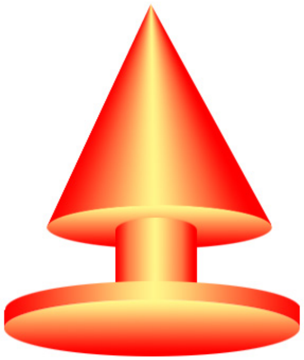
sampled

quantized

sampled & quantized

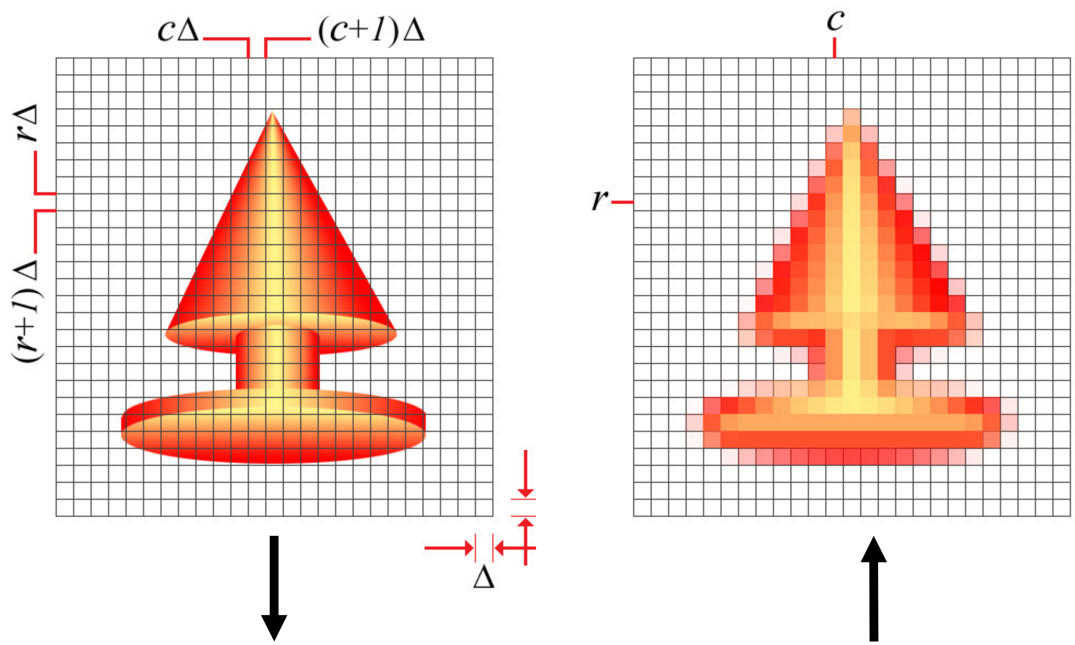
Sampling

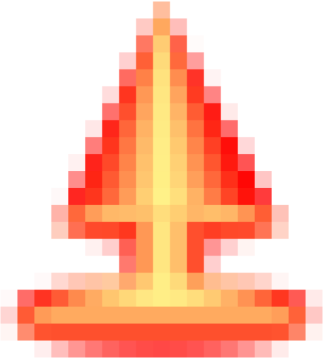
Take the average within each square.



$I_C(\rho,\chi)$

continuous image





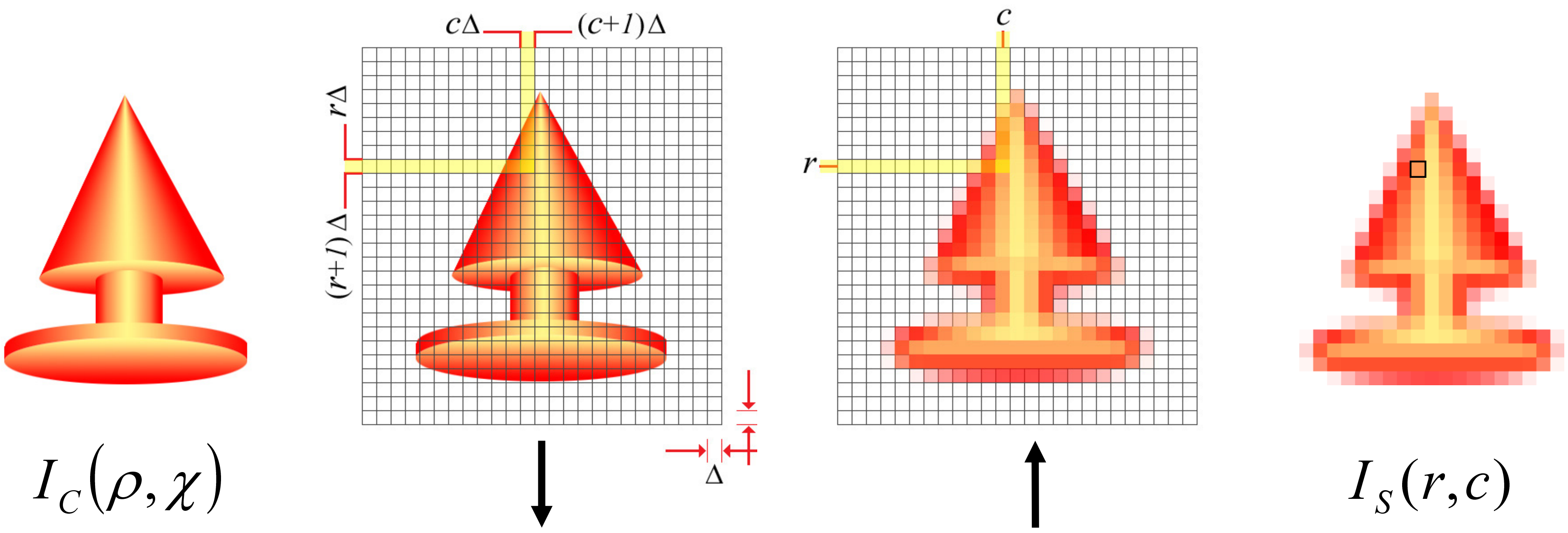
$I_S(r,c)$

sampled image

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

Sampling

Take the average within each square.



$I_C(\rho,\chi)$

continuous image

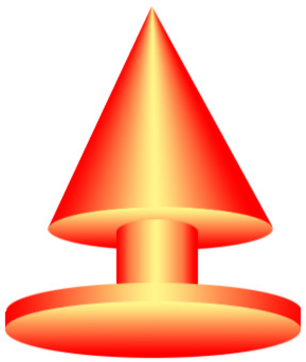
$$I_S(r,c)=\frac{1}{\Delta^2}\int_{r\Delta}^{(r+1)\Delta}\int_{c\Delta}^{(c+1)\Delta}I_C(\rho,\chi)\delta\rho\delta\chi$$

$I_S(r,c)$

sampled image

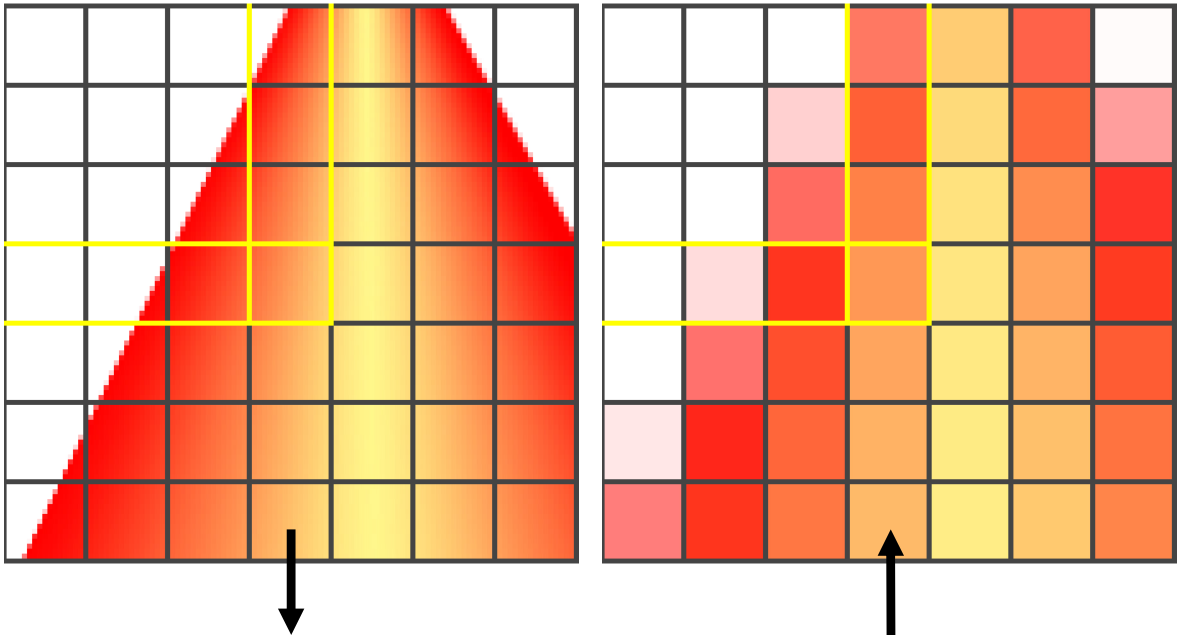
Sampling

Take the average within each square.



$I_C(\rho,\chi)$

continuous image



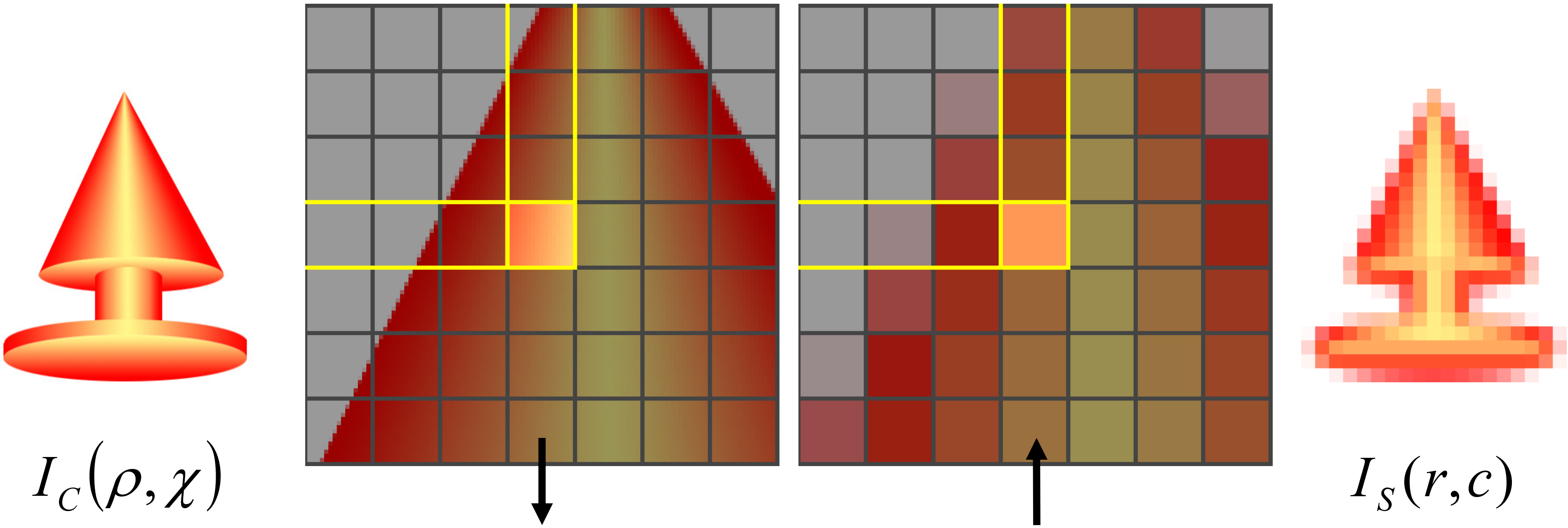
$$I_S(r,c)=\frac{1}{\Delta^2}\int_{r\Delta}^{(r+1)\Delta}\int_{c\Delta}^{(c+1)\Delta} I_C(\rho,\chi)\delta\rho\delta\chi$$

$I_S(r,c)$

sampled image

Sampling

Take the average within each square.



$I_C(\rho,\chi)$

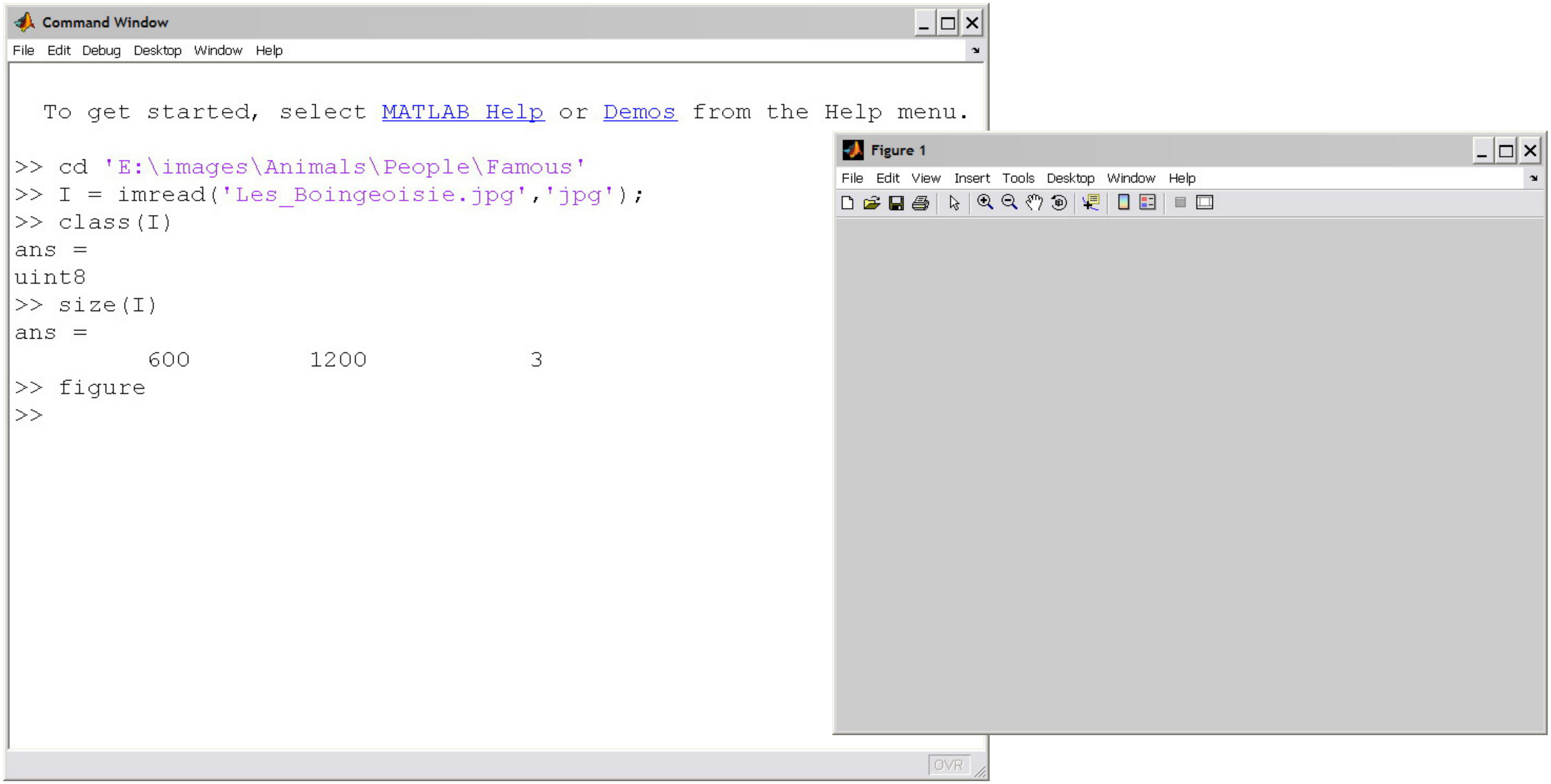
continuous image

$I_S(r,c)$

sampled image

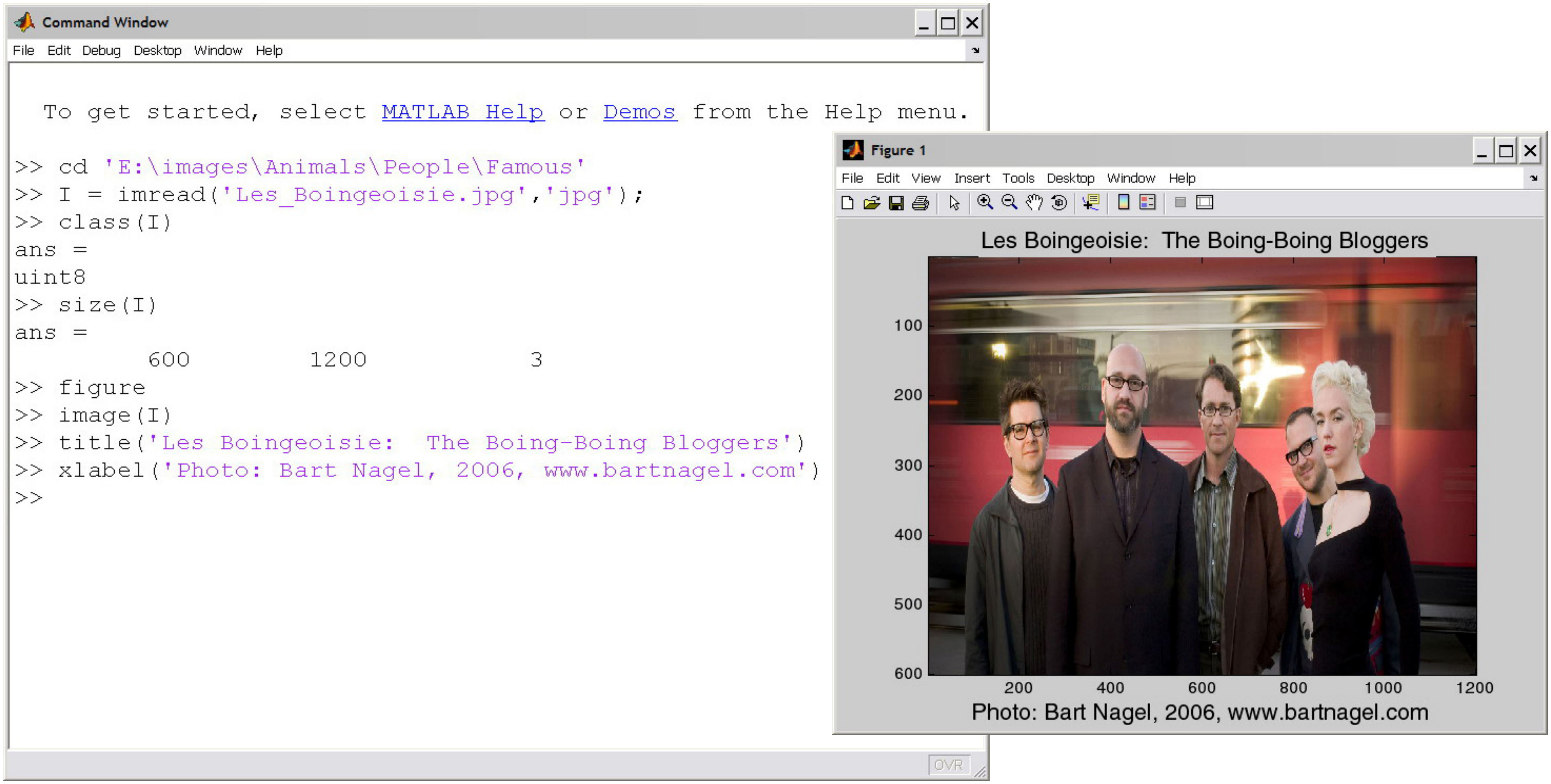
$$I_S(r,c)=\frac{1}{\Delta^2}\int_{r\Delta}^{(r+1)\Delta}\int_{c\Delta}^{(c+1)\Delta} I_C(\rho,\chi)\,\delta\rho\,\delta\chi$$

