CSE468: Digital Image Processing

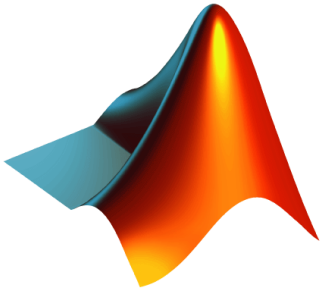
Introduction to Matlab & The Image Processing Toolbox

Disclaimer

- This Slides are based on:
  - The slides of Gowtham Bellala, University of Michigan.
  - Tuo Wang, University of Wisconsin-Madisen.

Introduction

- What is Matlab ?
  - MATLAB is a software package for doing numerical computation. It was originally designed for solving linear algebra type problems using matrices. It's name is derived from MATrix LABoratory.
- Why Matlab and not ***** ?
  1. Images are stored as matrices.
  2. Flexibility.
  3. Documentation.



Matlab Environment

- MATLAB window components:
  - Workspace
    > Displays all the defined variables
  - Command Window
    > To execute commands in the MATLAB environment
  - Command History
    > Displays record of the commands used
  - File Editor Window
    > Define your functions
  - Current Folder Window
    > Indicates the current working directory

Matlab Variable Names

- Variable names are case sensitive
- Variable names must start with a letter and can be followed by letters, digits and underscores.
  Examples :

```
>> x = 2;
>> abc_123 = 0.005;
>> 1ab = 2;
Error: Unexpected MATLAB expression
```

Matlab Special Variables

- pi Value of π
- eps Smallest incremental number
- Inf Infinity
- NaN Not a number e.g. 0/0
- i and j i = j = square root of -1

Matlab Relational Operators

- Matlab supports six relational operators

Less Than	<
Less Than or Equal	<=
Greater Than	>
Greater Than or Equal	>=
Equal To	==
Not Equal To	~= (NOT != like in C)

Matlab Matrices

- MATLAB treats all variables as matrices. For our purposes a matrix can be thought of as an array, in fact, that is how it is stored.

- Vectors are special forms of matrices and contain only one row OR one column.

- Scalars are matrices with only one row AND one column

Generating Matrices

- A scalar can be created in MATLAB as follows:

```
>> x = 23;
```

- A matrix with only one row is called a row vector. A row vector can be created in MATLAB as follows (note the commas):

```
>> y = [12, 10, -3]
y =
    12 10 -3
```

- A matrix with only one column is called a column vector. A column vector can be created in MATLAB as follows:

```
>> z = [12; 10; -3]
z =
    12
    10
    -3
```

