

# CSE 455 : Computer Vision

MATLAB 101

Getting Started with MATLAB

# Why?

- All assignments are going to be in MATLAB
- Interactive programming environment, easy manipulation of image data, allows rapid prototyping

# Getting Started

The image displays the MATLAB 7.7.0 (R2008b) interface. The main window is titled "MATLAB 7.7.0 (R2008b)" and has a menu bar with "File", "Edit", "Debug", "Parallel", "Desktop", "Window", and "Help". The "Current Directory" is set to "C:\Acads\CSE455".

Four panels are highlighted with blue rounded rectangles:

- Files in current directory:** A File Explorer window showing the contents of "C:\Acads\CSE455". It lists two files: "CSE 455.pptx" (modified 1/3/10 2:04 PM) and "~\CSE 455.pptx" (modified 1/3/10 2:00 PM).
- Command Prompt:** A Command Window showing the MATLAB prompt "fx >> |".
- Workspace:** A Workspace window showing a table with columns "Name", "Value", and "Min". It is currently empty.
- Command History:** A Command History window showing a list of commands entered in the Command Window, including "cd ..", "ls", "cd Acads", "cd CSE455", "clear", and "clrscr".

The bottom of the screen shows the Windows Start button and the system tray with "OVR" and a clock.

# Getting Started

- Common operators: +, -, /, \*
- Variables:
  - Assigned using =
  - There is no need to explicitly define the type of variable

# Control of Flow

- if statements and loops

If <logical expression>

    <statements>

end

for <var> = <start\_exp>:<end\_exp>

    <statements>

end

# Matrices

- MATLAB's power lies in efficiently manipulating matrices

```
g $r XG @M b < 0
+ M+ Zs CX 0 j
6 L2 e {( M K
N L? s2 = { $ P 8
z 4< &f 8 t
^ ( e i L Q
( O 5 i 6 Y X] zA & 0 5 ; * * &
W > t W = N 8 $ L : ; - I W r u
^ & F ] ; O 4 ; b h
; d :
M$ j+ Éa Pu
$ r XG @M b < 0
+ M+ Zs CX 0 j
6 L2 e {( M K
N L? s2 = { $ P 8
z 4< &f 8 t
^ ( e i L Q
( O 5 i 6 Y X] zA & 0 5 ; * * &
W > t W = N 8 $ L : ; - I W r u
^ & F ] ; O 4 ; b h
; d :
M$ j+ Éa Pu
```

# Matrices

- Initializing matrices
  - >> A = zeros(10,10)
  - >> A = zeros(10)
  - >> A = eye(10,10)
  - >> A = [1 2 3;4 5 6]
  - >> A = zeros(10,10,10)
- Accessing matrix elements
  - A(2,1) : 2<sup>nd</sup> row, 1<sup>st</sup> column of A
  - A(:,1) : 1<sup>st</sup> column of A
  - A(1:10,:) : first 10 rows and all columns of A

# Manipulating Matrices

- $+$ ,  $-$ ,  $*$ ,  $/$  can be applied to matrices (provided dimensions agree)
- Element wise operations:
  - $.*$  : Element wise multiplication
  - $./$  : Element wise division
- Transposing a matrix:  $A'$



# Manipulating Matrices

- Some very useful operations:
  - $B = \text{reshape}(A, m, n)$  : Takes a matrix  $A$  with  $m * n$  elements and reshapes it into a matrix with  $m$  rows and  $n$  columns

1	4	7	10
2	5	8	11
3	6	9	12

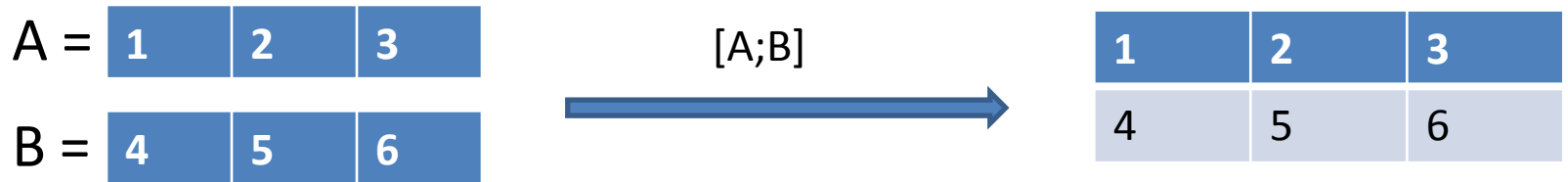


Reshape(A,2,6)

1	3	5	7	9	11
2	4	6	8	10	12

# Manipulating Matrices

- Some very useful operations:
  - Concatenating matrices:



# Manipulating Matrices

- Some very useful operations:
  - Concatenating matrices:

A = 

1	2	3
---	---	---

B = 

4	5	6
---	---	---

cat(1,A,B)



1	2	3
4	5	6

A = 

1
2
3

B = 

4
5
6

cat(2,A,B)




1	4
2	5
3	6

- Useful for concatenating along higher dimensions

# Manipulating Matrices

- Some very useful operations:
  - `repmat(A,m,n)` : Repeats A, m times along rows and n times along columns

1	2
3	4

`repmat(A,2,3)` 

1	2	1	2	1	2
3	4	3	4	3	4
1	2	1	2	1	2
3	4	3	4	3	4

# Manipulating Matrices

- *find*

```
>> inds = find(A>0);
```

- *sum*

```
>> sum(A,1); % sum A along first dimension
```

- *mean, var, etc.*

# Manipulating Matrices

- Summary:
  - Operators: + , - , / , \* , .\* , ./
  - reshape
  - cat
  - repmat
  - find, sum, mean, etc.

# MATLAB Help

- `help <function_name>`

OR

- `doc <function_name>`

# Matlab Scripts: m-files

- Save a sequence of MATLAB commands as a script
- MATLAB has a built-in editor which can be invoked using the *edit* command



# Matlab Functions

- Function name same as the filename
- Header of a function file:  
`function <retval> = <function_name>(arglist)`
- MATLAB will recognize all function files in the working directory. Additional directories may be added to the path.

# Debugging in MATLAB

- Demo

# Images in MATLAB

- Loading an image  
    >> I = imread('filename');
- Image is represented as a  $H \times W \times 3$  matrix
- `imagesc(I)` displays the image
- Saving images: `imwrite(I,'filename');`

# Image Filtering Example

Input Image					
1	0	1	2	7	8
0	5	0	3	6	9
0	3	0	0	6	1
0	3	0	6	7	8
6	6	5	5	4	4

Filtered Image					

Take weighted sum of values in the box, weights specified by the filter  
 $1*1 + 1*0 + 1*1 + 1*0 + 2*5 + 1*0 + 1*0 + 1*3 + 1*0 = 15$

Filter		
1	1	1
1	2	1
1	1	1

# Image Filtering Example

	Input Image					
1	0	1	2	7	8	
0	5	0	3	6	9	
0	3	0	0	6	1	
0	3	0	6	7	8	
6	6	5	5	4	4	

Filtered Image					

Assume zero values outside the boundary.

Filter		
1	1	1
1	2	1
1	1	1

# MATLAB Demo

# MATLAB Tips

- MATLAB is slow

# MATLAB Tips

- MATLAB can be slow
- Dynamic allocation is evil
- For loops are evil



# Data Visualization

- plot, bar, hist, scatter
- surf/mesh