Read a Truecolor Image into Matlab


Command Window
File Edit Debug Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
OVR

```
  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
>>
```

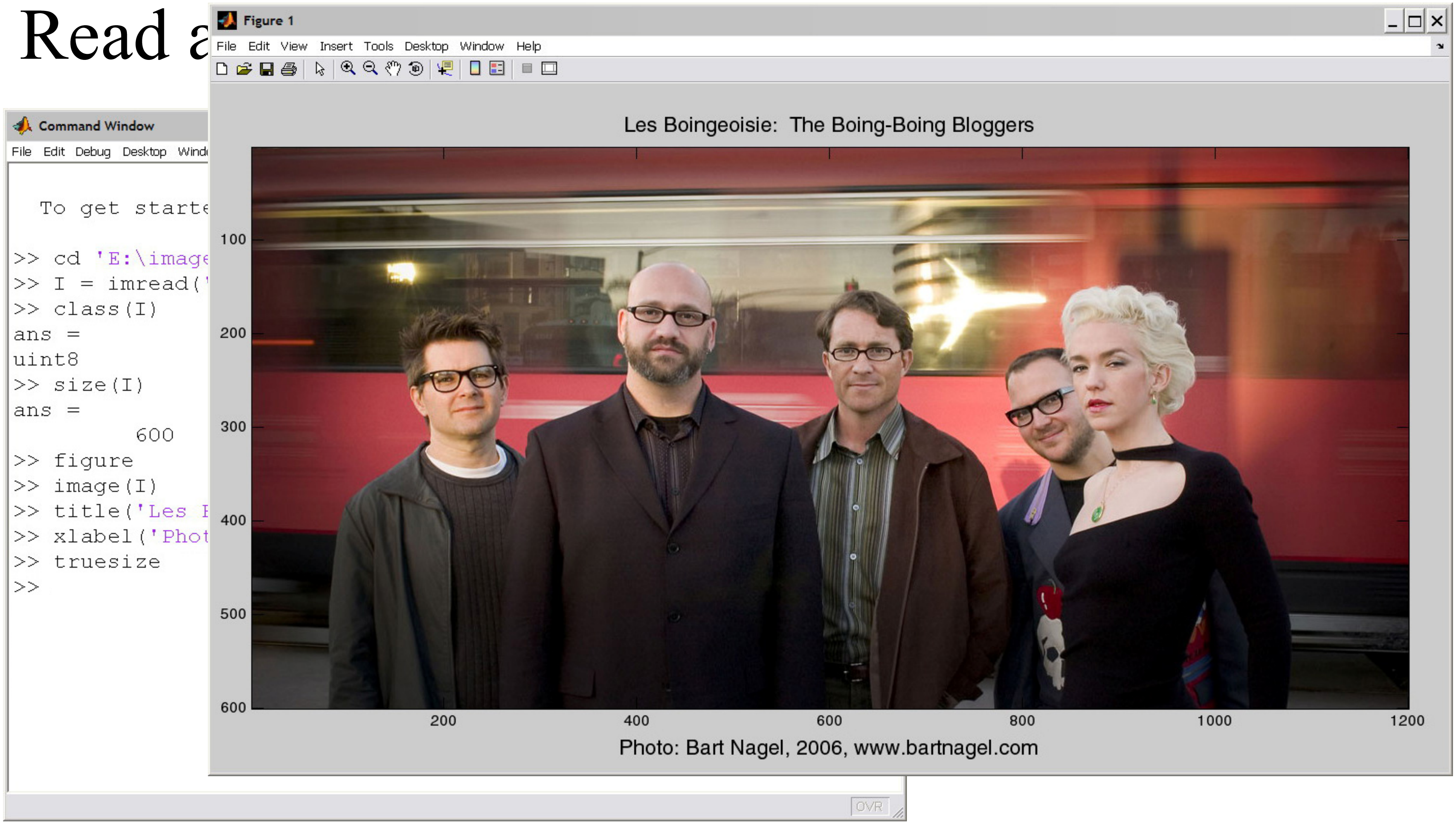
Read a Truecolor Image into Matlab



Command Window
File Edit Debug Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
Les Boingeoisie: The Boing-Boing Bloggers
100
200
300
400
500
600
200
400
600
800
1000
1200
Photo: Bart Nagel, 2006, www.bartnagel.com
OVR

```
  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

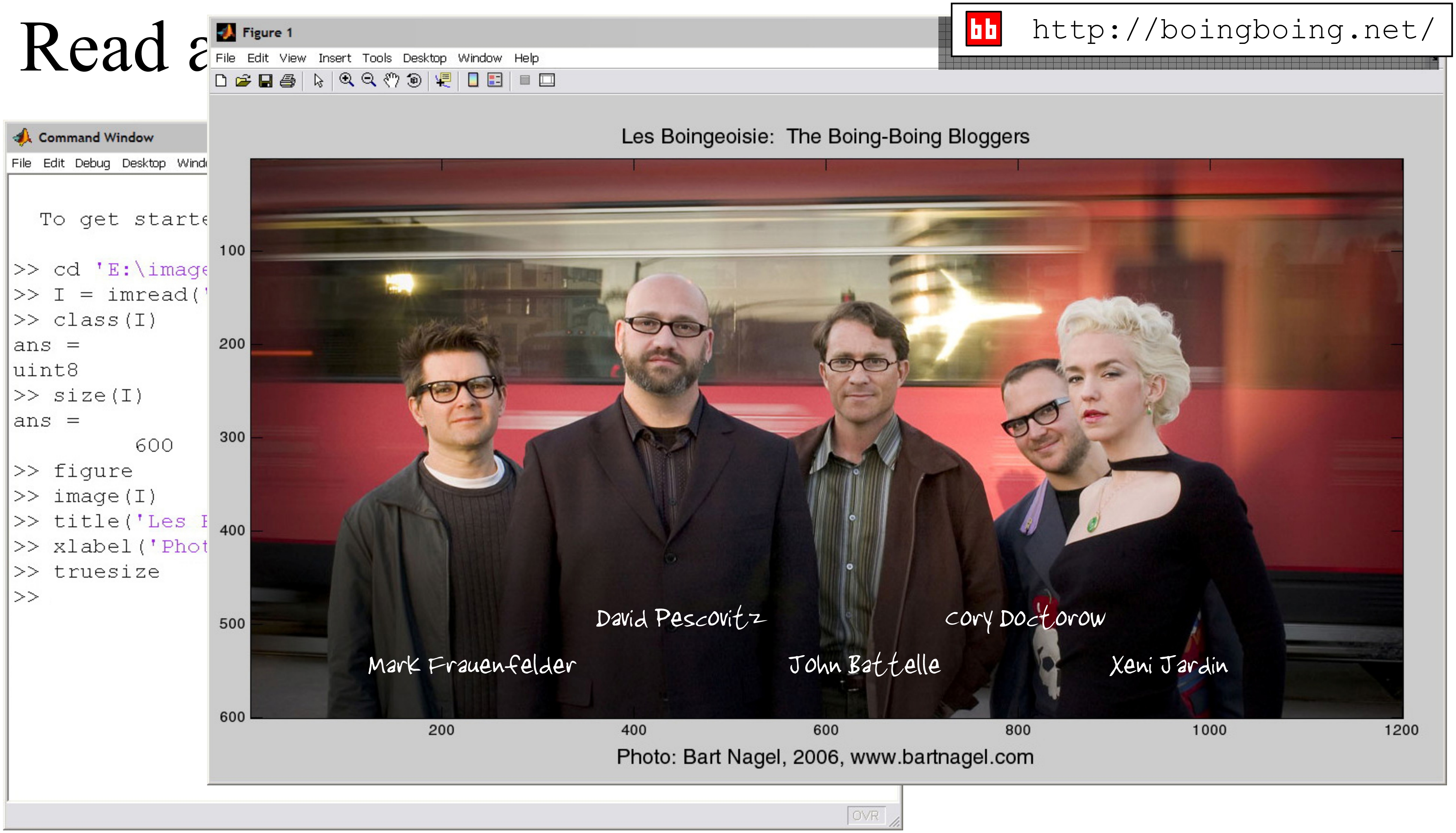

Command Window
File Edit Debug Desktop Wind
Figure 1
File Edit View Insert Tools Desktop Window Help
100
200
300
400
500
600
200
400
600
800
1000
1200
OVR

```
  To get starte

>> cd 'E:\image
>> I = imread('
>> class(I)
ans =
uint8
>> size(I)
ans =
         600
>> figure
>> image(I)
>> title('Les B
>> xlabel('Phot
>> truesize
>>
```

Les Boingeoisie: The Boing-Boing Bloggers

Photo: Bart Nagel, 2006, www.bartnagel.com

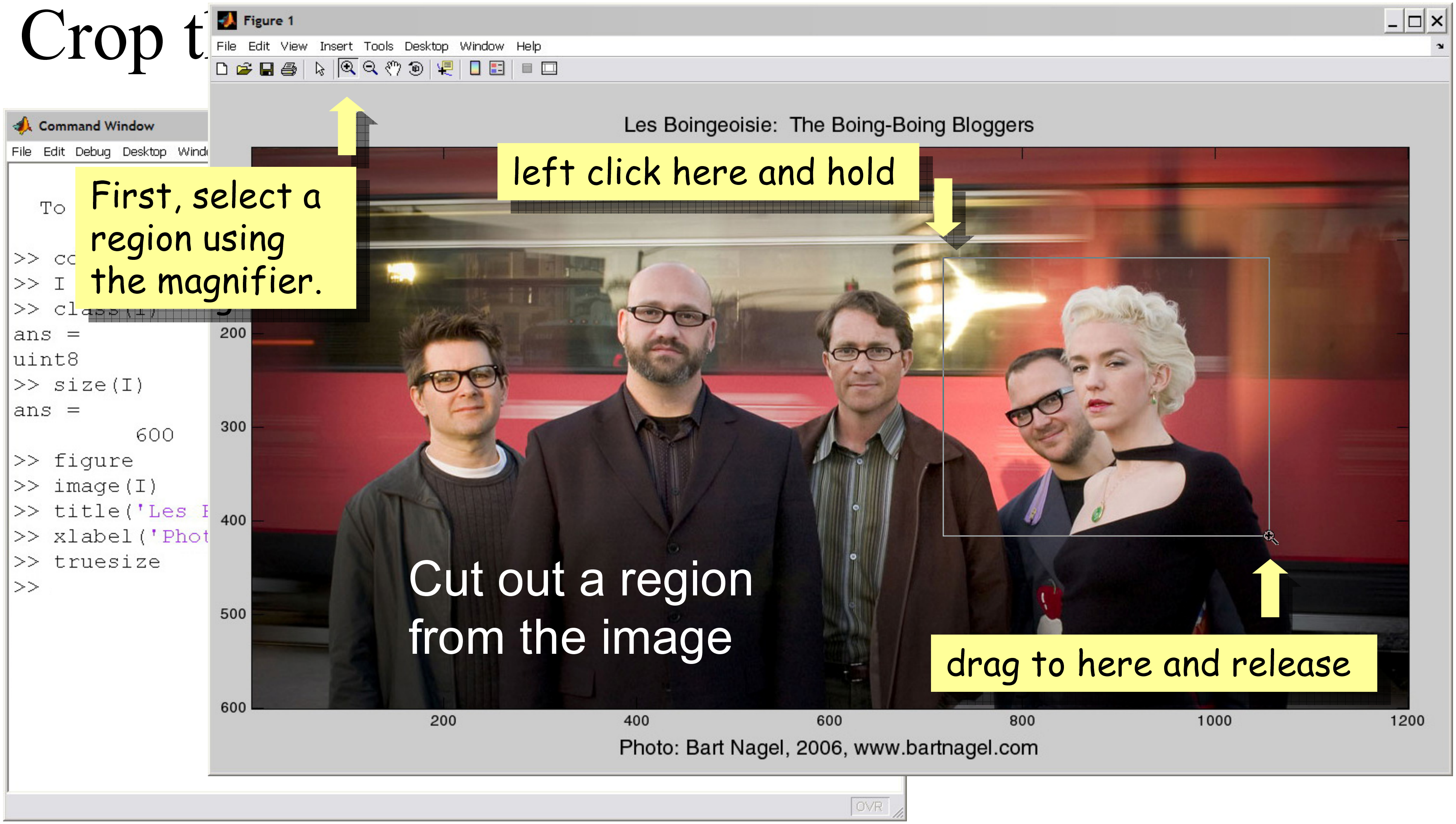
Read a



```
 To get starte

>> cd 'E:\image
>> I = imread('
>> class(I)
ans =
uint8
>> size(I)
ans =
         600
>> figure
>> image(I)
>> title('Les B
>> xlabel('Phot
>> truesize
>>
```

Command Window
File Edit Debug Desktop Wind
OVR
Figure 1
File Edit View Insert Tools Desktop Window Help
http://boingboing.net/
Les Boingeoisie: The Boing-Boing Bloggers
100
200
300
400
500
600
200
400
600
800
1000
1200
Photo: Bart Nagel, 2006, www.bartnagel.com
David Pescovitz
Cory Doctorow
Mark Frauenfelder
John Battelle
Xeni Jardin

Crop t


Command Window
File Edit Debug Desktop Wind
Figure 1
File Edit View Insert Tools Desktop Window Help
200
300
400
500
600
200
400
600
800
1000
1200
OVR

```
  To
>> co
>> I
>> class(I)
ans =
uint8
>> size(I)
ans =
        600
>> figure
>> image(I)
>> title('Les I
>> xlabel('Phot
>> truesize
>>
```

Les Boingeoisie: The Boing-Boing Bloggers

Photo: Bart Nagel, 2006, www.bartnagel.com

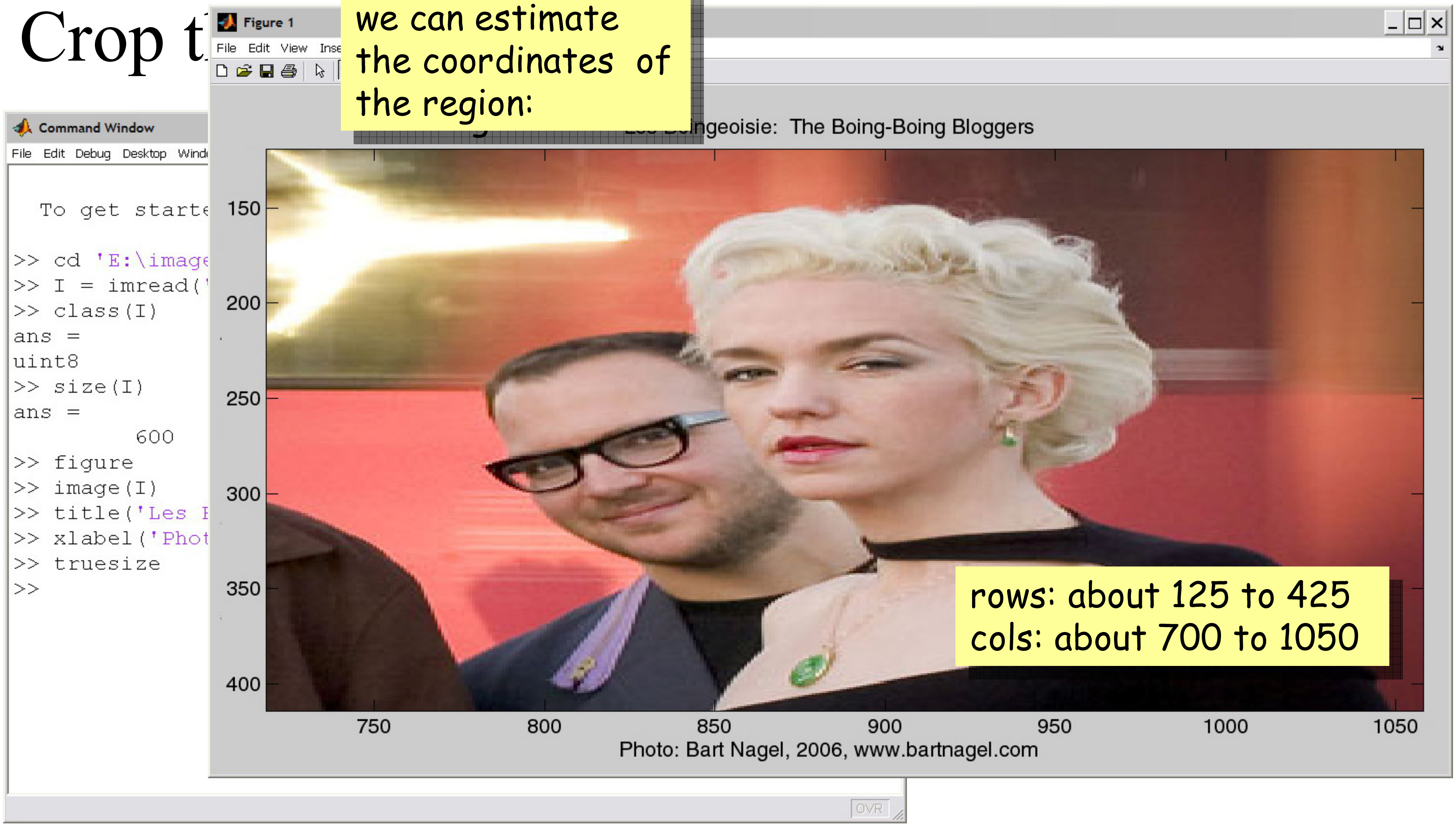
First, select a region using the magnifier.

left click here and hold

drag to here and release

Cut out a region from the image

Crop t


Command Window
File Edit Debug Desktop Wind
Figure 1
File Edit View Inse
ngeoisie: The Boing-Boing Bloggers
150
200
250
300
350
400
750
800
850
900
950
1000
1050
Photo: Bart Nagel, 2006, www.bartnagel.com
OVR