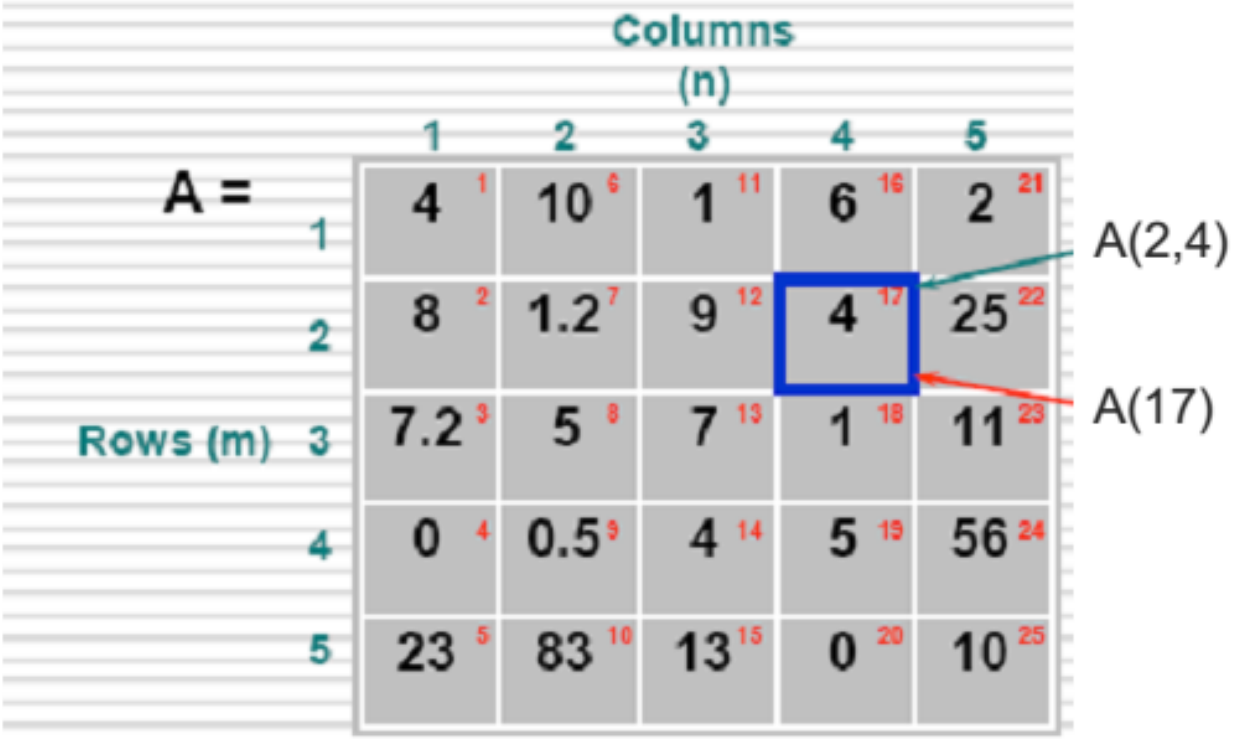
Generating Matrices

- A matrix can be created in MATLAB as follows (note the commas and semicolons)

```
>> X = [1, 2, 3; 4, 5, 6; 7, 8, 9]
X =
     1  2  3
     4  5  6
     7  8  9
```

Matrices must be rectangular!

The Matrix in Matlab



- Note: Unlike C, Matlab's indices start from 1

Extracting a Sub-matrix

- A portion of a matrix can be extracted and stored in a smaller matrix by specifying the names of both matrices and the rows and columns to extract. The syntax is:

  sub_matrix = matrix (r1 : r2 , c1 : c2) ;

- Where r1 and r2 specify the beginning and ending rows and c1 and c2 specify the beginning and ending columns to be extracted to make the new matrix.

Extracting a Sub-matrix

- Example :

```
>> X = [1, 2, 3; 4, 5, 6; 7, 8, 9]
X =
     1   2   3
     4   5   6
     7   8   9

>> X22 = X(1:2 , 2:3)
X22 =
     2   3
     5   6

>> X13 = X(3,1:3)
X13 =
     7   8   9

>> X21 = X(1:2,1)
X21 =
     14
```

Matrix Addition

- Increment all the elements of a matrix by a single value

```
>> x = [1,2;3,4]
x =
     1  2
     3  4
>> y = x + 5
y =
     6  7
     8  9
```

- Adding two matrices

```
>> xsy = x + y
xsy =
      7   9
     11  13
>> z = [1,0.3]
z =
      1   0.3
>> xsz = x + z
```

??? Error using => plus
Matrix dimensions must agree

Matrix Multiplication

- Matrix multiplication

```
>> a = [1,2;3,4]; (2x2)
>> b = [1,1]; (1x2)
>> c = b*a
c =
     4   6

>> c = a*b
??? Error using ==>
mtimes Inner matrix
dimensions
must agree.
```

- Element wise multiplication

```
>> a = [1,2;3,4];
>> b = [1,½;1/3,¼];
>> c = a.*b
c =
     1   1
     1   1
```

Matrix Element wise operations

- ```
  >> a = [1,2;1,3];
  >> b = [2,2;2,1];
  ```

- Element wise division
  ```
  >> c = a./b
  c =
      0.5   1
      0.5   3
  ```

- Element wise multiplication
  ```
  >> c = a.*b
  c =
      2     4
      2     3
  ```

