Designing Applications that See
Lecture 4: Matlab Tutorial

Dan Maynes-Aminzade

23 January 2007

Reminders

- Assignment #1 due now!
- Assignment #2 released today, due in one week
- All the readings are now available, linked from course calendar

Today's Goals

- Take the techniques covered in the last lecture and learn how to use them in Matlab
- Work through the process of building a complete example of a simple computer vision application

Image Processing in Matlab

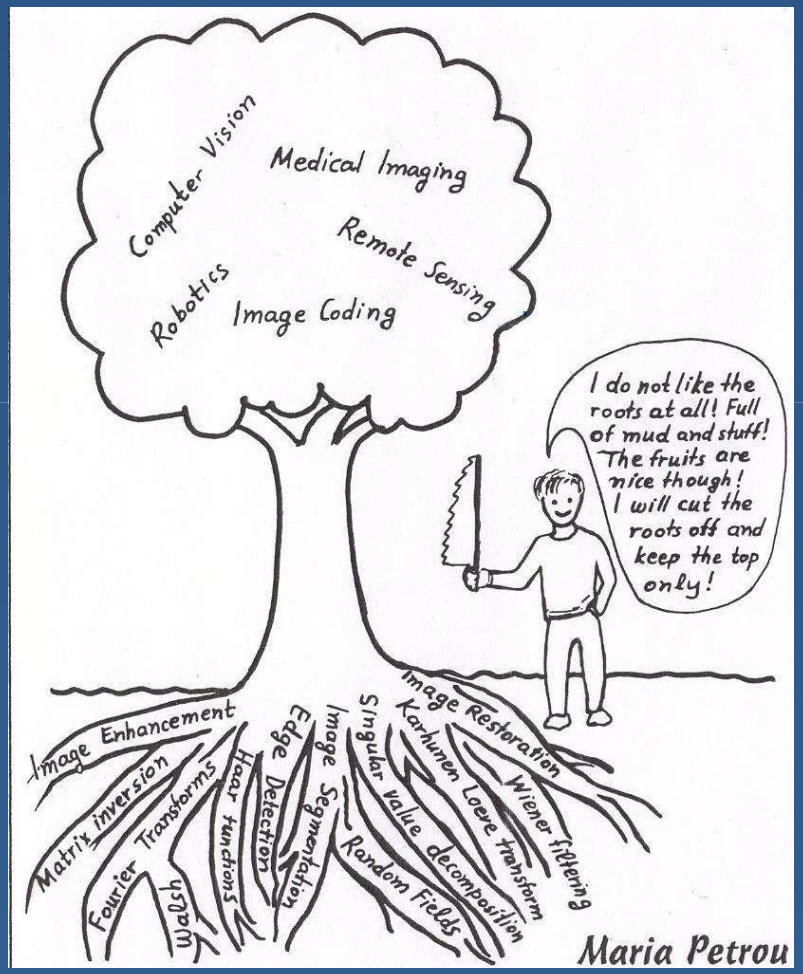


Image Conversion

`rgb2gray`

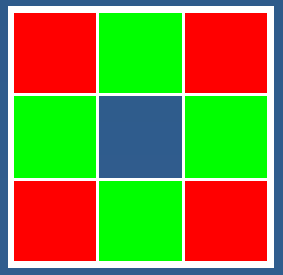
`im2bw`

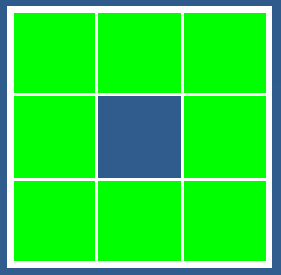
Dilation and Erosion

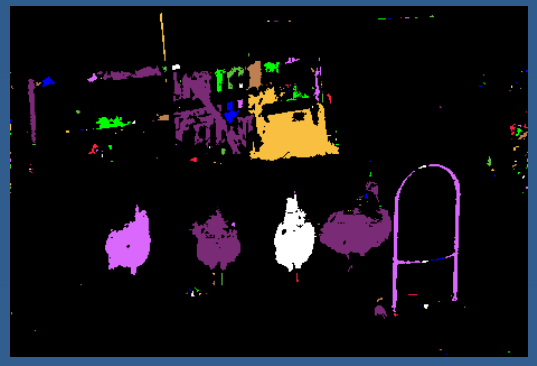
`imerode`

`imdilate`

Connected Components







`bwfill, bwselect`

Linear Filtering

10	5	3
4	5	1
1	1	7

*

0	0	0
0	0.5	0
0	1.0	0.5

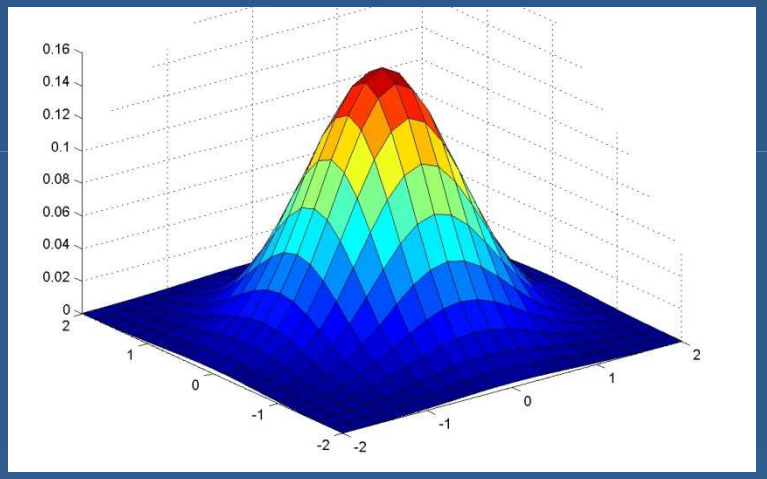
kernel

=

	7	

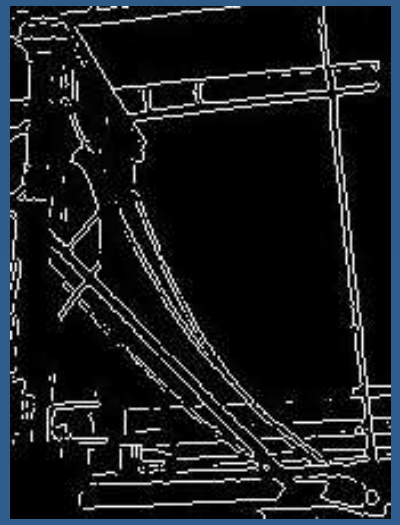
`imfilter, filter2`

Gaussian Kernel



`fspecial('gaussian',...)`

Sobel Edge Detection



1	0	-1
2	0	-2
1	0	-1

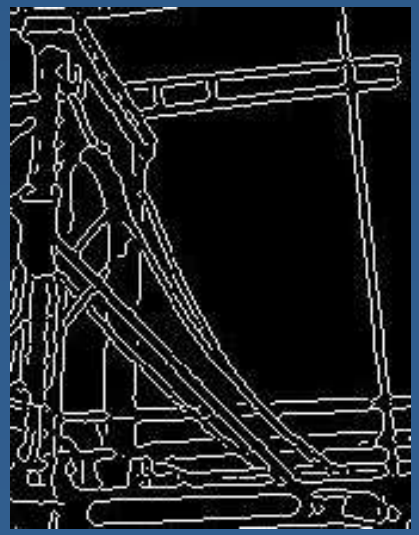
Sobel x