```
  To get starte

>> cd 'E:\image
>> I = imread('
>> class(I)
ans =
uint8
>> size(I)
ans =
         600
>> figure
>> image(I)
>> title('Les B
>> xlabel('Phot
>> truesize
>>
```

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

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

Crop t

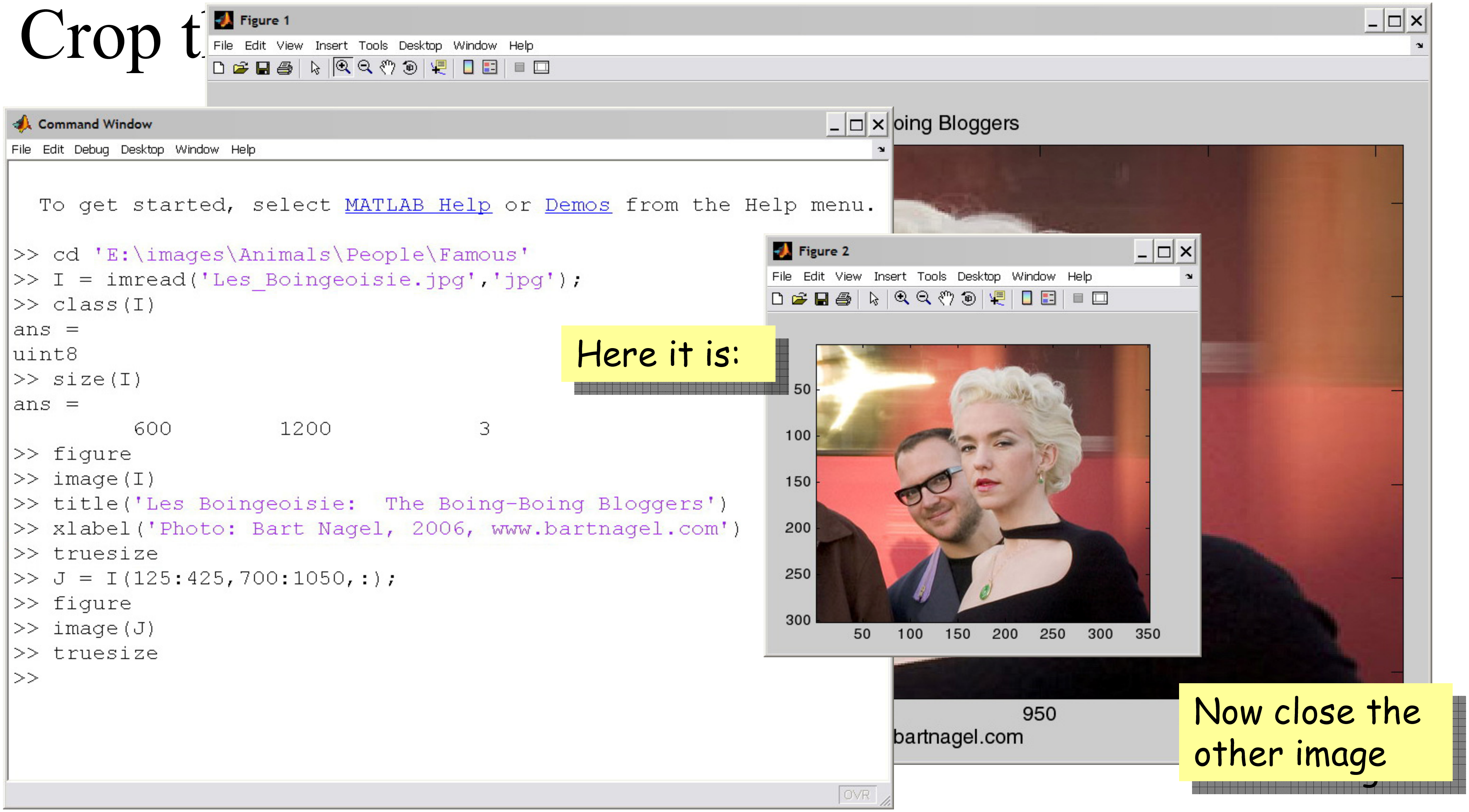


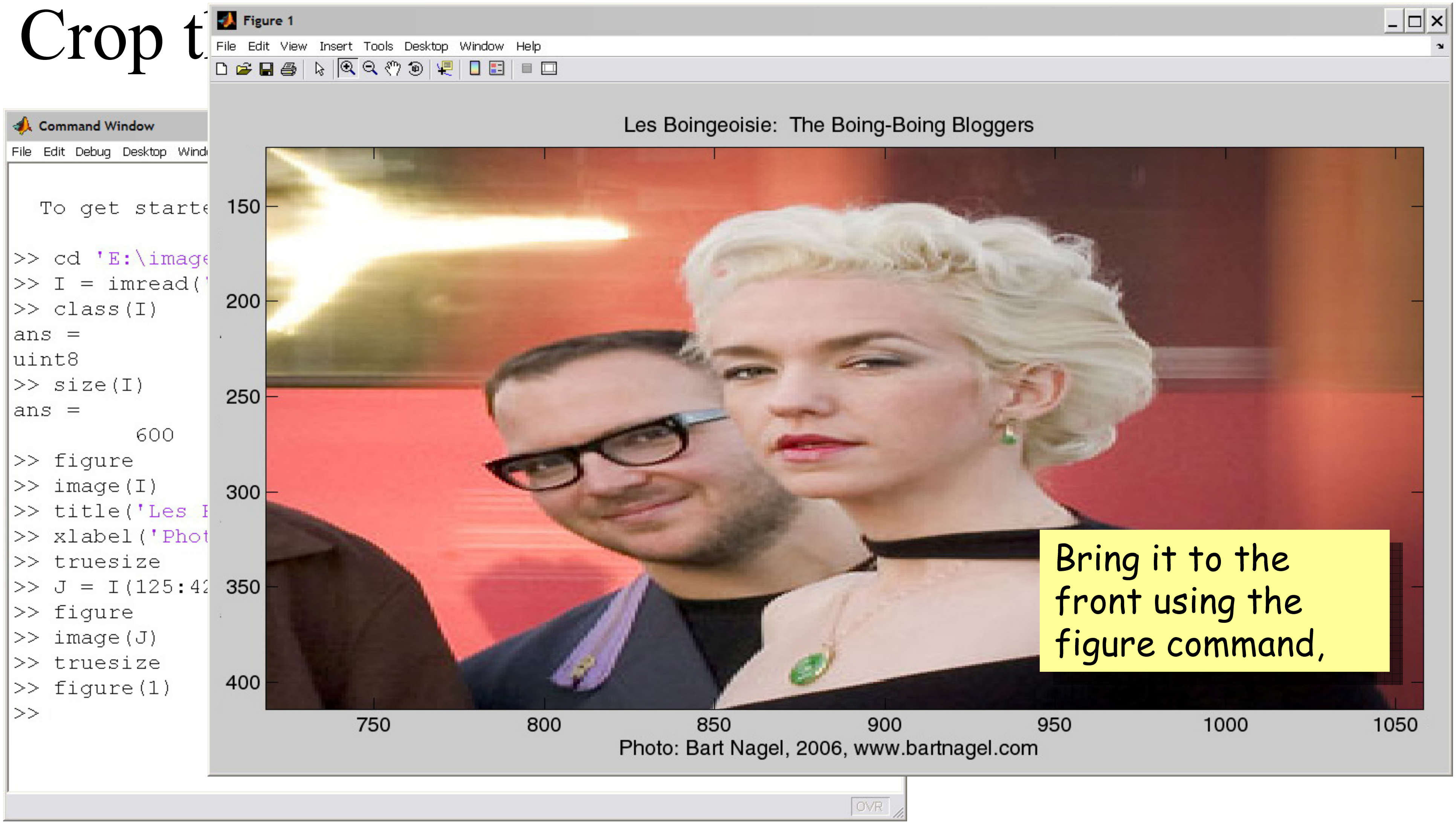
Figure 1
File Edit View Insert Tools Desktop Window Help
oing Bloggers
950
bartnagel.com
Command Window
File Edit Debug Desktop Window Help
OVR
Figure 2
File Edit View Insert Tools Desktop Window Help
50
100
150
200
250
300
50
100
150
200
250
300
350

```
  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:

Now close the other image

Crop t


Command Window
File Edit Debug Desktop Wind
Figure 1
File Edit View Insert Tools Desktop Window Help
150
200
250
300
350
400
750
800
850
900
950
1000
1050
OVR

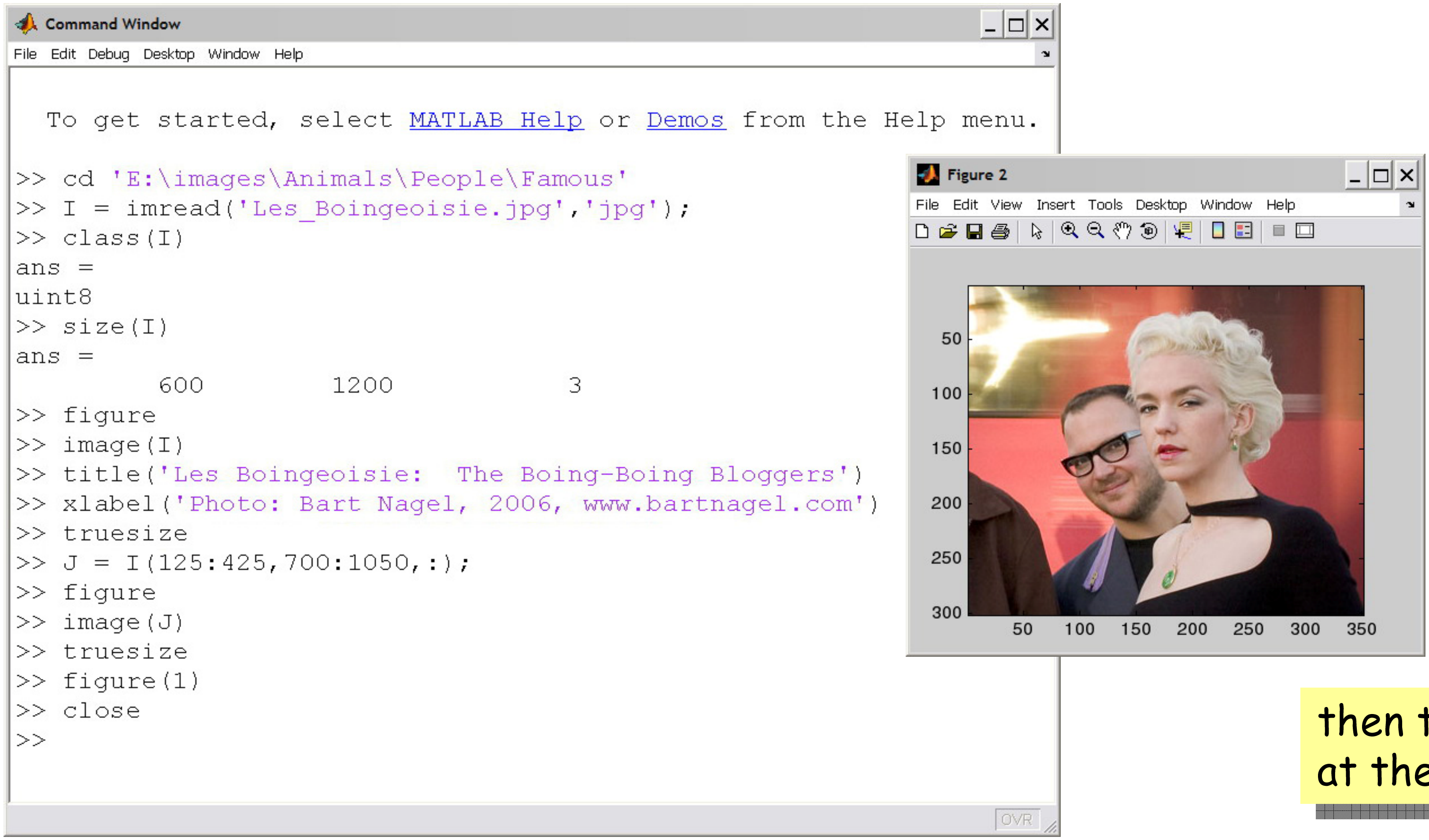
```
  To get starte

>> cd 'E:\image
>> I = imread('
>> class(I)
ans =
uint8
>> size(I)
ans =
         600
>> figure
>> image(I)
>> title('Les B
>> xlabel('Phot
>> truesize
>> J = I(125:42
>> figure
>> image(J)
>> truesize
>> figure(1)
>>
```

Les Boingeoisie: The Boing-Boing Bloggers

Photo: Bart Nagel, 2006, www.bartnagel.com

Bring it to the front using the figure command,

Crop the Image

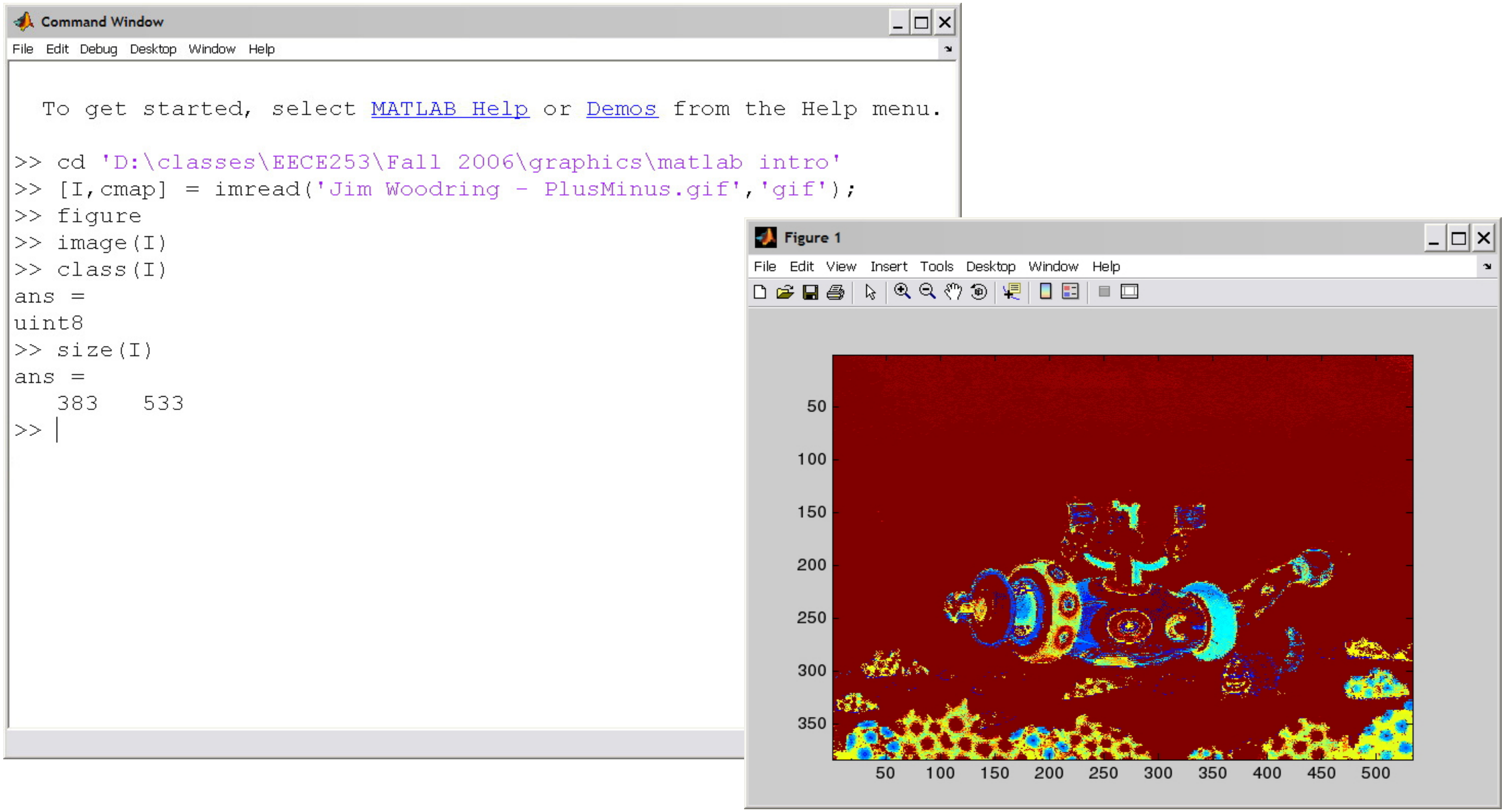


```
  To get started, select MATLAB Help or Demos from the Help menu.

>> 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
>>
```

Command Window
File Edit Debug Desktop Window Help
OVR
Figure 2
File Edit View Insert Tools Desktop Window Help
50
100
150
200
250
300
50
100
150
200
250
300
350

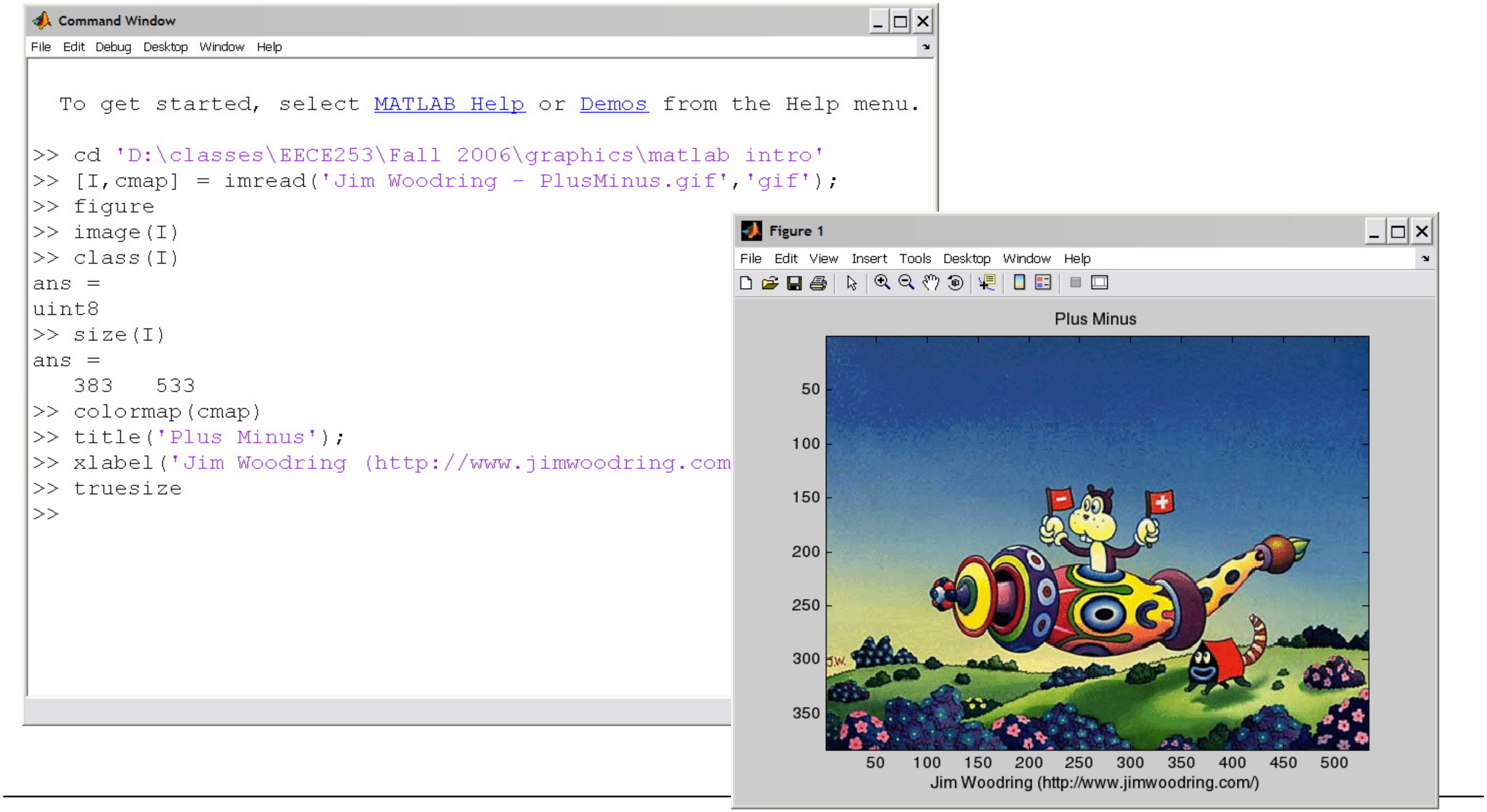
then type 'close' at the prompt.

