Numb3rs

Number and image types

CS112 Scientific Computation
Department of Computer Science
Wellesley College

Numb3rs

Most of our numbers so far have been of type double
for *double-precision* floating point

mantissa exponent

6.6260755×10^{34}

base

double numbers use 8 bytes (64 bits) to store each number – 52 for the mantissa (~ 16 significant digits), 11 bits for exponent, and one sign bit

High precision requires more memory space and processing time! And may not be needed....

Other number types

Name	Size(bytes)	Description
uint8	1	integers 0 to 255
uint16	2	integers 0 to 65,535
uint32	4	integers 0 to 4,294,967,295
int8	1	integers -128 to 127
int16	2	integers -32,768 to 32,767
int32	4	integers -2,147,483,648 to 2,147,483,647
single	4	single-precision floating point (fewer bits for mantissa & exponent)
double	8	double-precision floating point

There is also a logical type to represent binary values
0 (false) and 1 (true), stored in 1 byte

3

For each number type ...

... there is a built-in function of the same name that converts its input to the desired type:

```
>> bigNums = zeros(1, 100);
>> smallNums = uint8(zeros(1, 100));
>> whos
```

Name	Size	Bytes	Class
bigNums	1x100	800	double array
smallNums	1x100	100	uint8 array

zeros and ones can also be called with a third input:

```
>> smallNums = zeros(1, 100, 'uint8');
```

* What happens if we try to store a number in smallNums that is less than 0 or greater than 255? How are fractional numbers handled?

4

The silver screen

Some of our images have been black and white images stored in matrices of double type values spanning the range from 0.0 to 1.0

To conserve memory space, images are typically stored in files using formats that represent each image intensity using a small number of bits*

(e.g., 1, 8, 16 bits)

*note that the human eye can only distinguish about 100 shades of gray at one time

5

MATLAB's image processing toolbox

MATLAB's imread and imwrite read and write images in many possible formats, such as JPEG, GIF, TIFF, BMP, PNG

Any of these image types can then be displayed using imshow

```
>> mona = imread('monaLisa.jpg');
>> imshow(mona);
>> imwrite(mona, 'monaLisa.png');
```

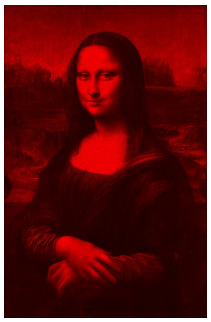
MATLAB uses the specified second filename to determine the format for the stored image

6

Storing a masterpiece

Recall that in an RGB color image, each picture element (pixel) consists of three values:

red



green

blue

7

Mona is three-dimensional

```
>> whos
  Name        Size              Bytes  Class
  mona        864x560x3       1451520  uint8 array
```

mona's third dimension

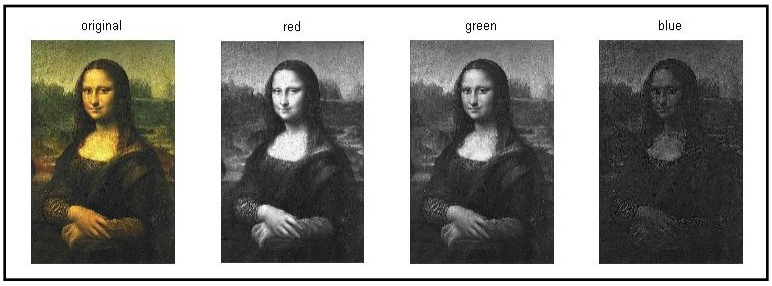
Third dimension has three indices, corresponding to the amount of red, green, and blue at each image location, specified as an integer between 0 & 255

RGB values at each location can be viewed with imtool(mona)

8

Exercise

How did I create this figure, where the gray-level images show red, green and blue components?



Suppose you want to show the components in shades of red/green/blue?