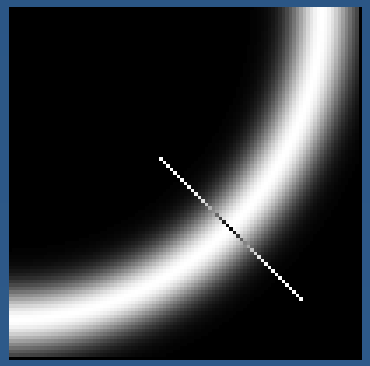
1	2	1
0	0	0
-1	-2	-1

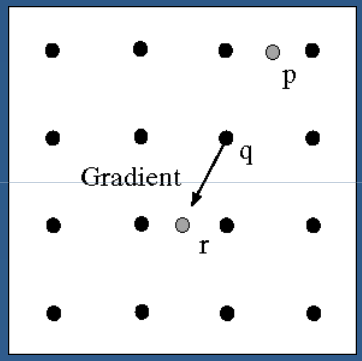
Sobel y

```
edge(I, 'sobel')
```

Canny Edge Detection

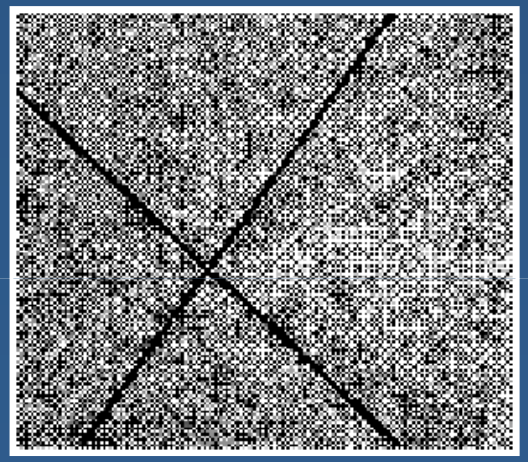






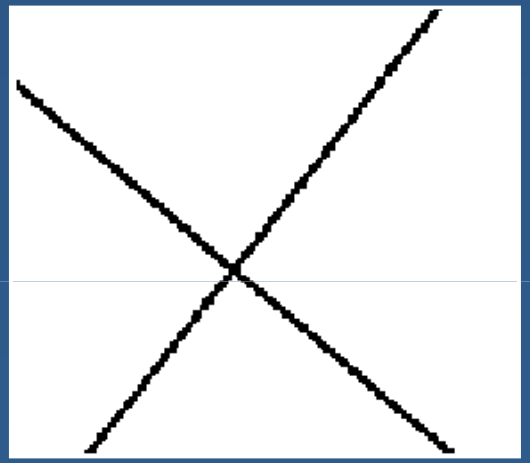

```
edge(I, 'canny')
```

Hough Transform



Image

Edge detection



Hough Transform

```
houghlines(BW,theta, rho, peaks)
```

Outline

- Matlab fundamentals*
- Walkthrough of developing a computer vision application in Matlab*
  - Designing an image processing algorithm
  - Building a GUI
  - Running on live video
  - Deploying an application

*Based on slides by Christopher Rasmussen (University of Delaware)

*Based on "Image Processing" seminar by Bruce Tannenbaum (MathWorks, Inc.)

What is Matlab?

- A high-level language for matrix calculations, numerical analysis, & scientific computing
- Language features
  - No variable declarations
  - Automatic memory management (but preallocation helps)
  - Variable argument lists control function behavior
  - Vectorized: Can use `for` loops, but largely unnecessary (and less efficient)