Read a Colormapped Image into Matlab


Command Window
File Edit Debug Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
50
100
150
200
250
300
350
50
100
150
200
250
300
350
400
450
500

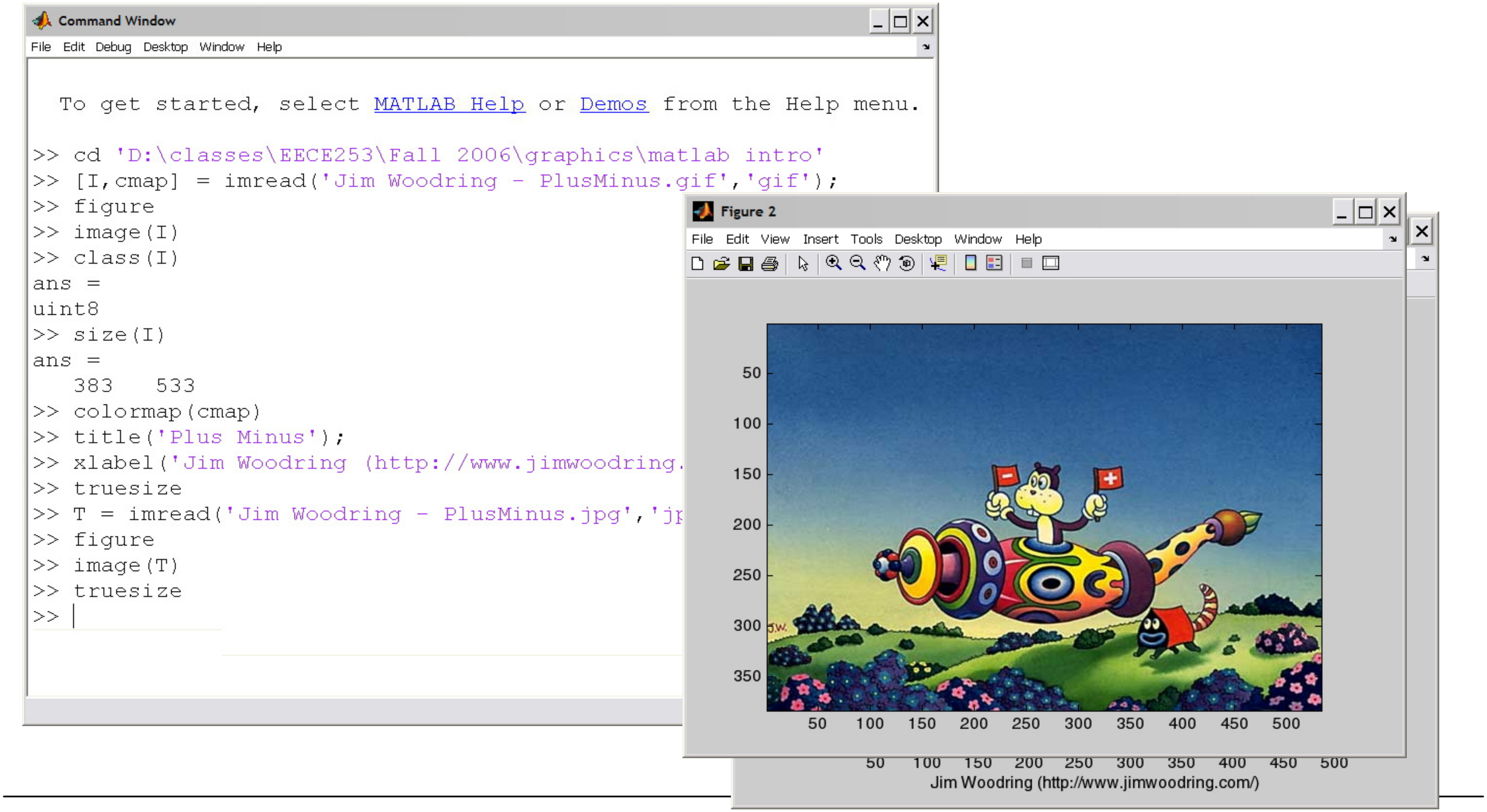
```
  To get started, select MATLAB Help or Demos from the Help menu.

>> cd 'D:\classes\EECE253\Fall 2006\graphics\matlab intro'
>> [I,cmap] = imread('Jim Woodring - PlusMinus.gif','gif');
>> figure
>> image(I)
>> class(I)
ans =
uint8
>> size(I)
ans =
   383   533
>> 
```

Read a Colormapped Image into Matlab


Command Window
File Edit Debug Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
Plus Minus
50
100
150
200
250
300
350
50 100 150 200 250 300 350 400 450 500
Jim Woodring (http://www.jimwoodring.com/)

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

>> cd 'D:\classes\EECE253\Fall 2006\graphics\matlab intro'
>> [I,cmap] = imread('Jim Woodring - PlusMinus.gif','gif');
>> figure
>> image(I)
>> class(I)
ans =
uint8
>> size(I)
ans =
   383   533
>> colormap(cmap)
>> title('Plus Minus');
>> xlabel('Jim Woodring (http://www.jimwoodring.com
>> truesize
>>
```

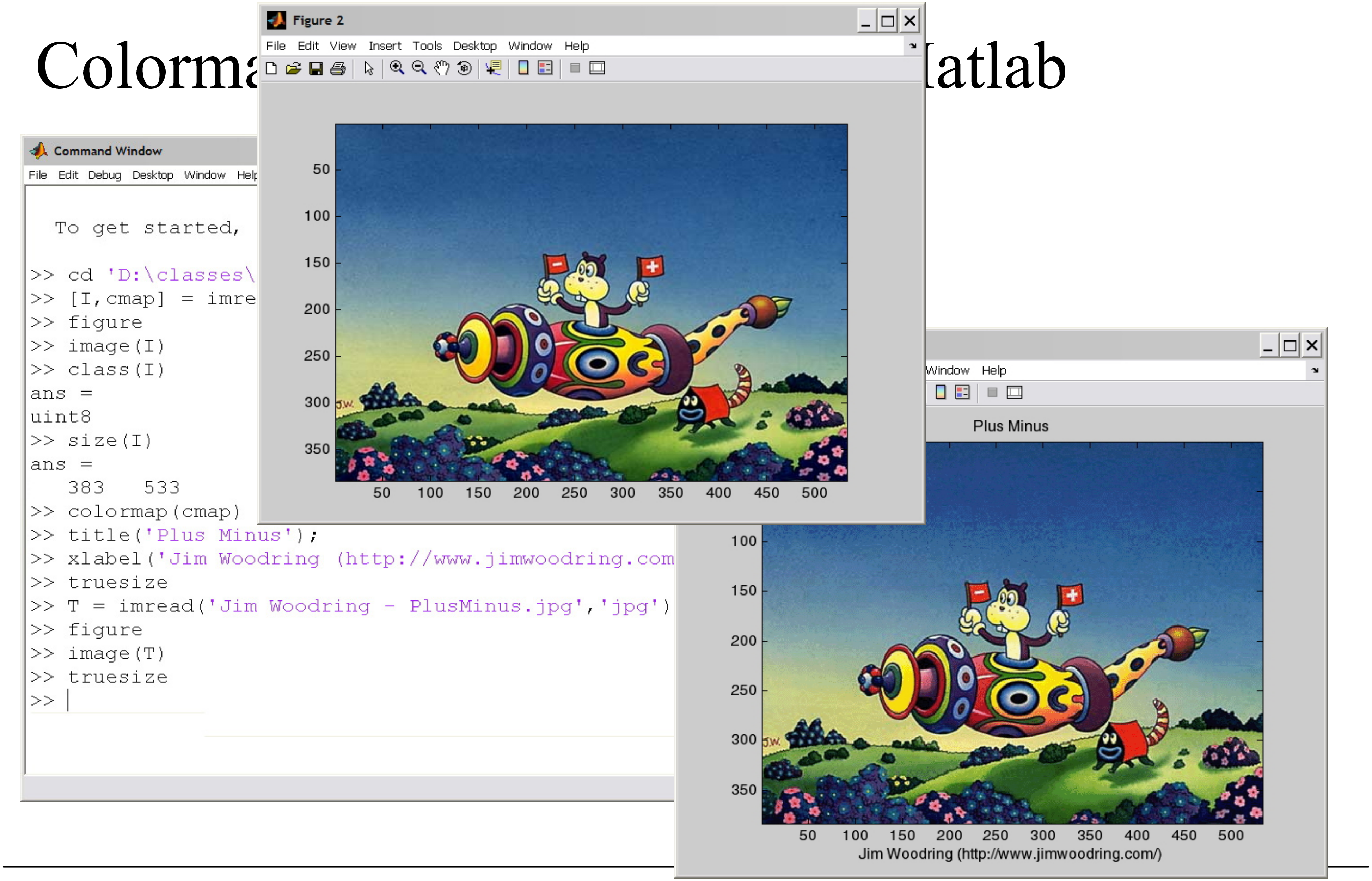
Colormapped vs. Truecolor in Matlab



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

>> cd 'D:\classes\EECE253\Fall 2006\graphics\matlab intro'
>> [I,cmap] = imread('Jim Woodring - PlusMinus.gif','gif');
>> figure
>> image(I)
>> class(I)
ans =
uint8
>> size(I)
ans =
   383   533
>> colormap(cmap)
>> title('Plus Minus');
>> xlabel('Jim Woodring (http://www.jimwoodring.
>> truesize
>> T = imread('Jim Woodring - PlusMinus.jpg','jp
>> figure
>> image(T)
>> truesize
>> |
```

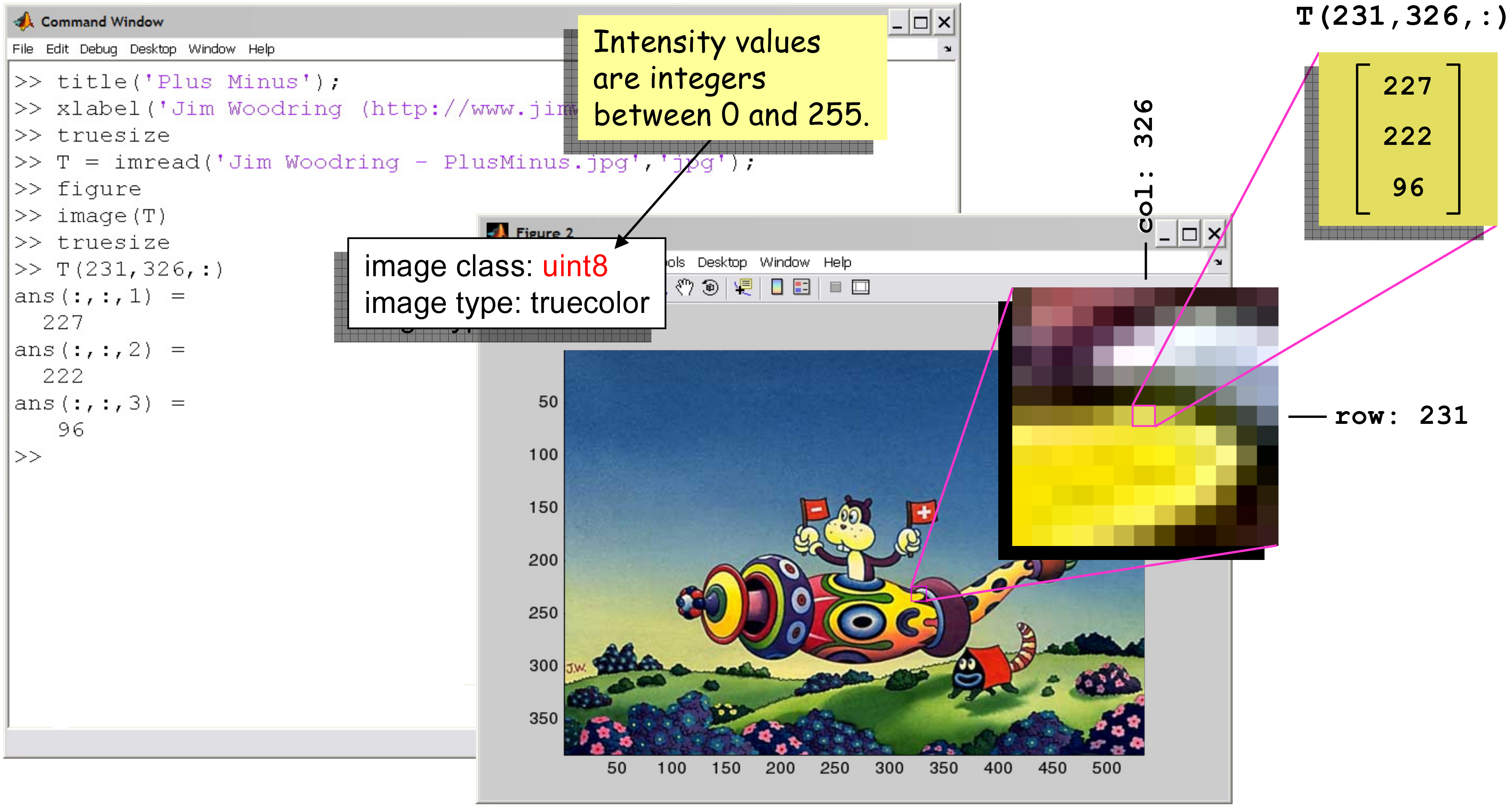
Command Window
File Edit Debug Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help
50
100
150
200
250
300
350
50 100 150 200 250 300 350 400 450 500
50 100 150 200 250 300 350 400 450 500
Jim Woodring (http://www.jimwoodring.com/)

Colorma atlab



```
  To get started,

>> cd 'D:\classes\
>> [I,cmap] = imre
>> figure
>> image(I)
>> class(I)
ans =
uint8
>> size(I)
ans =
   383   533
>> colormap(cmap)
>> title('Plus Minus');
>> xlabel('Jim Woodring (http://www.jimwoodring.com
>> truesize
>> T = imread('Jim Woodring - PlusMinus.jpg','jpg')
>> figure
>> image(T)
>> truesize
>>
```

Colormapped vs. Truecolor in Matlab



```
>> title('Plus Minus');
>> xlabel('Jim Woodring (http://www.jim
>> truesize
>> T = imread('Jim Woodring - PlusMinus.jpg','jpg');
>> figure
>> image(T)
>> truesize
>> T(231,326,:)
ans(:,:,1) =
   227
ans(:,:,2) =
   222
ans(:,:,3) =
    96
>>
```

Intensity values are integers between 0 and 255.

image class: uint8
image type: truecolor

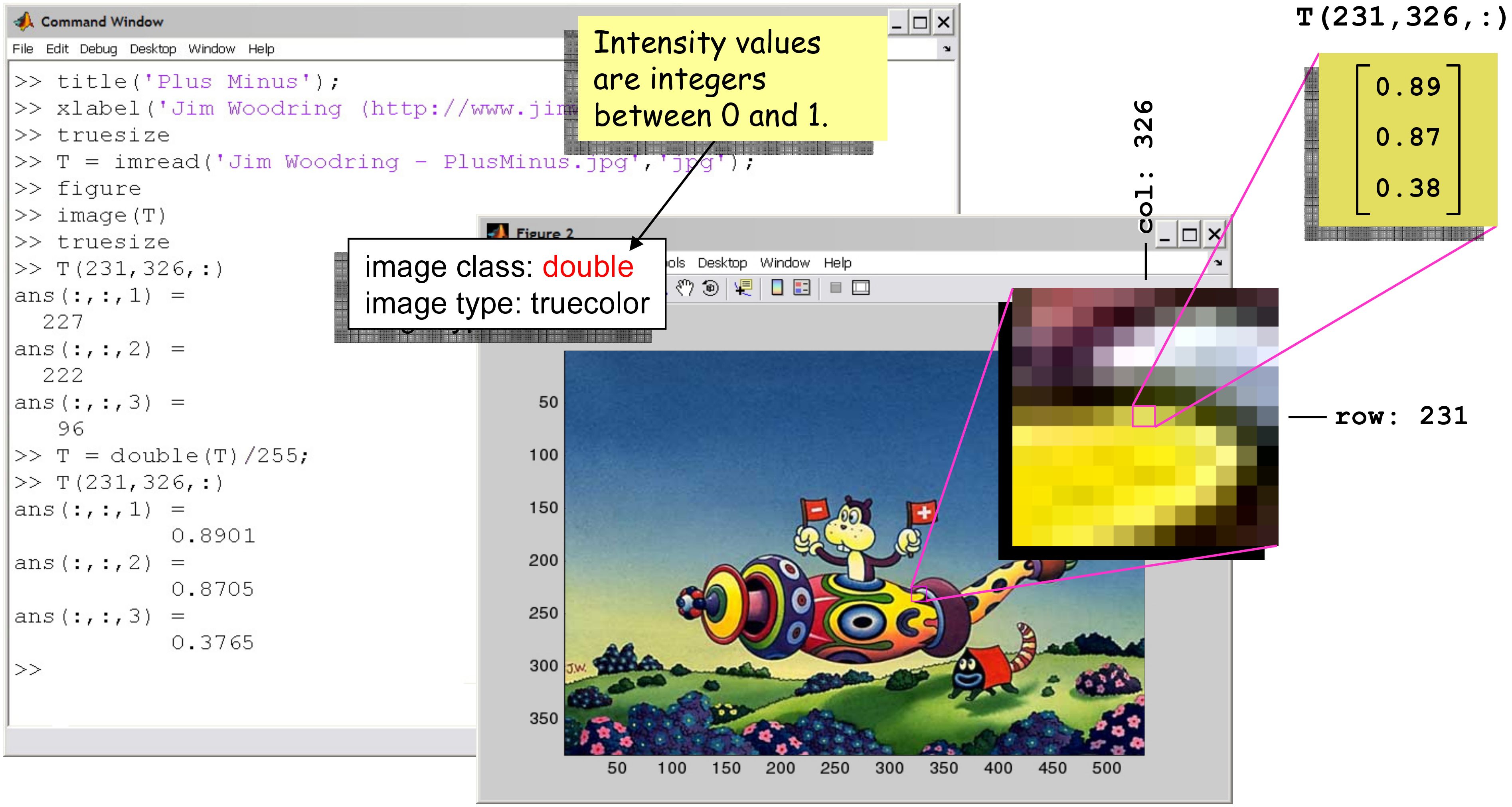
T(231,326,:)