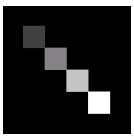
9

imshow with a colormap

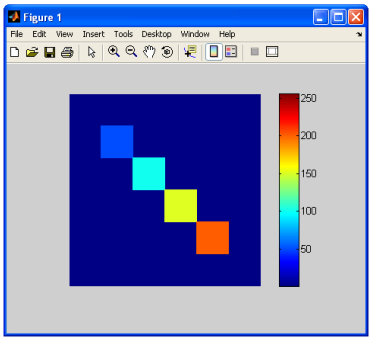
imshow can display an *indexed image* with a colormap:

```
image = zeros(300, 300, 'uint8');
for pos = 50:50:200
    image(pos:pos+50,  pos:pos+50) = pos;
end
```

```
>> imshow(image, [0 200])
```



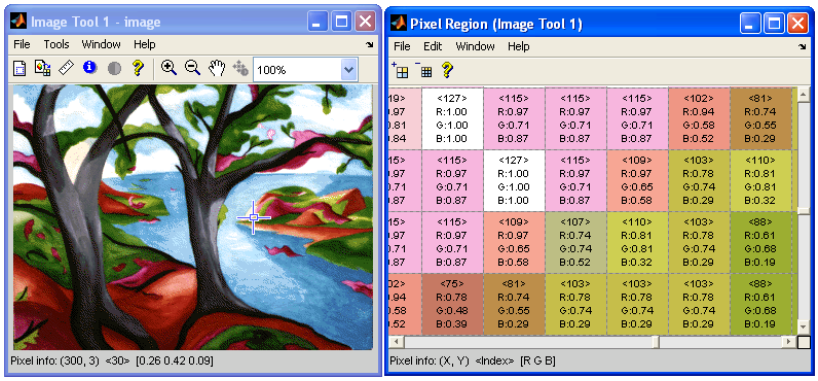
```
>> imshow(image, jet)
>> colorbar
```



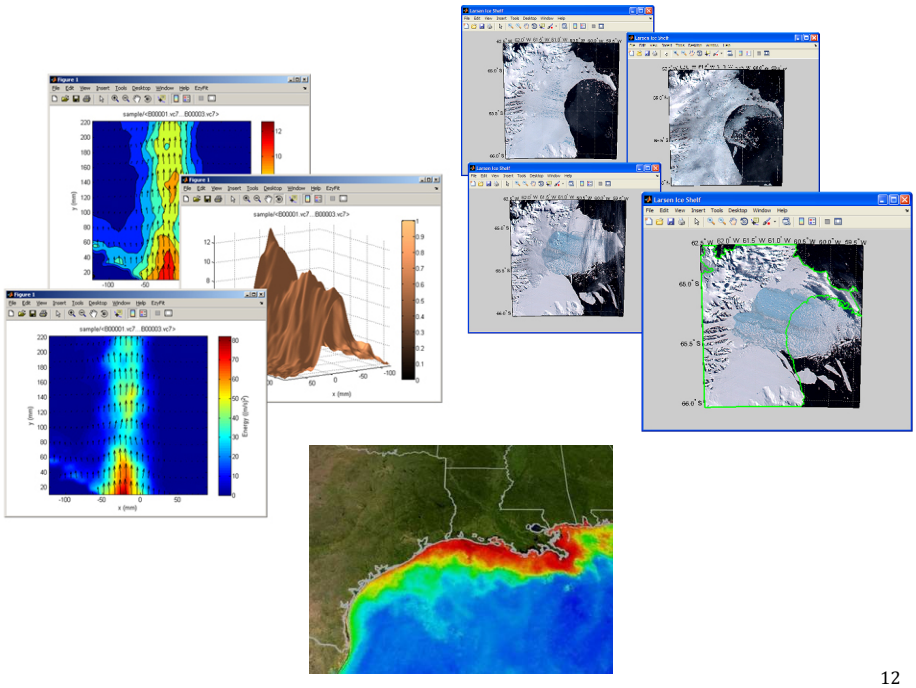
10

A “real” indexed image

```
>> [image cmap] = imread('trees.tif');
>> imtool(image, cmap);
```



11



12