Need Matlab Help?

- In Matlab
  - Highlight a term, right-click, and select “help”
  - Type “`help`” to get a listing of topics
  - “`help <topic>`” gets help for that topic
- On the web
  - CS377S Resources page has links
  - In particular, the MathWorks help desk:

`www.mathworks.com/access/helpdesk/help/helpdesk.shtml`

Entering Variables

- Entering a vector, matrix
  - `V = [10, 4.5, 1];`
  - `M = [3, 4 ; -6, 5];`
- Without semi-colon, input is echoed (this is bad when you're loading images!)
- Comma to separate statements on same line
- size: Number of rows, columns

Constructing Matrices

- Basic built-ins:
  - All zeroes, ones: `zeros`, `ones`
  - Identity: `eye`
  - Random: `rand` (uniform), `randn` (unit normal)
- Ranges: `m:n`, `m:i:n` (`i` is step size)
- Composing big matrices out of small matrix blocks
- `repmat(A, m, n)`: "Tile" a big matrix with `m` x `n` copies of `A`

Multiplications & Calculations

- Transpose (‘), inverse (`inv`)
- Matrix arithmetic: `+`, `-`, `*`, `/`, `^`
- Elementwise arithmetic: `.*`, `./`, `.^`
- Functions
  - Vectorized
  - `sin`, `cos`, etc.