$$\begin{bmatrix} 227 \\ 222 \\ 96 \end{bmatrix}$$

col: 326

row: 231

Colormapped vs. Truecolor in Matlab



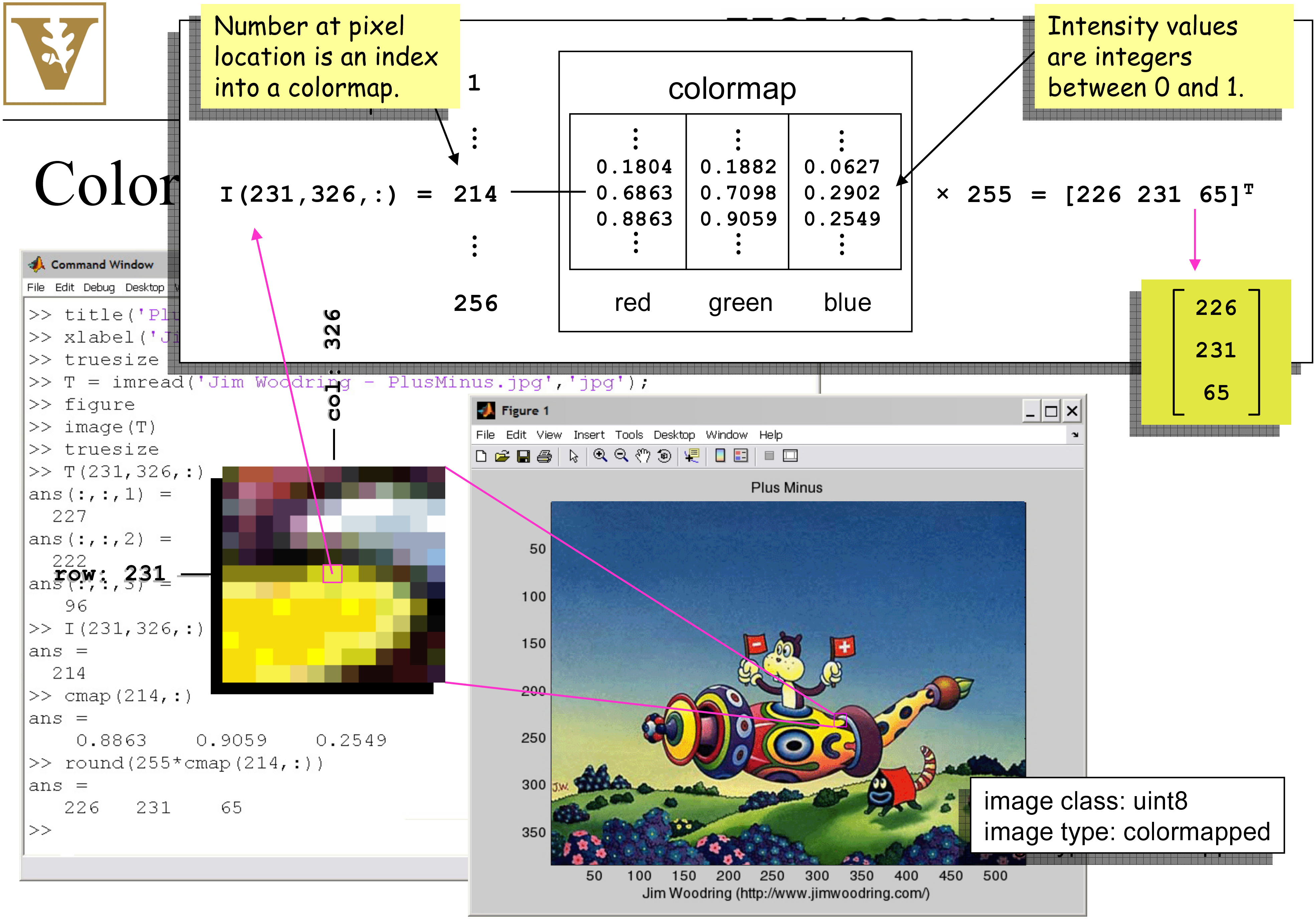
```
>> title('Plus Minus');
>> xlabel('Jim Woodring (http://www.jim
>> truesize
>> T = imread('Jim Woodring - PlusMinus.jpg','jpg');
>> figure
>> image(T)
>> truesize
>> T(231,326,:)
ans(:,:,1) =
   227
ans(:,:,2) =
   222
ans(:,:,3) =
    96
>> T = double(T)/255;
>> T(231,326,:)
ans(:,:,1) =
            0.8901
ans(:,:,2) =
            0.8705
ans(:,:,3) =
            0.3765
>>
```

Intensity values are integers between 0 and 1.

image class: double
image type: truecolor

$$\begin{bmatrix} 0.89 \\ 0.87 \\ 0.38 \end{bmatrix}$$

Color



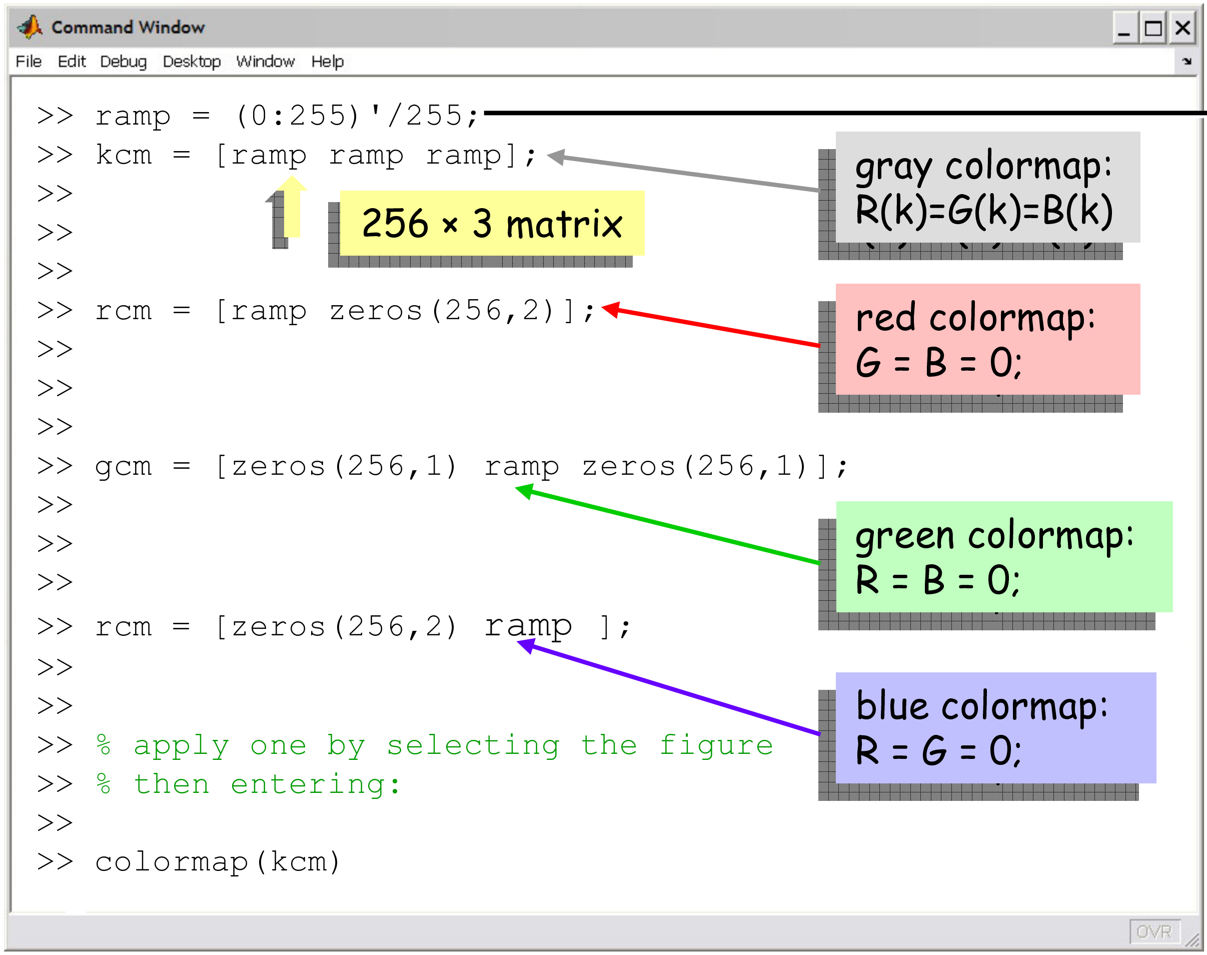
Number at pixel location is an index into a colormap.

Intensity values are integers between 0 and 1.

```
>> title('Pl
>> xlabel('Ji
>> truesize
>> T = imread('Jim Woodring - PlusMinus.jpg','jpg');
>> figure
>> image(T)
>> truesize
>> T(231,326,:)
ans(:,:,1) =
  227
ans(:,:,2) =
  222
ans(:,:,3) =
  96
>> I(231,326,:)
ans =
  214
>> cmap(214,:)
ans =
    0.8863    0.9059    0.2549
>> round(255*cmap(214,:))
ans =
   226   231    65
>>
```

image class: uint8
image type: colormapped

How to Make Colormaps



```
>> ramp = (0:255)'/255;
>> kcm = [ramp ramp ramp];
>>
>>
>>
>> rcm = [ramp zeros(256,2)];
>>
>>
>>
>> gcm = [zeros(256,1) ramp zeros(256,1)];
>>
>>
>>
>> rcm = [zeros(256,2) ramp ];
>>
>>
>> % apply one by selecting the figure
>> % then entering:
>>
>> colormap(kcm)
```

256 × 3 matrix

gray colormap:
R(k)=G(k)=B(k)

red colormap:
G = B = 0;

green colormap:
R = B = 0;

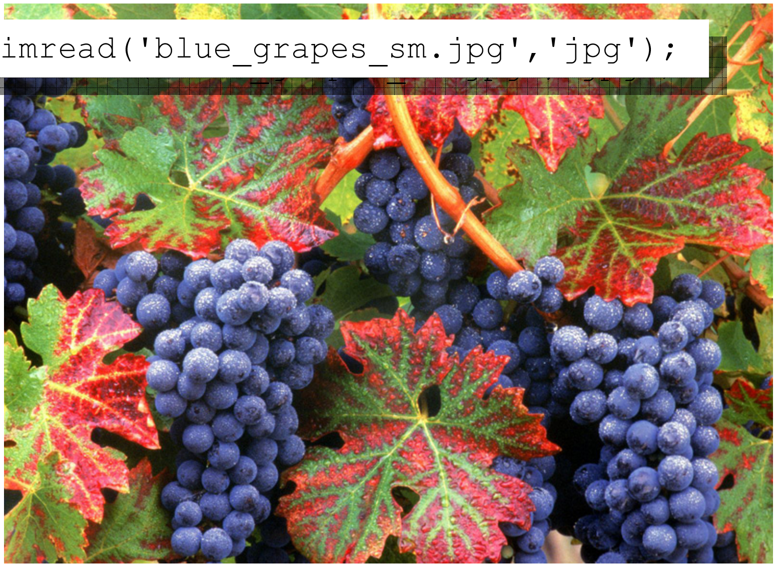
blue colormap:
R = G = 0;

```
0
0.0039
0.0078
0.0118
0.0157
  ⋮

0.9843
0.9882
0.9922
0.9961
1.0000
```

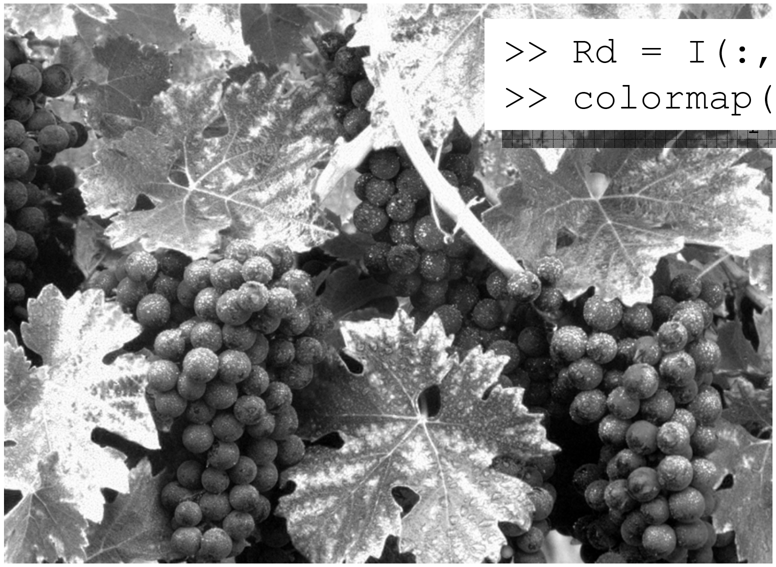
This code, 0:255 , generates a 1 row by 256 element vector of class double that contains numbers 0 through 255 inclusive.

This, (0:255)' , has the same contents and class but is a 256 row by 1 column vector. The apostrophe (') is the matrix transpose operator.

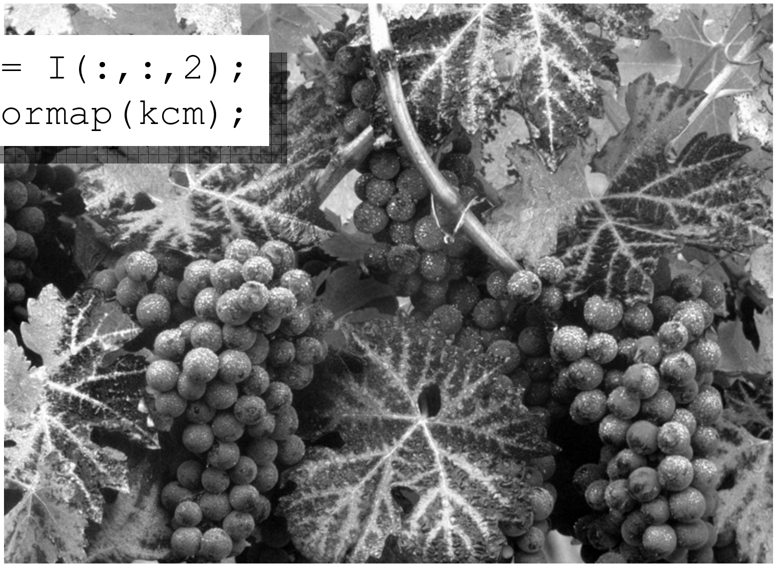
R, G, & B bands of a truecolor image displayed with grayscale colormaps



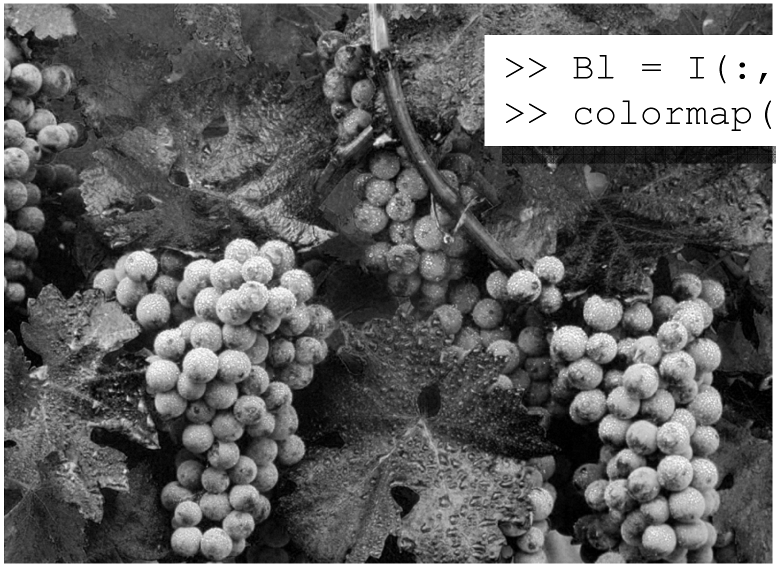
```
>> I = imread('blue_grapes_sm.jpg','jpg');
```



```
>> Rd = I(:,:,1);
>> colormap(kcm);
```



```
>> Gn = I(:,:,2);
>> colormap(kcm);
```

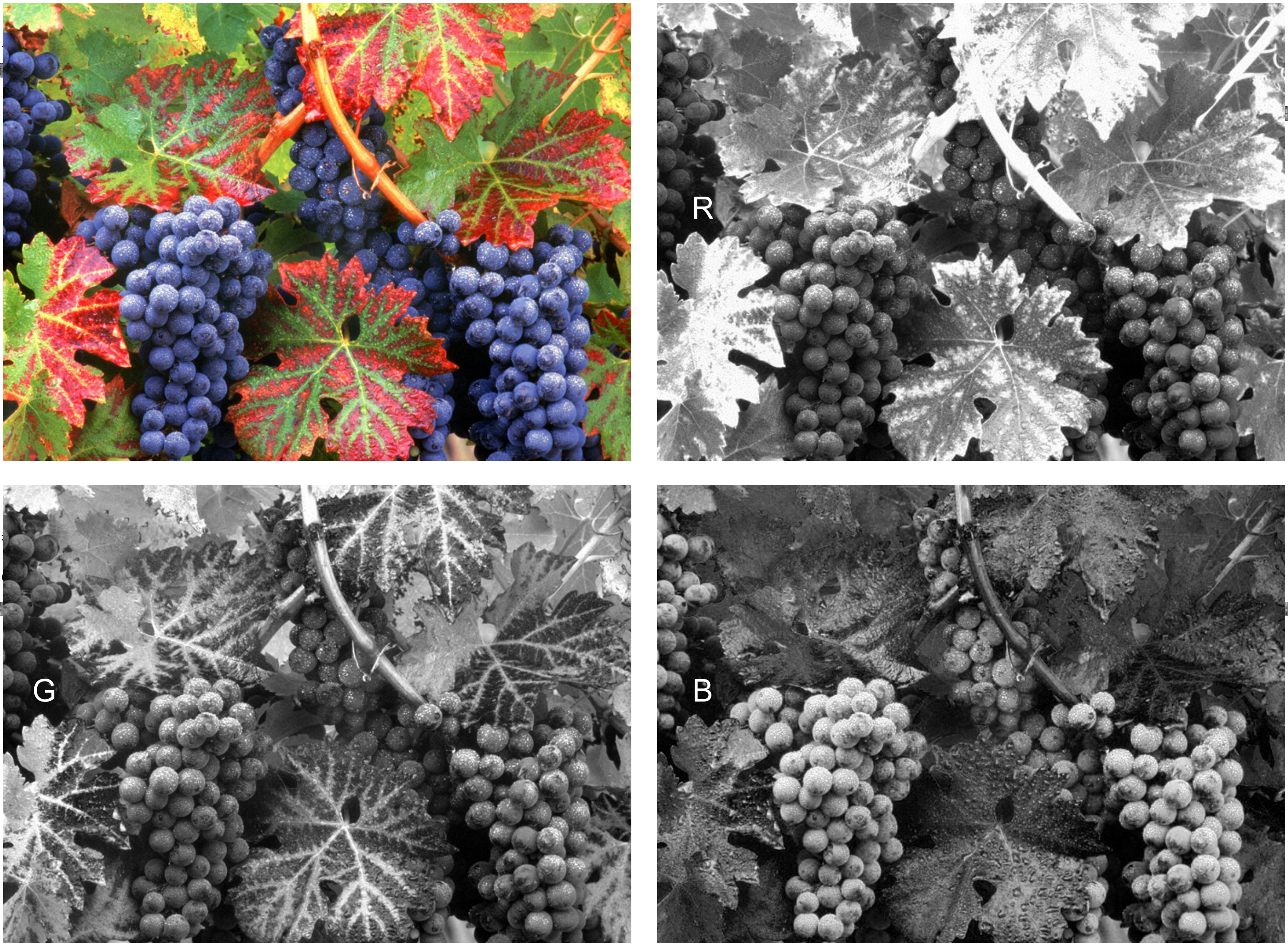


```
>> Bl = I(:,:,3);
>> colormap(kcm);
```