- Element wise power operation
  ```
  >> c = a.^2
  c =
      1     4
      1     9
  >> c = a.^b
  c =
      1     4
      1     3
  ```
  ```

Matrix Manipulation Functions

- zeros : creates an array of all zeros, Ex: x = zeros(3,2)
- ones : creates an array of all ones, Ex: x = ones(2)
- size : returns array dimensions
- length : returns length of a vector (row or column)

Elementary Math Functions

- abs - finds absolute value of all elements in the matrix
  sign - signum function
  sin,cos,… - Trignometric functions
  asin,acos… - Inverse trignometric functions
  exp - Exponential
  log,log10 - natural logarithm, logarithm (base 10)
  ceil,floor - round towards +infinity, -infinity respectively
  round - round towards nearest integer
  real,imag - real and imaginary part of a complex matrix
  sort - sort elements in ascending order
  sum,prod - summation and product of elements
  max,min - maximum and minimum of arrays
  mean,median - average and median of arrays
- *and many more…*

Flow Control

- MATLAB has five flow control statements

  - if statements
  - switch statements
  - for loops
  - while loops
  - break statements

‘if’ statement

- Example 1:

```
>> if i == j
>>     a(i,j) = 2;
>> elseif i >= j
>>     a(i,j) = 1;
>> else
>>     a(i,j) = 0;
>> end
```

- Example 2:

```
>> if (attn>0.9)&(grade>60)
>>     pass = 1;
>> end
```

‘for’ loop

- Example 1:

```
>> for x = 0:0.05:1
>>    printf('%d\n',x);
>> end
```

- Example 2:

```
>> a = zeros(n,m);
>> for i = 1:n
>>    for j = 1:m
>>       a(i,j) = 1/(i+j);
>>    end
>> end
```

‘while’ loop

- Example 1:

```
>> n = 1;
>> y = zeros(1,10);
>> while n <= 10
>>    y(n) = 2*n/(n+1);
>>    n = n+1;
>> end
```

- Example 2:

```
>> x = 1;
>> while x
>>    %execute statements
>> end
```

- Note: In MATLAB ‘1’ is synonymous to TRUE and ‘0’ is synonymous to ‘FALSE’

Scripts and Functions

- M-files are used to write blocks of codes to be executed at once.
- There are two kinds of M-files:
  - Scripts, which do not accept input arguments or return output arguments. They operate on data in the workspace
  - Functions, which can accept input arguments and return output arguments. Internal variables are local to the function

Functions in Matlab

- Example:
  - A file called stat.m:

