University of Massachusetts Lowell

Structural Engineering Research Group (SERG)
Summer Seminar Series #15
August 22, 2014

Image Processing using Matlab

Tzuyang Yu

Associate Professor, Ph.D.
Structural Engineering Research Group (SERG)
Department of Civil and Environmental Engineering
The University of Massachusetts Lowell

SERG

Outline

- Introduction
- What is Matlab?
- Matlab Basics
- Image Processing in Matlab
- Summary
- References

Introduction

- Image Processing generally involves extraction of useful information (qualitative and/or quantitative) from an image.

- This useful information may be the dimensions of an engineering component, size of diagnosed tumor, a three-dimensional rendering of an unborn baby, spatial distribution (e.g., length, shape) of a surface crack, area of a corroded steel rebar, or volume of a crumbled concrete.

What is Matlab?

- Matlab is an abbreviation of Matrix Laboratory by MathWork®.
- Scientific and engineering standard for mathematical processing and programming → Large user group around the world
- In Matlab, everything is represented in the form of arrays or matrices.à◊Linear Algebra
- Code developed in Matlab can be converted into C, C++ or Visual C++. → Portability, flexibility
- Matlab codes may be called as ActiveX Object from higher level languages like Visual Basic. → Expendability

Matlab Basics

- Matrix Declaration:
  - Null Matrix → *null*
  - Matrix with Ones → *ones(m, n)*
  - Identity Matrix → *eye(n)*
  - Random Matrix → *randn(n)*

- Matrix Arithmetic:
  - Addition
  - Subtraction
  - Multiplication
  - Division

Matlab Basics

- Matrix manipulation:
  - Addressing of individual element → *A(i, j)*
  - Complete row addressing → *A(:, n)*
  - Complete column addressing → *A(n, :)*
  - Transpose → *A = B'*
  - Flip matrices → *fliplr* and *flipud*
- Saving and loading data → *save* and *load*
- Concept of function and m-files
- Concept of path → *cd*, *pwd*

Matlab Basics

- Useful commands:
  - *whos*
  - *help → Learn how to use help to teach yourself Matlab commands.*
  - *clear*
  - *path*
  - *cd*
  - *dir*
  - *lookfor*

Image Processing in Matlab

- Images can be conveniently represented as matrices in Matlab.
- Images can be conveniently represented as matrices (2D or 3D) in Matlab.
- The matrix may simply be *m x n* form or it may be 3D array or it may be an indexed matrix, depending upon image type.
- Image processing is carried out by matrix calculation or matrix manipulation.
- Image can be read by using *imread* command, displayed with *imshow* command, and saved with *imwrite* command.

Image Processing in Matlab

- Generally speadking, images can be of three types; black & white (BW), grey scale, and colored (Red-Green-Blue or **RGB**).

- In Matlab, there are the following types:
  - Black & White images are called binary images, containing 1 for white and 0 for black.
  - Grey scale images are called intensity images, containing numbers in the range of 0 to 255 or 0 to 1.
  - Colored images may be represented as RGB Image or Indexed Image.

Image Processing in Matlab

- In **RGB** images there exist three indexed images.

- First image contains all the red portion of the image, second green and third contains the blue portion. → *Therefore for a 640 x 480 sized image the matrix will be 640 x 480 x 3.*

- An alternate method of colored image representation is Indexed Image. → *Each color in the image is given an index number and in image matrix each color is represented as an index number.*

- Map matrix contains the database of which index number belongs to which color.

Image Processing in Matlab

- Read in an image.

- Validates the graphic format.

  (bmp, hdf, jpeg, pcx, png, tiff, xwd)

- Store it in an array.

```
clear, close all
I = imread(‘pout.tif`);
[X, map] = imread(‘pout.tif’);
```

Image Processing in Matlab

- RGB Image to Intensity Image (*rgb2gray*)
- RGB Image to Indexed Image (*rgb2ind*)
- RGB Image to Binary Image (*im2bw*)
- Indexed Image to RGB Image (*ind2rgb*)
- Indexed Image to Intensity Image (*ind2gray*)
- Indexed Image to Binary Image (*im2bw*)
- Intensity Image to Indexed Image (*gray2ind*)
- Intensity Image to Binary Image (*im2bw*)
- Intensity Image to RGB Image (*gray2ind*, *ind2rgb*)

Image Processing in Matlab

- There are a number of ways to get statistical information about data in the image. → Image histogram is on such way.
- An image histogram is a chart that shows the distribution of intensities in an image.
- Each color level is represented as a point on x-axis and on y-axis is the number instances a color level repeats in the image.
- Histogram may be view with *imhist* command.
- Sometimes all the important information in an image lies only in a small region of colors, hence it usually is difficult to extract information out of that image.
- To balance the brightness level, one can perform histogram equalization.