R, G, & B bands of a truecolor image displayed with grayscale colormaps

```
>> I = 
```

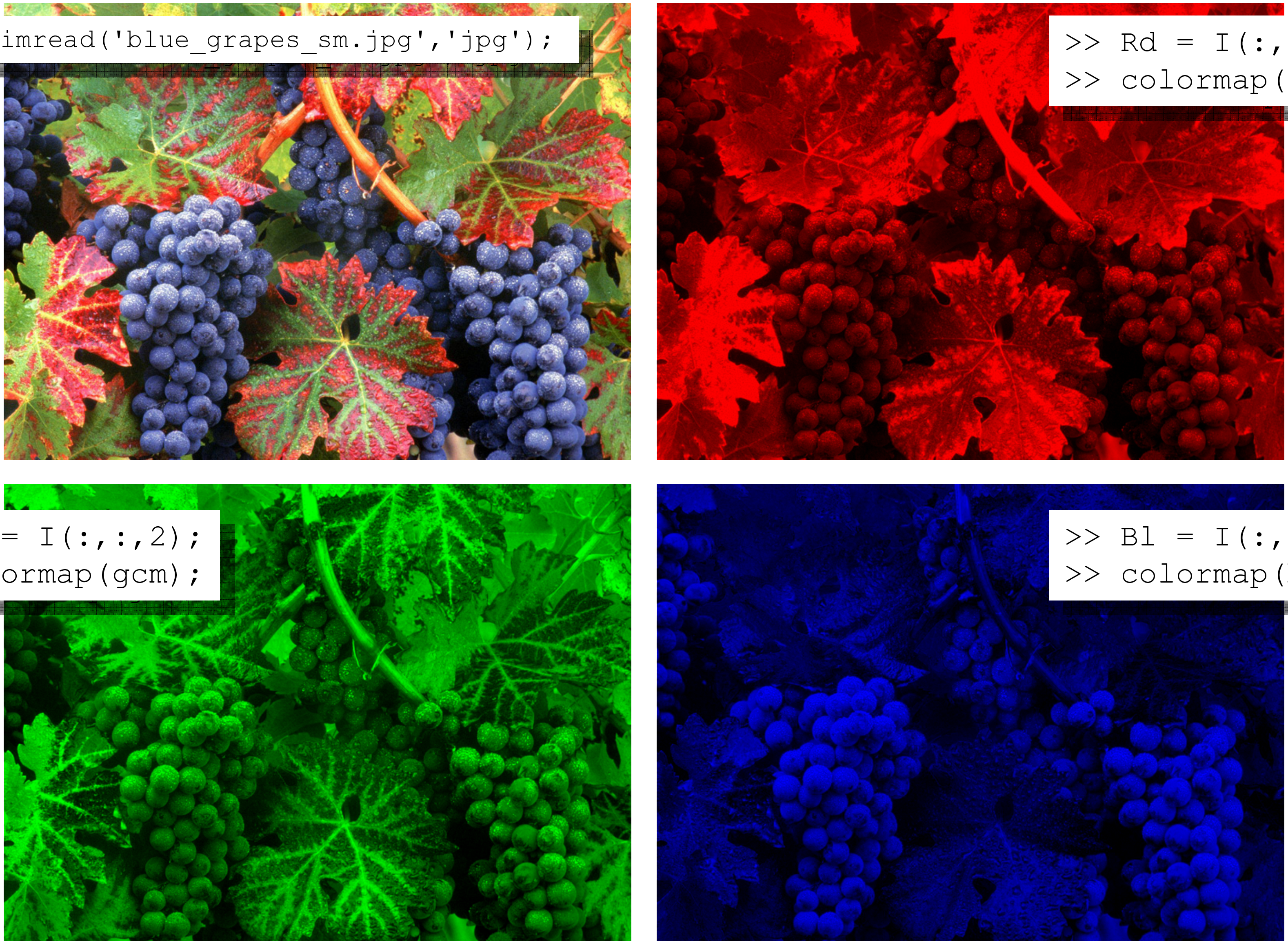
```
:,1);
kcm);
```



```
>> Gn =
>> col
```

```
:,3);
kcm);
```

R, G, & B bands of a truecolor image displayed with tinted colormaps



```
>> I = imread('blue_grapes_sm.jpg','jpg');
```

```
>> Rd = I(:,:,1);
>> colormap(rcm);
```

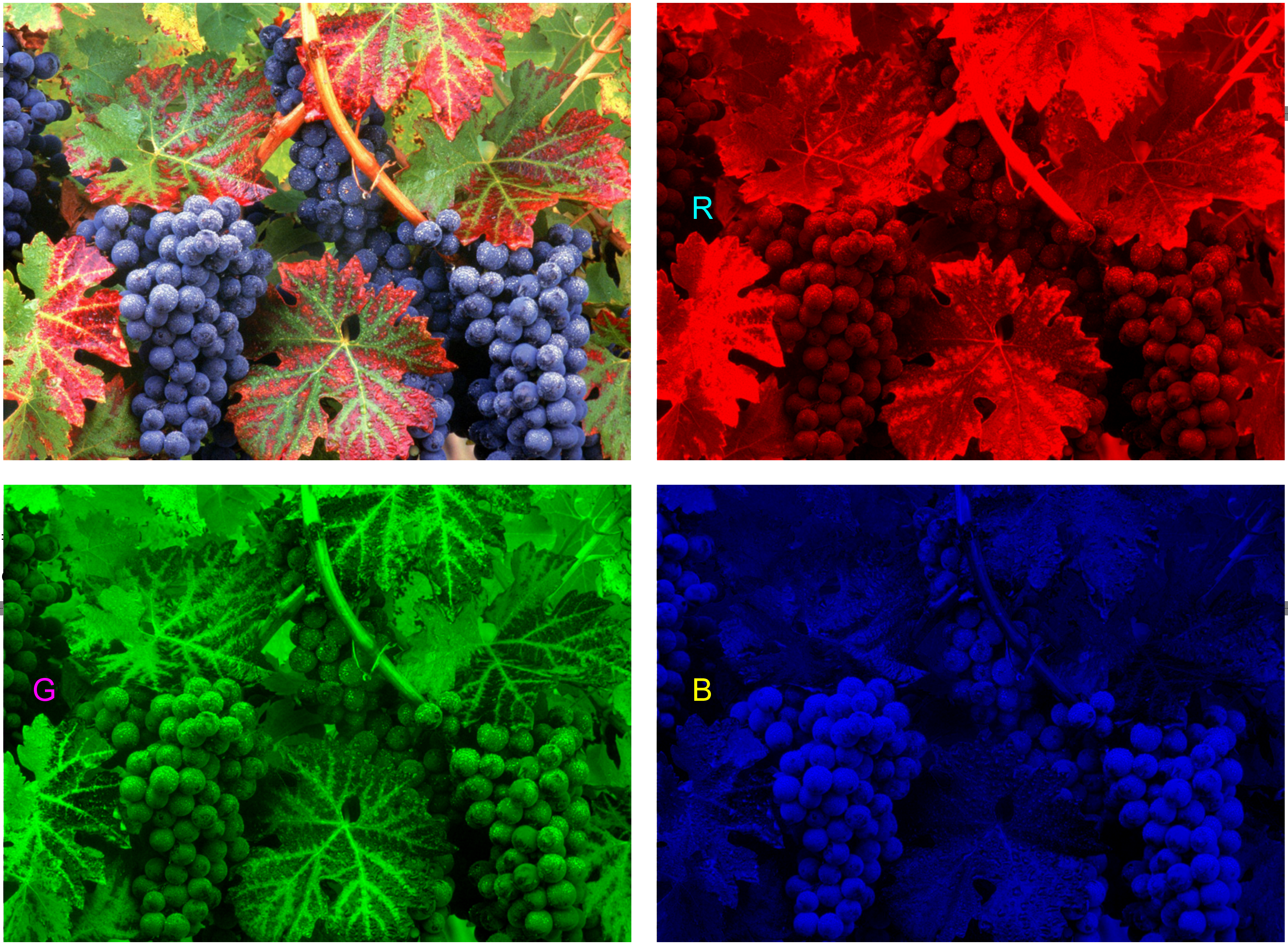
```
>> Gn = I(:,:,2);
>> colormap(gcm);
```

```
>> Bl = I(:,:,3);
>> colormap(bcm);
```

R, G, & B bands of a truecolor image displayed with tinted colormaps

```
>> I = 
```

```
:,1);
rcm);
```



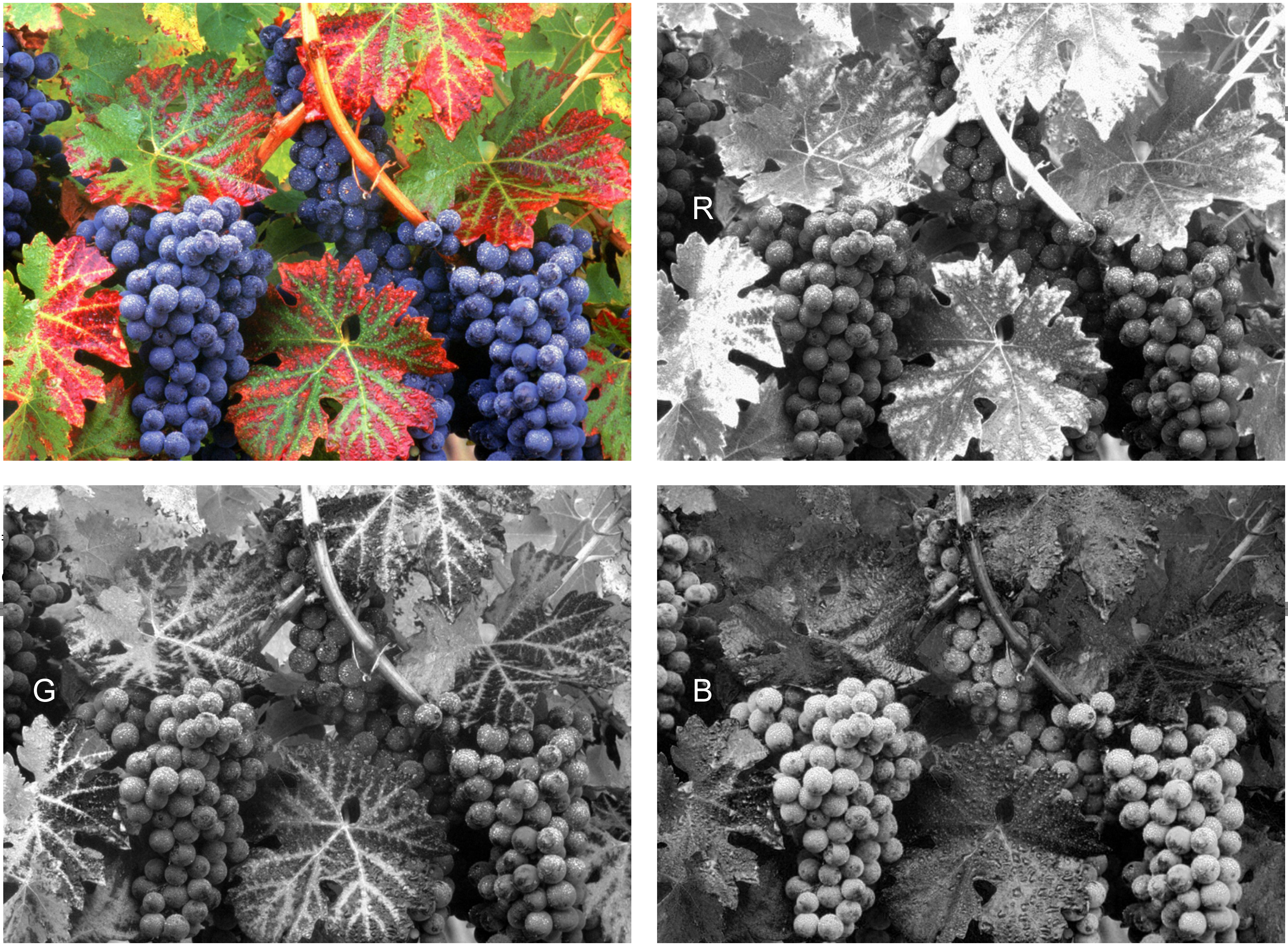
```
>> Gn 
>> col
```

```
:,3);
bcm);
```

R, G, & B bands of a truecolor image displayed with grayscale colormaps

```
>> I = 
```

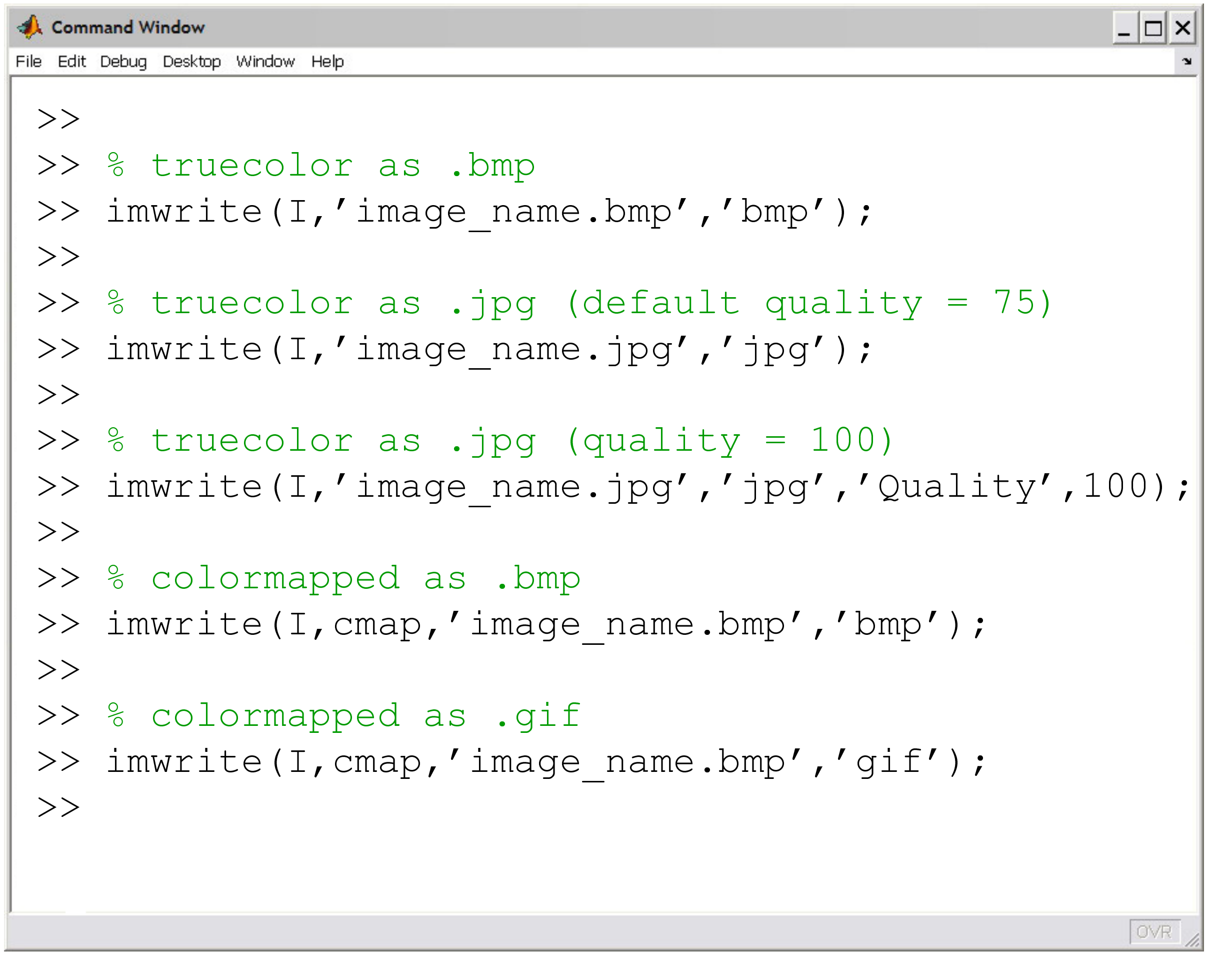
```
:,1);
kcm);
```



```
>> Gn 
>> col
```

```
:,3);
kcm);
```

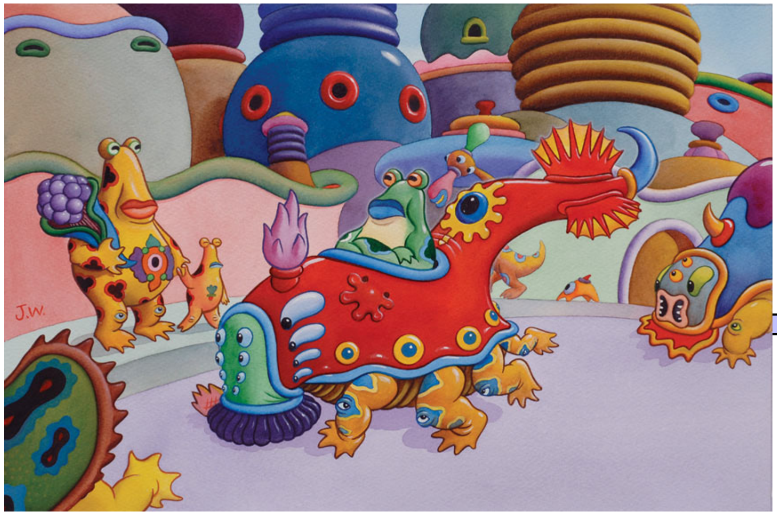
Saving Images as Files



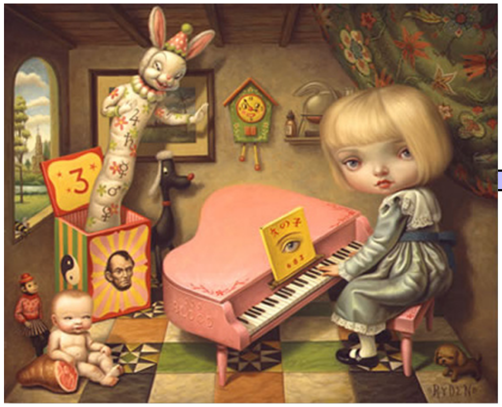
```
>>
>> % truecolor as .bmp
>> imwrite(I,’image_name.bmp’,’bmp’);
>>
>> % truecolor as .jpg (default quality = 75)
>> imwrite(I,’image_name.jpg’,’jpg’);
>>
>> % truecolor as .jpg (quality = 100)
>> imwrite(I,’image_name.jpg’,’jpg’,’Quality’,100);
>>
>> % colormapped as .bmp
>> imwrite(I,cmap,’image_name.bmp’,’bmp’);
>>
>> % colormapped as .gif
>> imwrite(I,cmap,’image_name.bmp’,’gif’);
>>
```

Assuming that
  ‘I’ contains the image of the correct class,
that
  ‘cmap’ is a colormap,
and that
  ‘image_name’ is the file-name that you want.