Deconstructing Matrices

- Indexing individual entries by row, col: `A(1, 1)` is upper-left entry
- Ranges: e.g., `A(1:10, 3)`, `A(:, 1)`
- Matrix to vector and vice versa by column: `B = A(:)`, `A(:) = B`
  - Transpose to use row order
- `find`: Indices of non-zero elements

Matrix Analysis

- Basics (by column)
  - `norm`
  - `max`, `min`
  - `sum`
- More advanced
  - Linear systems: `A\b` solves `A*x = b`
  - QR decomposition: `qr`
  - Singular value decomposition: `svd`
  - Eigenvalues: `eig`
  - Etc.

Control Structures

- Expressions, relations (==, >, |, &, functions, etc.)
- `if/while` *expression statements* `end`
  - Use comma to separate expression from statements if on same line
  - `if a == b & isprime(n), M = inv(K); else M = K; end`
- `for` *variable* = *expression statements* `end`
  - `for i=1:2:100, s = s / 10; end`

The M-Files

- Any text file ending in ".m"
- Use `path` or `addpath` to tell Matlab where code is (or select in directory window)
- Script: Collection of command line statements
- Function: Take argument(s), return value(s). First line defines:
  - `function y = foo(A)`
  - `function [x, y] = foo2(a, M, N)`
- Comment: Start line with %

Plotting

- 2-D vectors: `plot(x, y)`
  - `plot(0:0.01:2*pi, sin(0:0.01:2*pi))`
- 3-D: `plot3(x, y, z)` (space curve)
- Surfaces
  - `meshgrid` makes surface from axes, `mesh` plots it
    - ```
      [X,Y] = meshgrid(-2:.2:2, -2:.2:2);
      Z = X .* exp(-X.^2 - Y.^2);
      mesh(Z)
      ```
  - `surf`: Solid version of `mesh`
- Saving figures, plots: `print -depsc2` *filename*
      ```

Image Processing Toolbox

- Loading, displaying images:
  `I=imread('im1.jpg'), imshow(I)`
- Saving images:
  `imwrite(I, 'newim.jpg')`
- Image representation
  - Grayscale: Matrix of `uint8`
  - Color: Stack of 3 matrices for R, G, and B
- Conversion: `I2 = double(I1)`

Building an Example Application

- Image analysis with the Matlab Image Processing Toolbox
- Getting live data with the Matlab Image Acquisition Toolbox
- Building a GUI with GUIDE
- Deploying an application with the Matlab compiler
- Try to follow along!