Image Processing in Matlab

- In image processing useful pixels in the image are separated from the rest by image segmentation.

- *Brightness threshold* and *edge detection* are the two most common image segregation techniques.

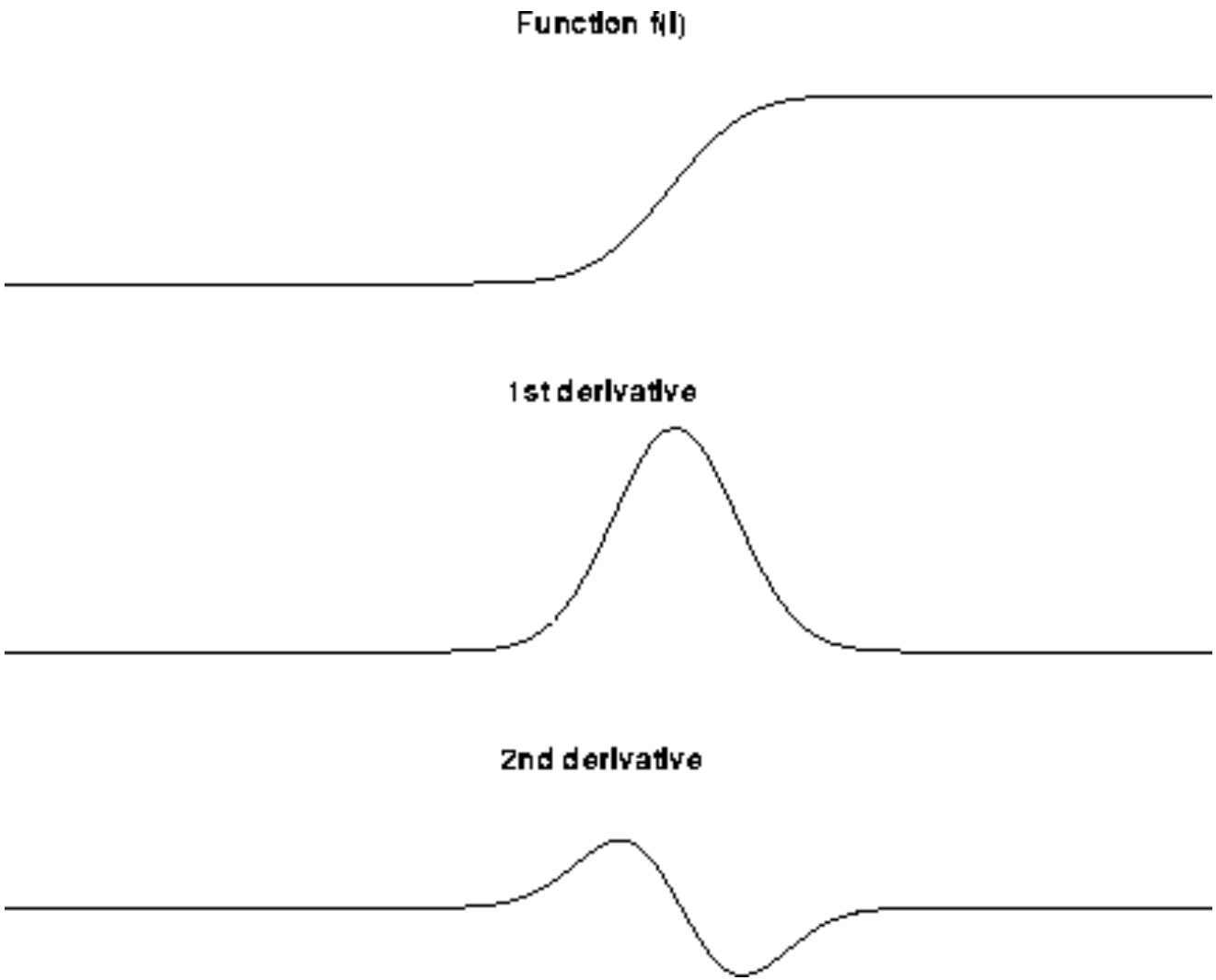
- In brightness threshold, all the pixels brighter than a specified brightness level are taken as 1 and rest are left 0. → *This leads to a binary image with useful image as 1 and unwanted as 0.*

- In edge detection special algorithms are used to detect edges of objects in the image. → *edge*

Edge Detection

- Edge detection extract edges of objects from an image.
- There are a number of algorithms for this, but these may be classified as derivative based or gradient based.
- In derivative based edge detection the algorithm takes first or second derivative on each pixel in the image. → *In case of first derivative at the edge of the image there is a rapid change of intensity.*
- While in case of second derivative there is a zero pixel value, termed zero crossing.
- In gradient based edge detection a gradient of consecutive pixels is taken in x and y direction.