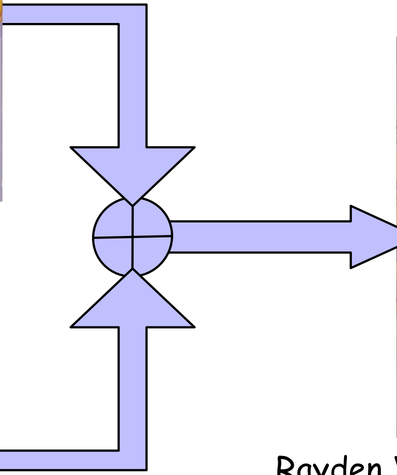
Double Exposure: Adding Two Images

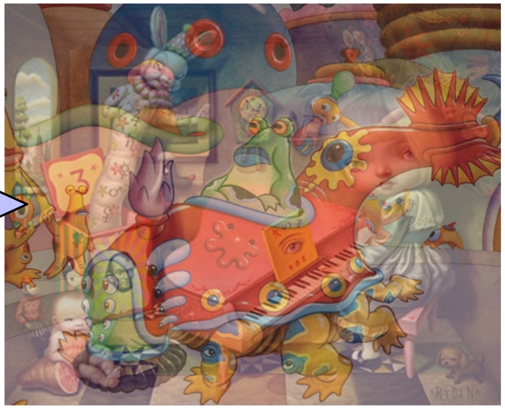


Jim Woodring - Bumperillo



Mark Rayden – The Ecstasy of Cecelia





Rayden Woodring – The Ecstasy of Bumperillo (?)

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, pixelwise.

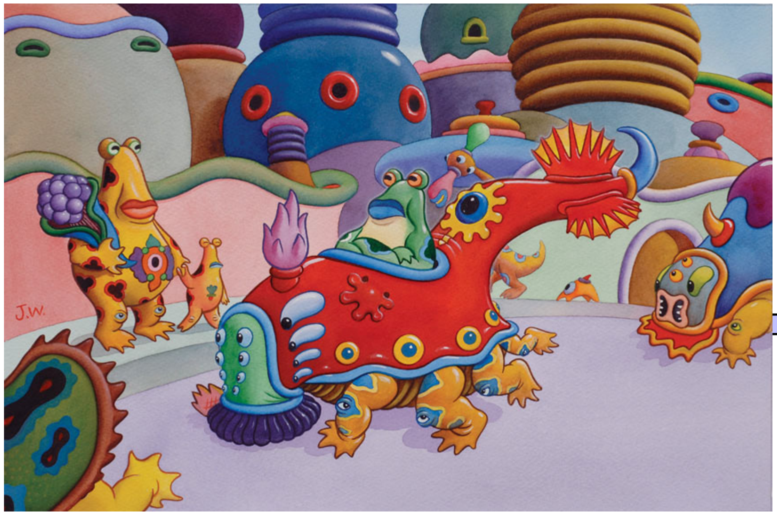
Double Exposure: Adding Two Images

Example Matlab Code

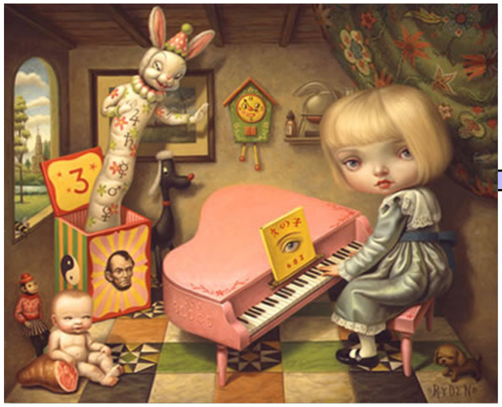
```
>> cd 'D:\Classes\EECE253\Fall 2006\Graphics\matlab intro'
>> JW = imread('Jim Woodring - Bumperillo.jpg','jpg');
>> figure
>> image(JW)
>> truesize
>> title('Bumperillo')
>> xlabel('Jim Woodring')
>> MR = imread('Mark Ryden - The Ecstasy of Cecelia.jpg','jpg');
>> figure
>> image(MR)
>> truesize
>> 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.

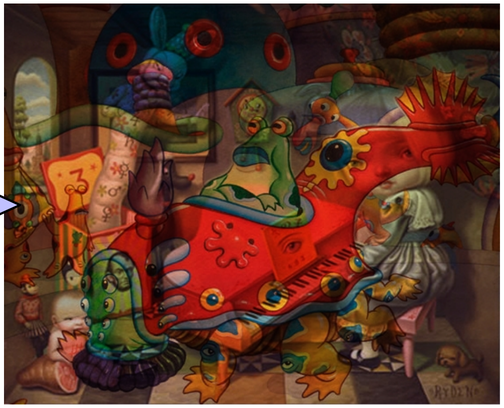
Intensity Masking: Multiplying Two Images



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')
```

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.

Pixel Indexing in Matlab

"IP_Function" is some arbitrary image processing function that you or someone else has written.

"For" loops in Matlab are inefficient, whereas Matlab's native indexing procedures are very fast.

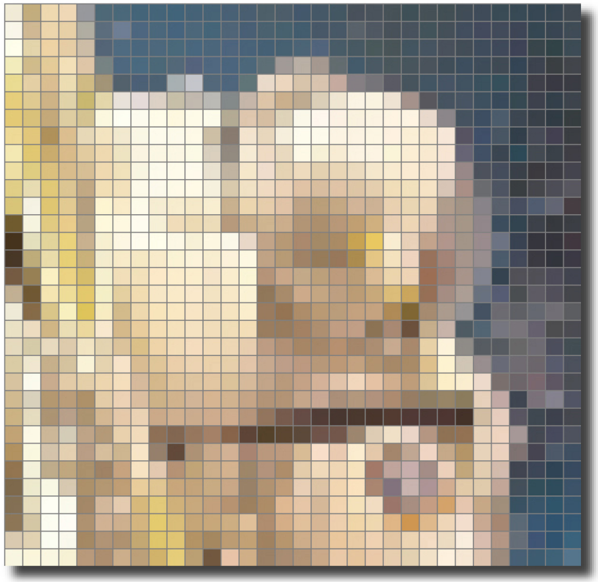
Rather than

```
for r = 1:R
   for c = 1:C
      J(r,c,:) = IP_Function(I(r,c,:));
   end
end
```

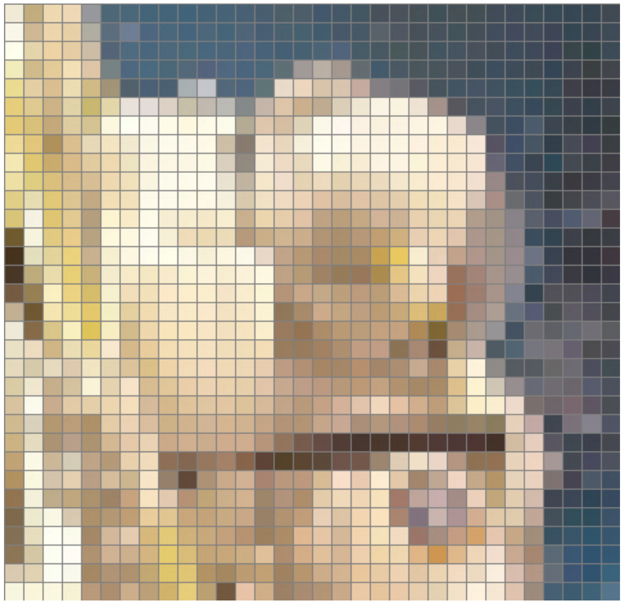
use, if possible

```
J = IP_Function(I);
```

But, sometimes that is not possible.
For example, if the output, *J*, is decimated with respect to the input, *I*, the above will not work (unless, of course, it is done within IP_function).



Pixel Indexing in Matlab

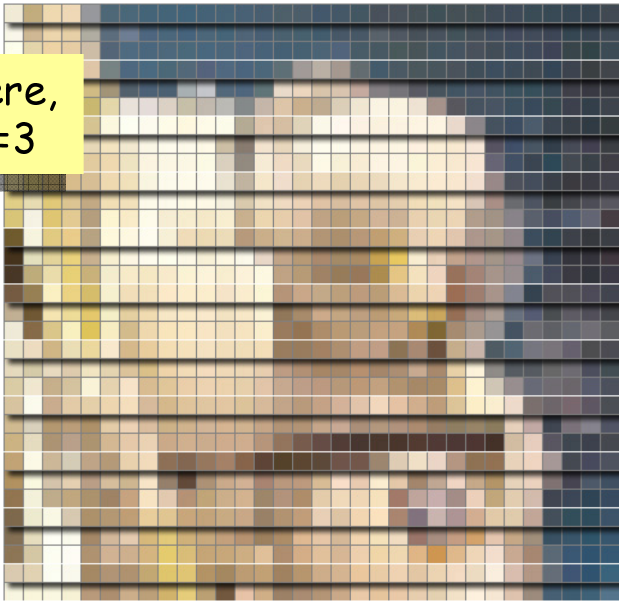


```
r = 1:n:R;
```

```
I(r,:,:)
```

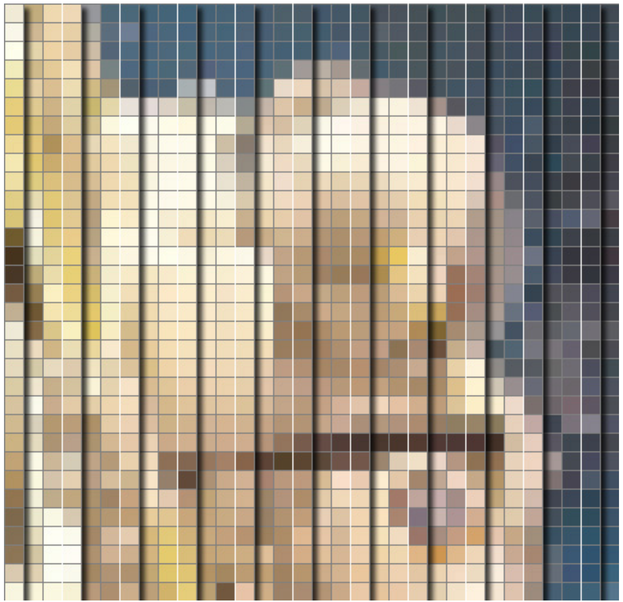
```
c = 1:n:C;
```

```
I(:,c,:)
```



Here, n=3

$r = [1\ 4\ 7\ 10\ 13\ 16\ 19\ 22\ 25\ 28\ 31]$

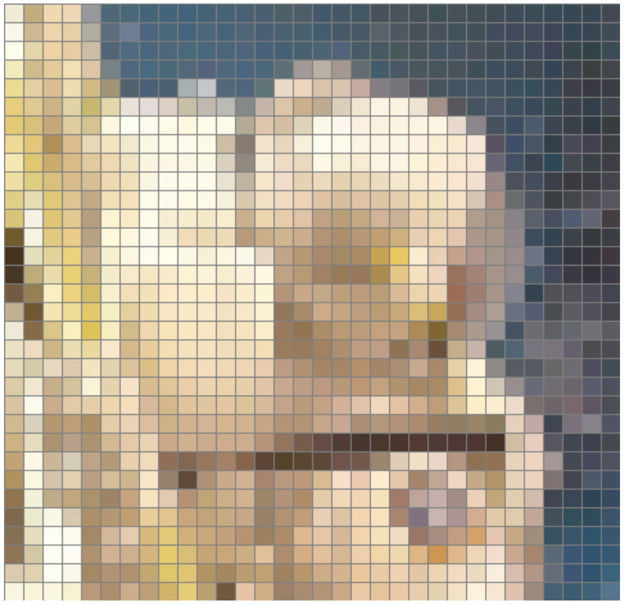


$c = [1\ 4\ 7\ 10\ 13\ 16\ 19\ 22\ 25\ 28\ 31]$

To decimate the above image by a factor of *n*, create a vector, **r**, that contains the index of every *n*th row, and a similar vector, **c**.

Pixel Indexing in Matlab

This is called, 'vectorizing'.

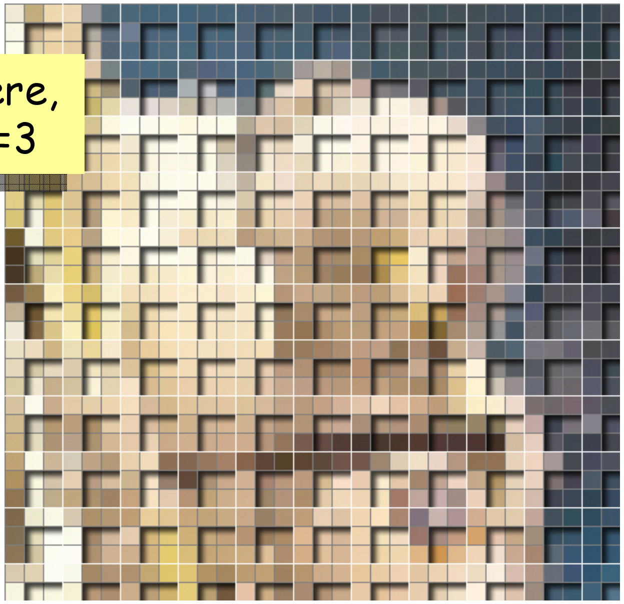


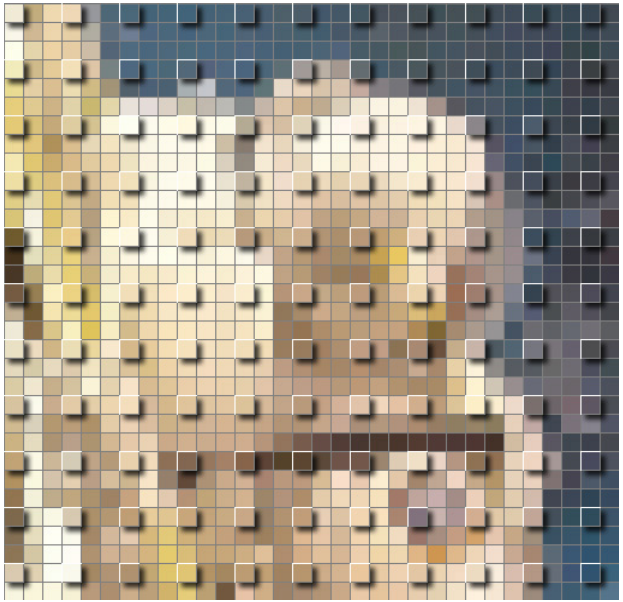
Take the pixels indexed by both *r* and *c*.

↓

```
I(r,c)
```

Here, n=3

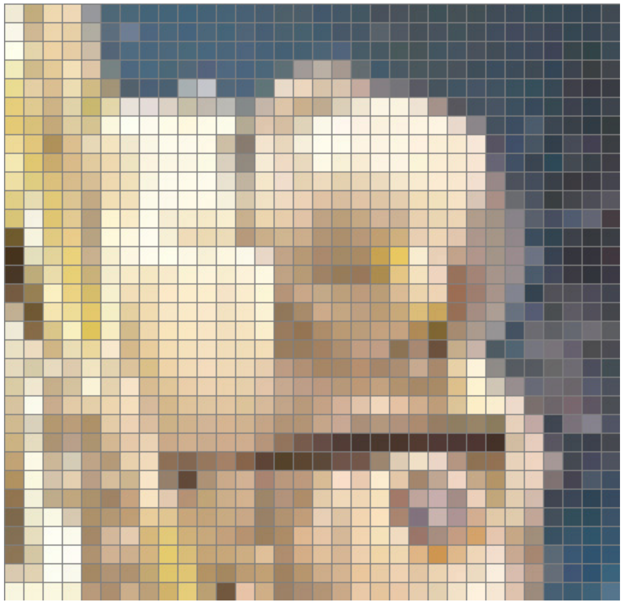




Then, vectors **r** and **c** used as index arguments for image *I* select every *n*th column in every *n*th row.