```
function [mean, stdev]=stat(x)
%STAT Interesting statistics.
n=length(x);
mean=sum(x)/n;
stdev=sqrt(sum((x-mean).^2)/n);
```

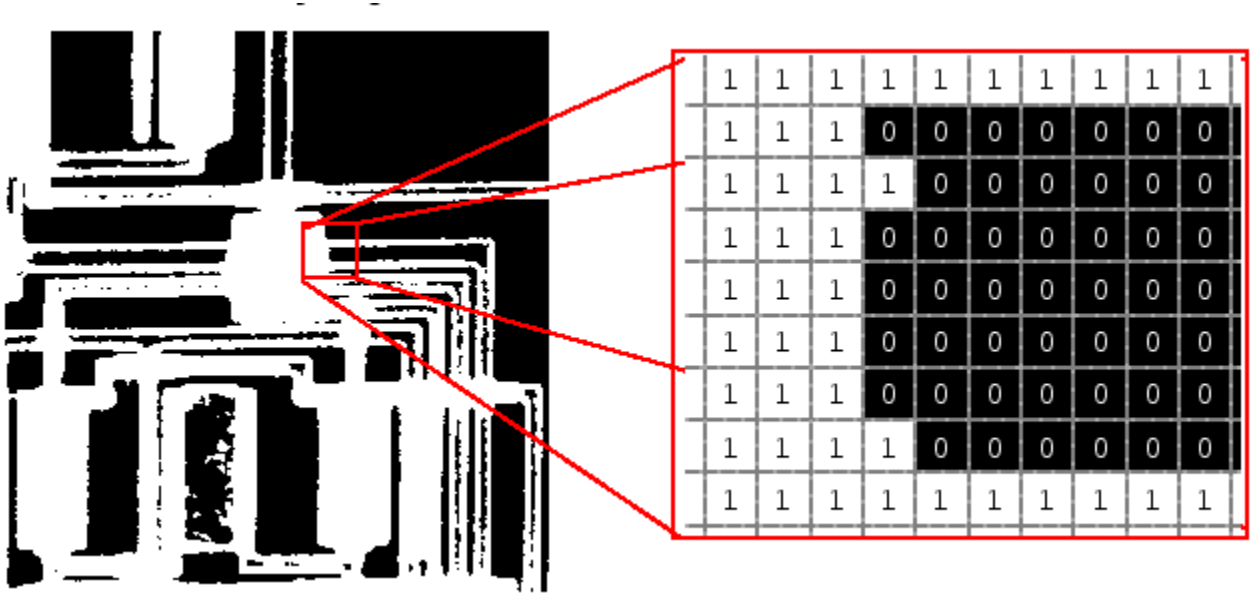
  - Defines a new function called "stat" that calculates the mean and standard deviation of a vector. Function name and file name should be the SAME!

What is the Image Processing Toolbox?

- The Image Processing Toolbox is a collection of functions that extend the capabilities of the MATLAB's numeric computing environment. The toolbox supports a wide range of image processing operations, including:
  - Geometric operations
  - Neighborhood and block operations
  - Linear filtering and filter design
  - Transforms
  - Image analysis and enhancement
  - Binary image operations