Edge Detection



Edge Detection

- Taking derivative on each and every pixel of the image consumes a lot of computer resources and is not practical.
  → *Typically an operation called kernel operation is carried out.*

- A kernel is a small matrix sliding over the image matrix containing coefficients which are multiplied to corresponding image matrix elements and their sum is put at the target pixel.

Sobel Edge Detection

- In *sobel* following formulas are applied on each pixel *(i,j)* in the image and two matrices S_x and S_y are obtained:

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$
$$C = 2$$

- Alternatively this can be done by applying following two kernels:

$S_x =$

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

$S_y =$

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

- The resultant matrix is then obtained by taking the square root of the sum of the squares of S_x and S_y, as follows:

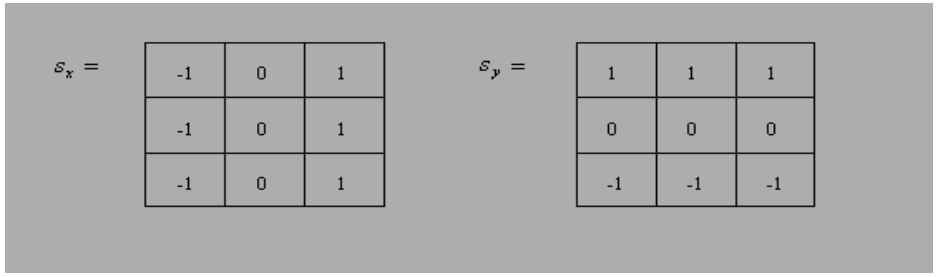
$$M(i,j) = (S_x^2 + S_y^2)^{1/2}$$

Prewitt Edge Detection

- In *Prewitt* everything is same but $C=1$:

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$
$$C = 1$$

- Alternatively this can be done by applying following two kernels:



- The resultant matrix is then obtained by taking the square root of the sum of the squares of S_x and S_y, as follows:

$$M(i,j) = (S_x^2 + S_y^2)^{1/2}$$

Morphological Operations

- These are image processing operations done on binary images based on certain morphologies or shapes.
- The value of each pixel in the output is based on the corresponding input pixel and its neighbors.
- By choosing appropriately shaped neighbors one can construct an operation that is sensitive to a certain shape in the input image.

Morphological Operation Skeletonize

- It creates skeleton of an object, by removing pixels on the boundaries but does not allow objects to break apart.
- It is an extremely important operation in image processing as it removes complexities from an image without loosing details.

Erosion & Dilation

- These are the most fundamental of binary morphological operations.
- In dilation if any pixel in the input pixel's neighborhood is on, the output pixel is on otherwise off.
- In actual dilation grows the area of the object. Small holes in the object are removed.
- In erosion if every pixel in the input pixel's neighborhood is on the output pixel is on otherwise off
- This in actual works as shrinking the object's area, thus small isolated regions disappear.

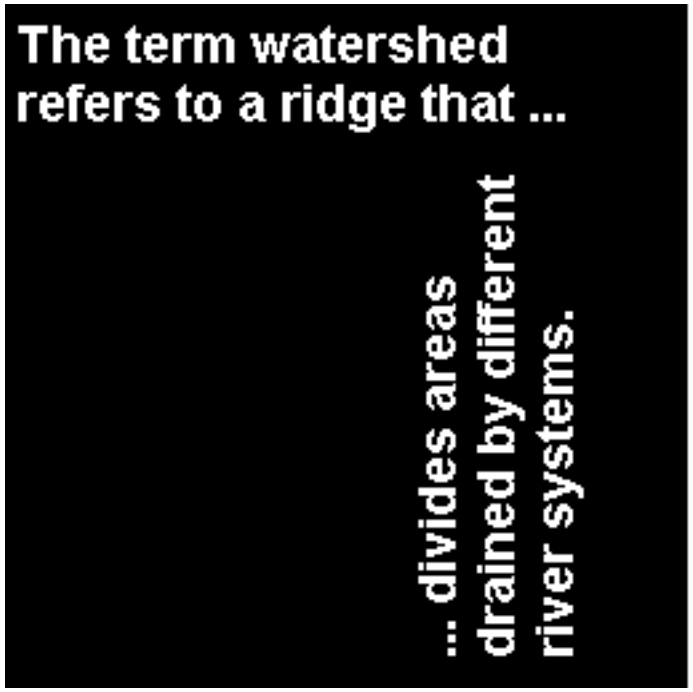
Simple Morphological Operations

- *imerode* – Erode image.
- *imclose* – Morphologically close image.
- *imopen* – Morphologically open image.
- *imdilate* – Dilate image.
- *strel* – Create morphological structuring element (STREL).