Pixel Indexing in Matlab

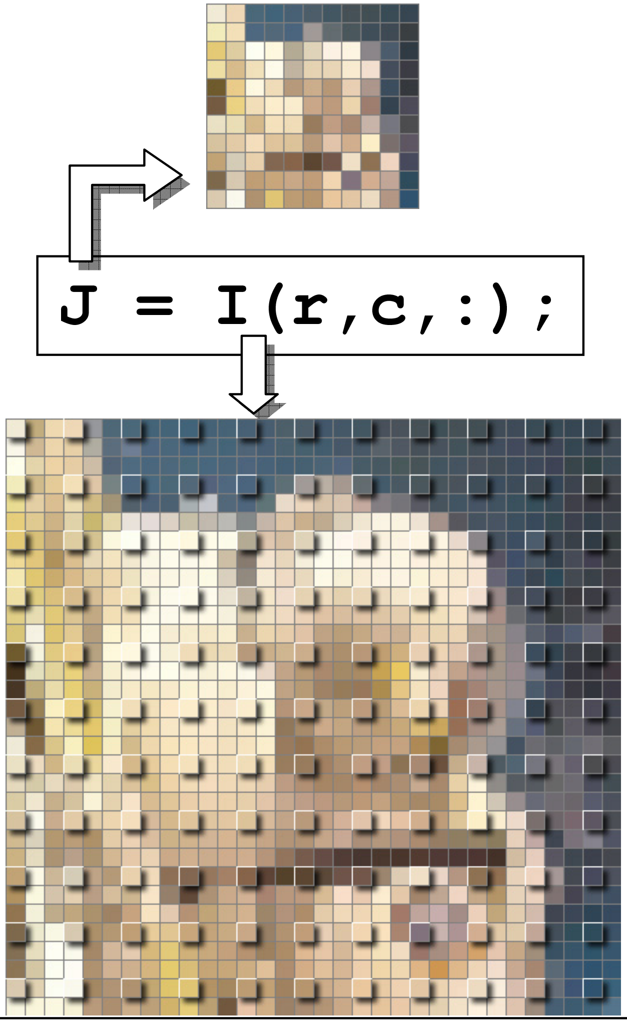
Here, n=3



image, `I`

```
r = 1:n:R;
```

```
c = 1:n:C;
```



```
J = I(r,c,:);
```

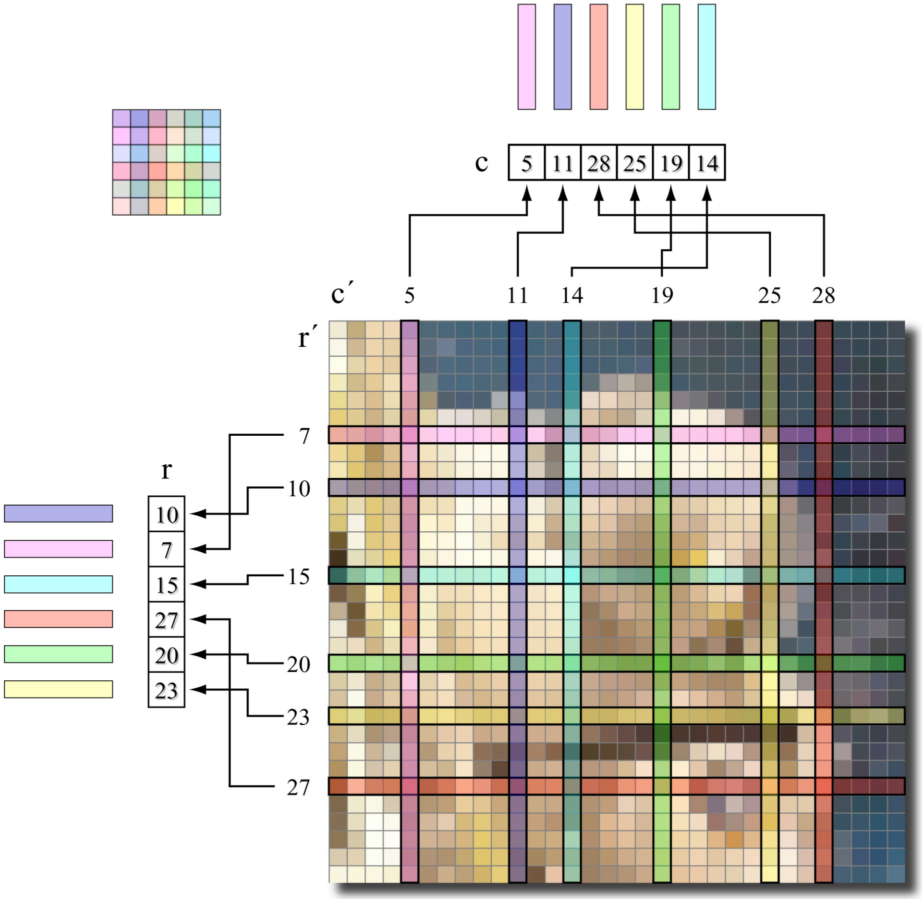
Pixel Indexing in Matlab

Indexing in Matlab is fully general.

If `I` is R x C x B, vectors **r** and **c** can contain any numbers $1 \le r_k \le R$ and $1 \le c_k \le C$.

The numbers can be in any order and can be repeated within **r** and **c.**

The result of `I(r,c)` is an ordinal shuffling of the pixels from I as indexed by **r** and **c**.



Whenever possible, avoid using 'for' loops; vectorize instead.

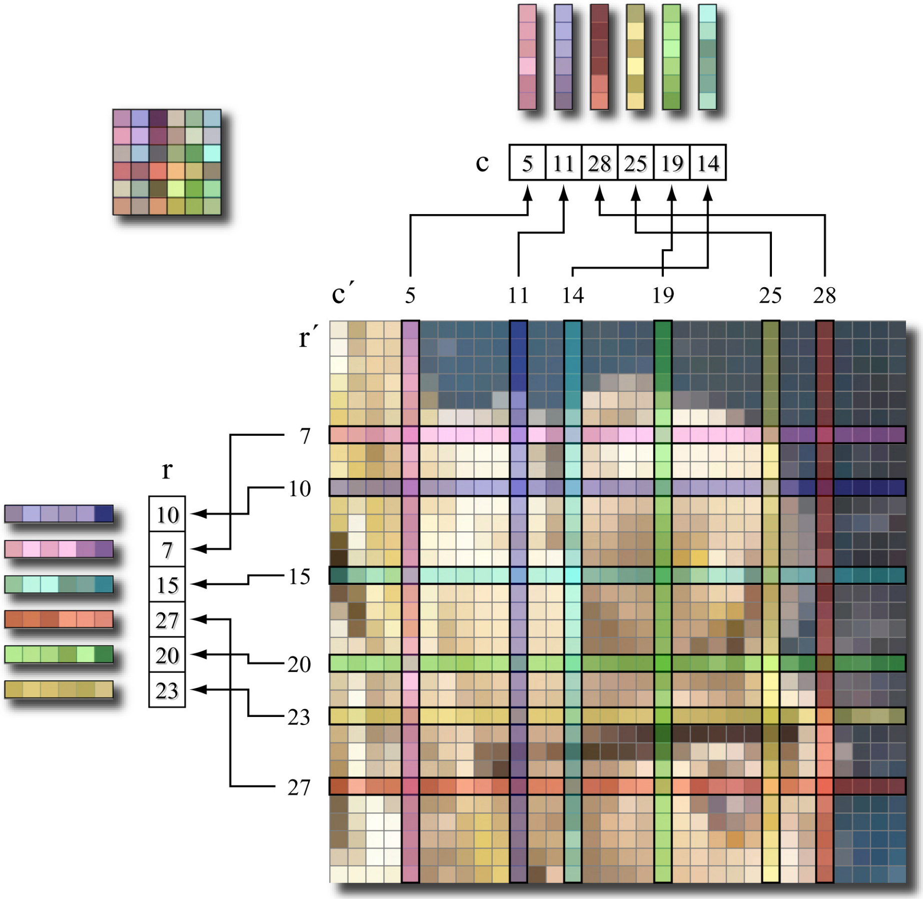
Pixel Indexing in Matlab

Indexing in Matlab is fully general.

If `I` is R x C x B, vectors **r** and **c** can contain any numbers $1 \leq r_k \leq R$ and $1 \leq c_k \leq C$.

The numbers can be in any order and can be repeated within **r** and **c.**

The result of `I(r,c)` is an ordinal shuffling of the pixels from I as indexed by **r** and **c**.



Whenever possible, avoid using 'for' loops; vectorize instead.

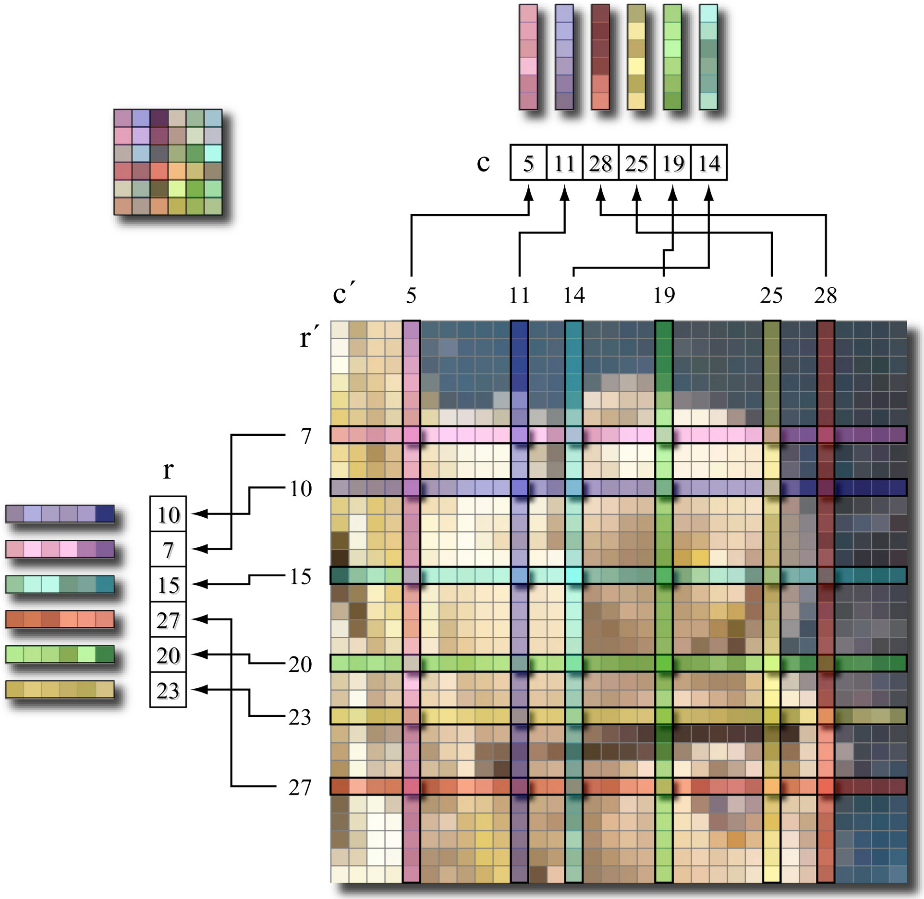
Pixel Indexing in Matlab

Indexing in Matlab is fully general.

If `I` is R x C x B, vectors **r** and **c** can contain any numbers $1 \le r_k \le R$ and $1 \le c_k \le C$.

The numbers can be in any order and can be repeated within **r** and **c.**

The result of `I(r,c)` is an ordinal shuffling of the pixels from I as indexed by **r** and **c**.



Whenever possible, avoid using 'for' loops; vectorize instead.

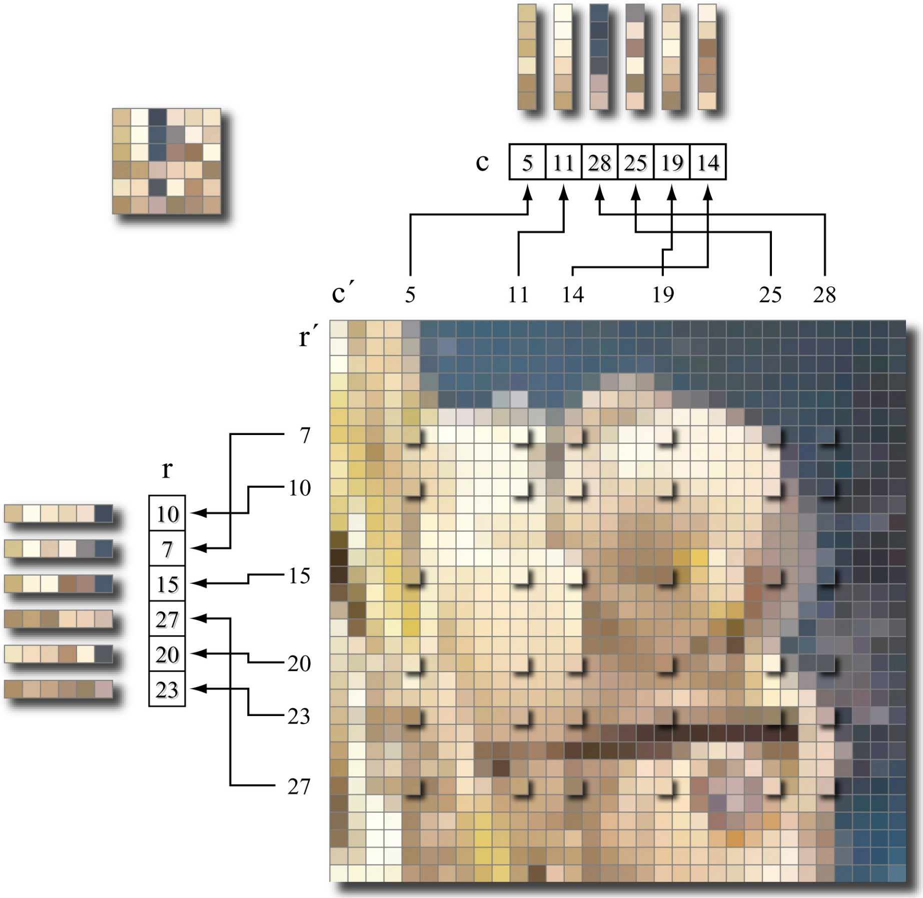
Pixel Indexing in Matlab

Indexing in Matlab is fully general.

If `I` is R x C x B, vectors **r** and **c** can contain any numbers $1 \leq r_k \leq R$ and $1 \leq c_k \leq C$.

The numbers can be in any order and can be repeated within **r** and **c.**

The result of `I(r,c)` is an ordinal shuffling of the pixels from I as indexed by **r** and **c**.



Whenever possible, avoid using 'for' loops; vectorize instead.

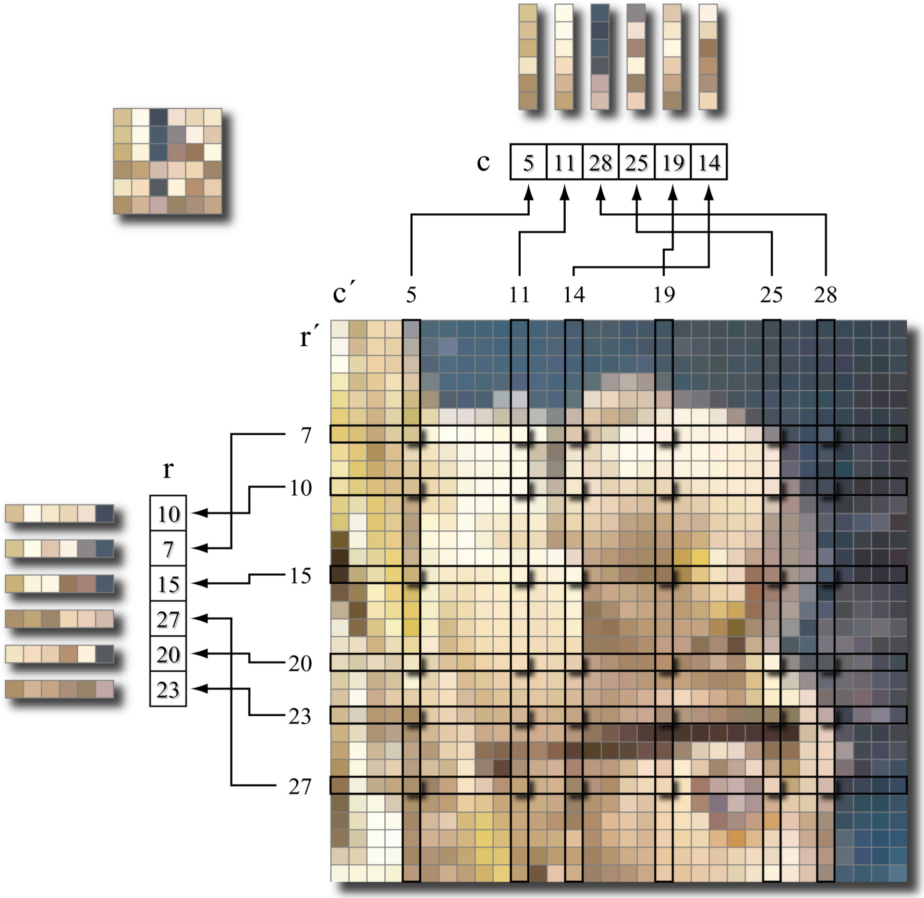
Pixel Indexing in Matlab

Indexing in Matlab is fully general.

If `I` is R x C x B, vectors **r** and **c** can contain any numbers $1 \le r_k \le R$ and $1 \le c_k \le C$.

The numbers can be in any order and can be repeated within **r** and **c.**

The result of `I(r,c)` is an ordinal shuffling of the pixels from I as indexed by **r** and **c**.



Whenever possible,
avoid using 'for' loops;
vectorize instead.

Pixel Indexing in Matlab

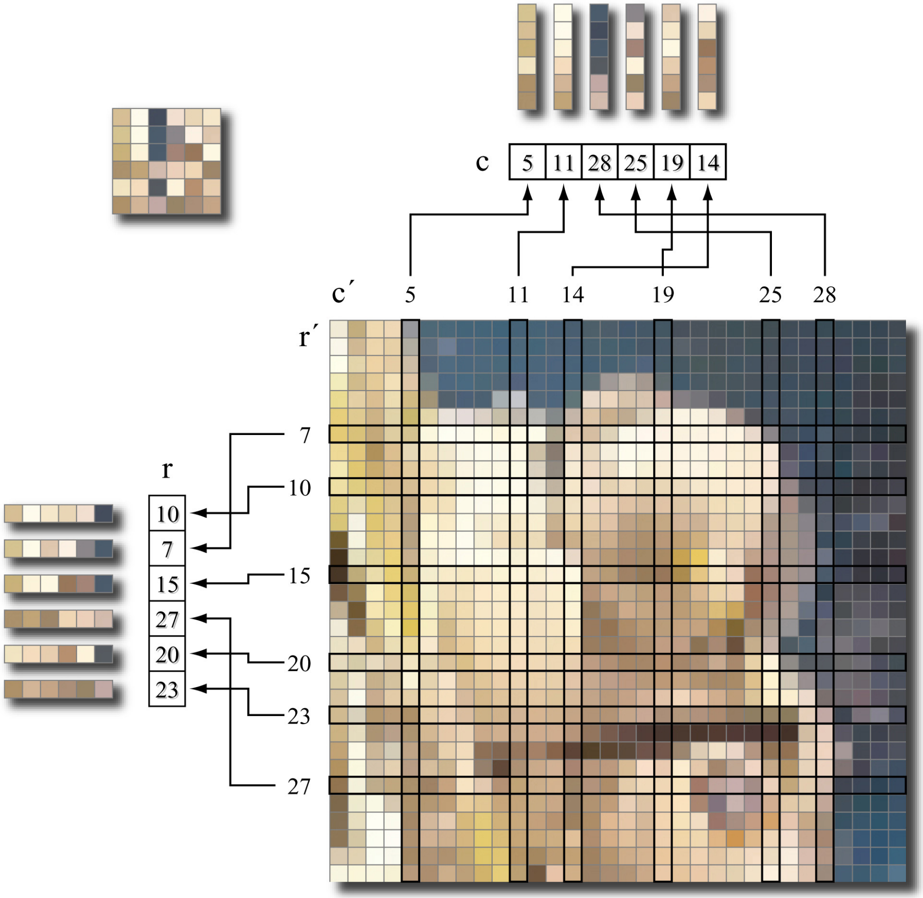
Indexing in Matlab is fully general.

If `I` is R x C x B, vectors **r** and **c** can contain any numbers $1 \leq r_k \leq R$ and $1 \leq c_k \leq C$.

The numbers can be in any order and can be repeated within **r** and **c.**

The result of `I(r,c)` is an ordinal shuffling of the pixels from I as indexed by **r** and **c**.

Whenever possible, avoid using 'for' loops; vectorize instead.


c
5
11
28
25
19
14
c′
5
11
14
19
25
28
r′
r
10
7
15
27
20
23
7
10
15
20
23
27