  - Region of interest operations

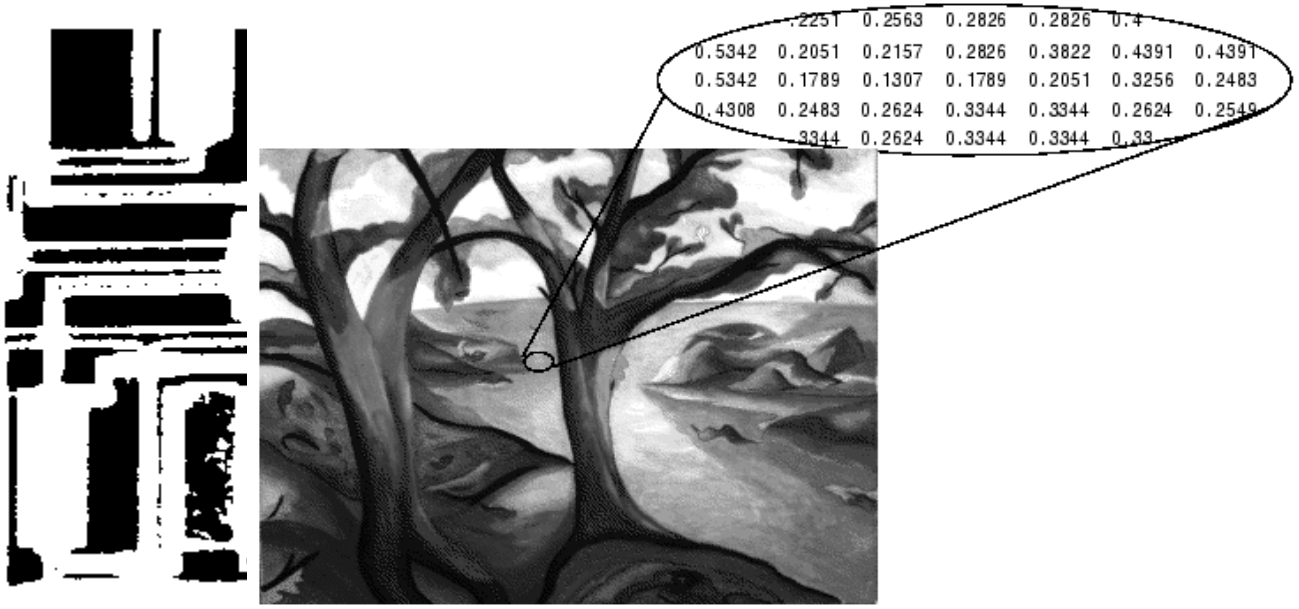
Images in Matlab

- Binary images : {0,1}
- Intensity images : [0,1] or `uint8`, `double` etc.
- RGB images : m × n × 3
- Multidimensional images: m × n × p (p is the number of layers)



Images in Matlab

- Binary images : {0,1}
- Intensity images : [0,1] or `uint8`, `double` etc.
- RGB images : m × n × 3
- Multidimensional images: m × n × p (p is the number of layers)



Images in Matlab

- Binary images : {0,1}
- Intensity images : [0,1] or `uint8`, `double` etc.
- RGB images : m × n × 3
- Multidimensional images: m × n × p (p is the number of layers)

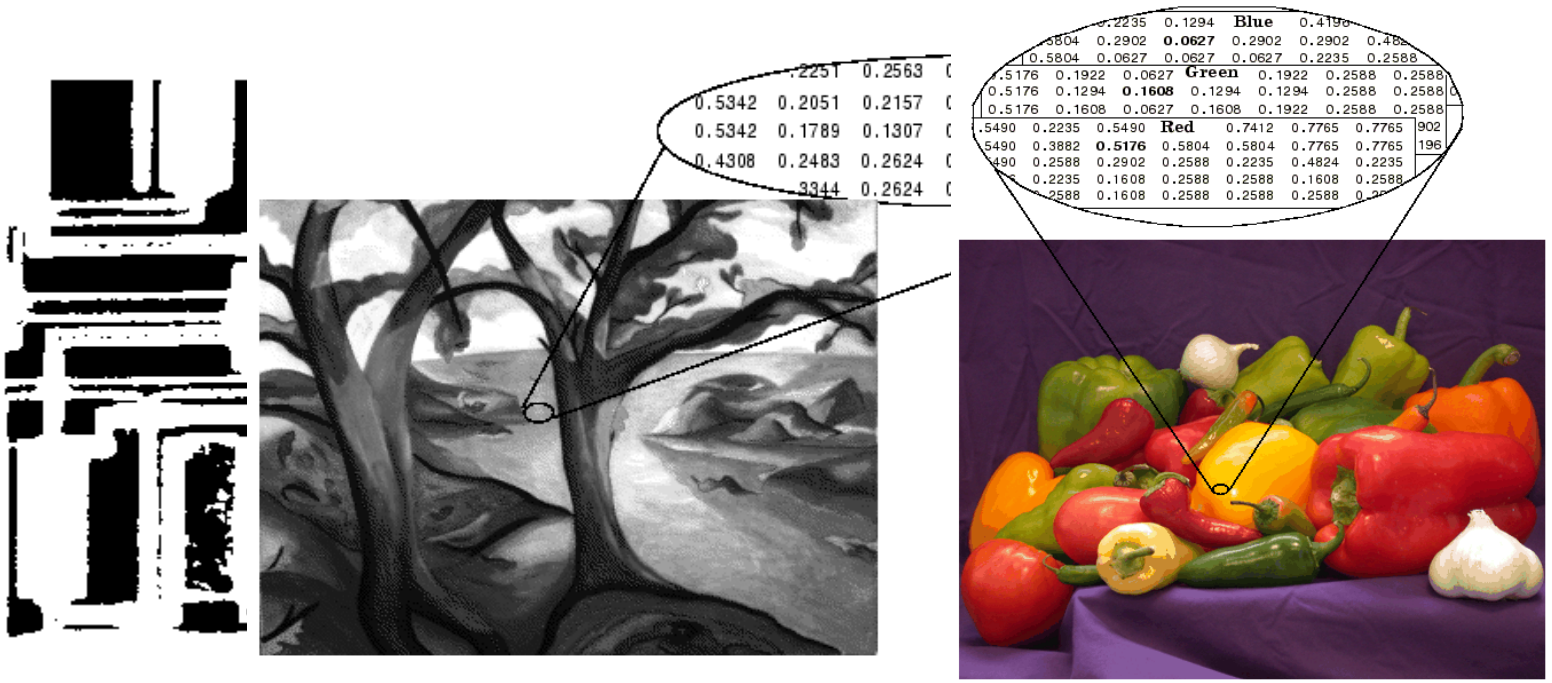


Image Import and Export

- Read and write images in Matlab