- *imread* – Read image from graphics file.
- *imwrite* – Write image to graphics file.
- *imshow* – Display image.
- *imsubtract* – Subtract one image from another or subtract constant from image.
- *imadjust* – Adjust image.
- *imcrop* – Crop image.

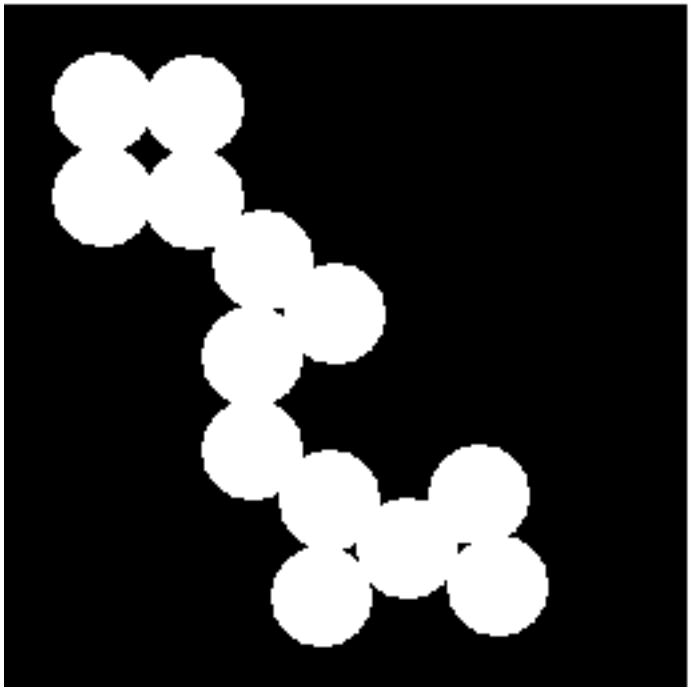
Erosion & Dilation

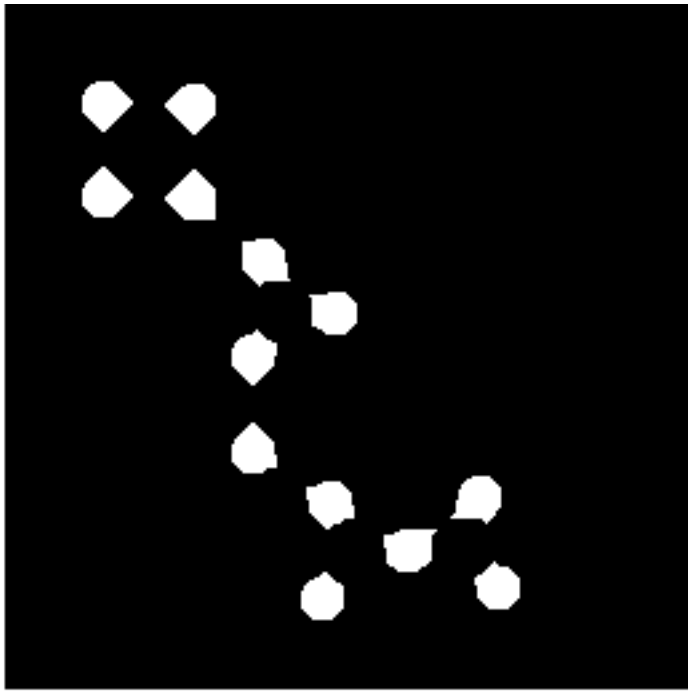
- *imdilate*



Erosion & Dilation

- *imerode*





Summary

- Imaging (rendering data into image) is an approach in data visualization. Mathematically, an image is a two-dimensional tensor.

- *Linear Algebra* is the mathematical basis in image processing.

- The objective of image processing is ***to reveal the physical pattern in the data, not to play with mathematical skills.***

References

- Gonzalez, R. C., R. E. Woods, and S. L. Eddins (2009), *Digital Image Processing Using MATLAB*, Gatesmark Publishing, Knoxville, TN.

- Haralick, R. M., and L. G. Shapiro (1992), *Computer and Robot Vision*, Vol. I, Addison-Wesley, pp. 158-205.

- van den Boomgard, R, and R. van Balen (1992), "Methods for Fast Morphological Image Transforms Using Bitmapped Images," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 54 (3), pp. 254-258.