Matlab Workflow

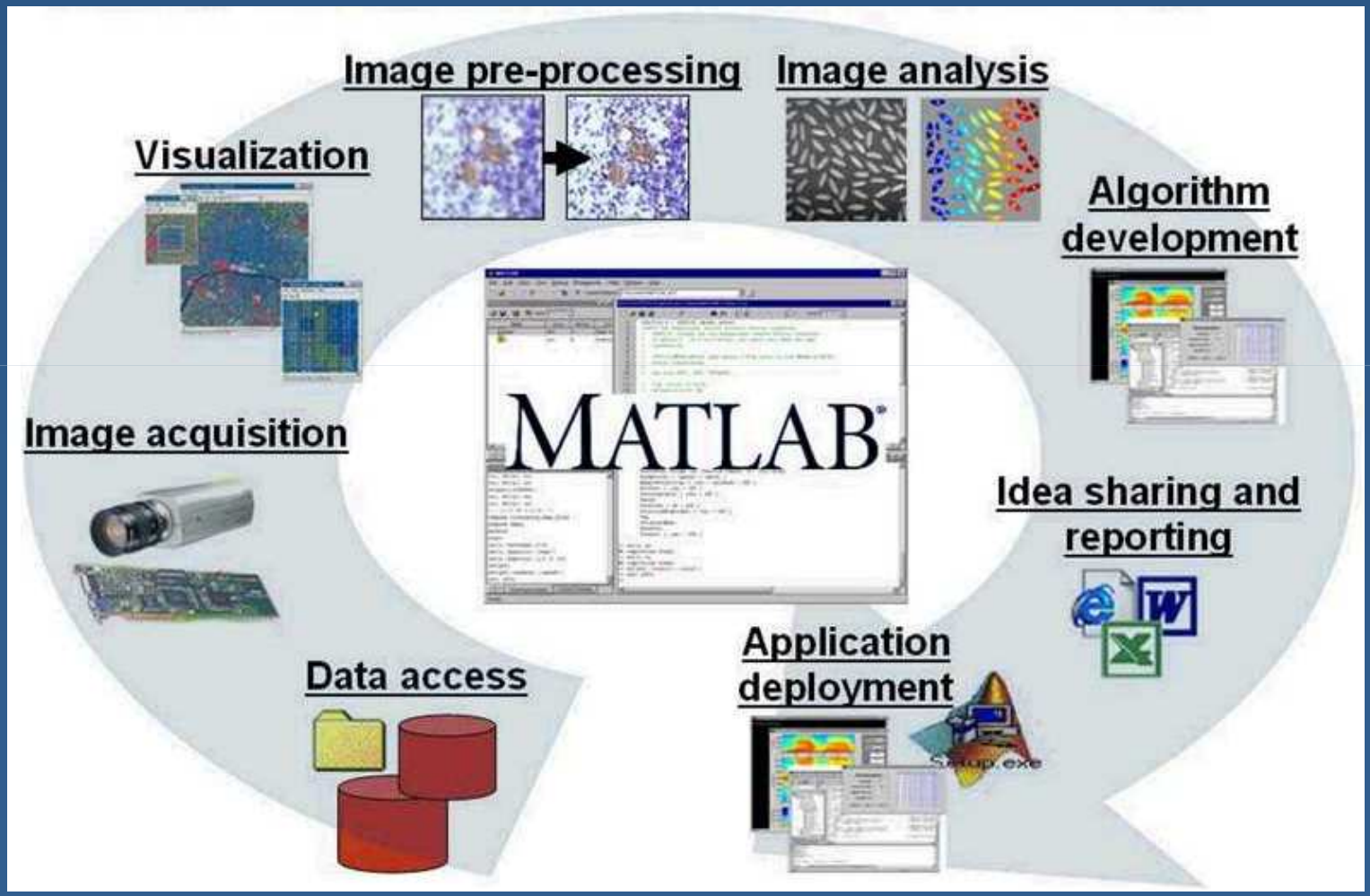
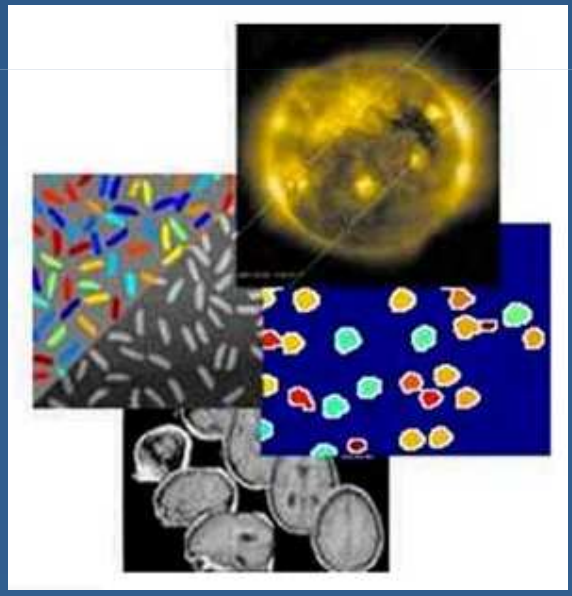