```
img = imread('apple.jpg');
dim = size(img);
imshow(img);
imwrite(img, 'output.bmp', 'bmp');
```

- Alternatives to `imshow`

```
imagesc(I)
imtool(I)
image(I)
```

Images & Matrices

- **How to build a matrix (or image)?**
- **Intensity Image:**

```
row = 256;
col = 256;
img = zeros(row, col);
img(100:105, :) = 0.5;
img(:, 100:105) = 1;
figure;
imshow(img);
```

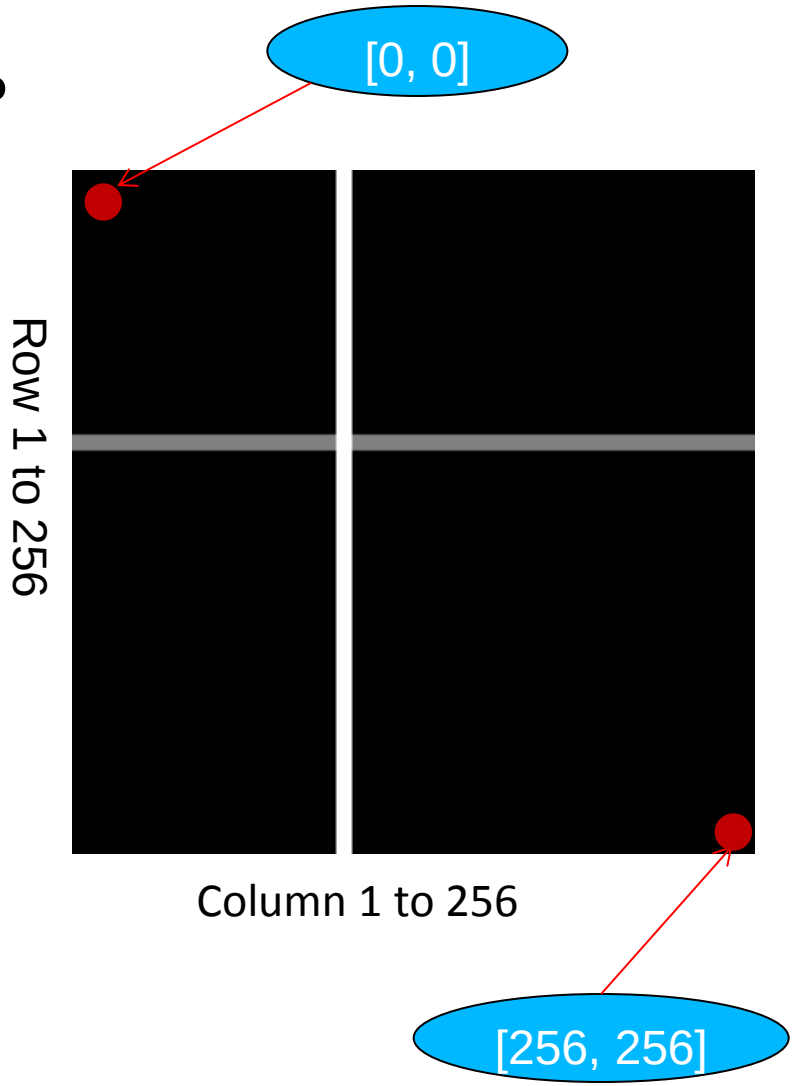


Image Display

- `image` - create and display image object
- `imagesc` - scale and display as image
- `imshow` - display image
- `colorbar` - display colorbar
- `getimage` - get image data from axes
- `truesize` - adjust display size of image
- `zoom` - zoom in and zoom out of 2D plot

Image Conversion

- `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