Image Processing Toolbox

- Image visualization
- Image pre- and post-processing
- Image analysis
- Spatial transformations
- Color processing



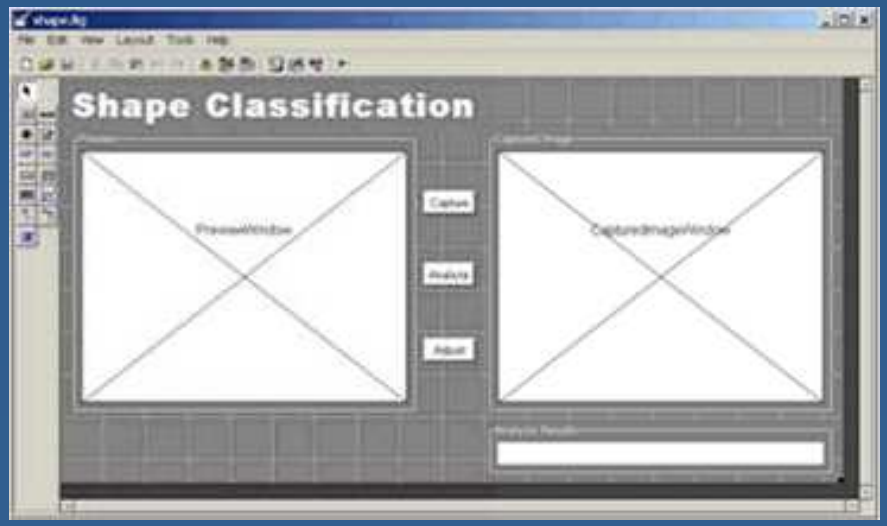
Traditional Image Processing Tasks

Image Acquisition Toolbox

- Stream video and images into Matlab
- Supports a wide variety of frame grabbers and digital cameras
- Configure device properties
- Live video previewing
- Background image acquisition

Designing a GUI with GUIDE

- Design and edit GUI
- Add buttons, pull-down menus, etc.
- Generate Matlab code
- Finish the code yourself

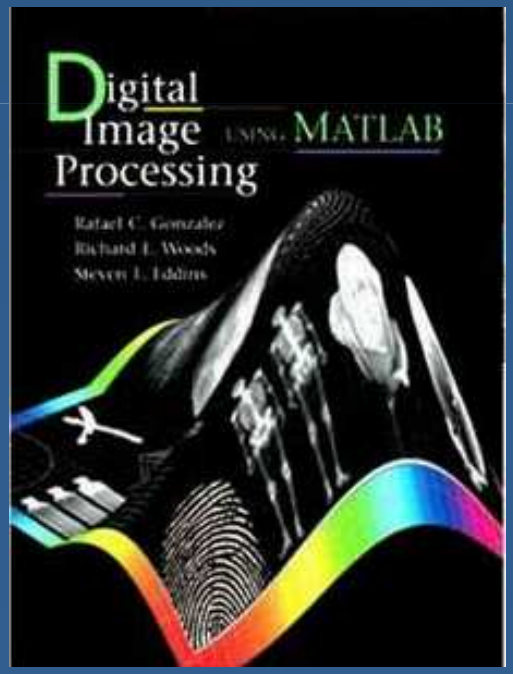


Nice Things about Matlab

- Unified environment
- Quick iteration through different algorithms
- Interactive graphics and visualizations
- High level language
- Lots of built-in routines, useful Toolbox functions, and code available on the web

To Learn More...

- *Digital Image Processing Using Matlab* by Gonzalez, Woods, and Eddins



Tutorial Files

- Download the tutorial files:

http://cs377s.stanford.edu/code/matlab-tutorial.zip

- Copy them to your Matlab working directory (probably `C:\MATLAB701\work`)