ELIC 629
Digital Image Processing

Fall 2005

Smoothing and Sharpening Filters, Discontinuity Detection & Image Segmentation

Bill Kapralos

Monday, December 5 2005

Overview (1):

- **Filtering in the Frequency Domain**
  - Smoothing filters
  - Sharpening filters
- **Discontinuity Detection**
  - Introduction to image segmentation
  - Point detection
  - Line detection
  - Edge detection

Overview (2):

- **Thresholding**
  - Foundation
  - Introduction

Before We Begin

Administrative Details (1):

- **Lab Eight Today**
  - Final lab
  - No lab report required

Some Questions to Consider (1):

- Why filter in the frequency domain ?
- What are the steps to filtering an image in the frequency domain ?
- Why do we shift the origin of the DFT output ?
- Why do we scale (with an exponential function) the output of the Fourier output ?
- From the origin, what can we say about the DFT frequency ?
- What is a low/high pass frequency domain filter ?
- What is a "notch" filter ?

Administrative Details (2):

- **Exam Dec. 19 2005**
  - Review next week during lab period
    - I will make some comments regarding the exam
  - Exam will be similar in format to mid-term
    - No surprises!
  - Focus on material after mid-term but you are still responsible for all material
    - Still need to know spatial filtering in the frequency domain

Smoothing Frequency Domain Filters

Introduction (1):

- **What is a Smoothing Filter (Review)**
  - Edges, noise, sharp transitions in intensity levels lead to the majority of high frequency components in the frequency domain (e.g., Fourier transform)
  - Smoothing in the frequency domain is therefore achieved by (ideally) removing a specified range of high frequency components in the transform
    - Remember → ideally these components are removed but in practice, they are attenuated
    - Gaussian is one type of smoothing filter

Introduction (2):

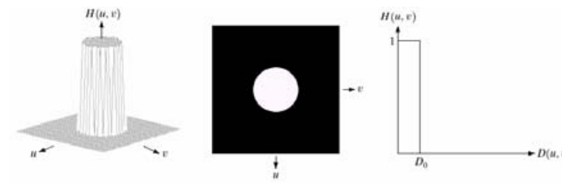
- ***Mathematically***

G[u,v] = H[u,v]F[u,v]

  - Recall
    - F[u,v] → Fourier transform of image to be filtered
    - H[u,v] → filter applied to image
    - G[u,v] → filtered image (output image)

Introduction (3):

- ***Graphical Illustration of Low Pass Filtering***



Ideal low-pass filter

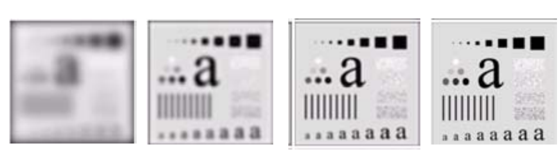
low pass filter displayed as an image

Filter radial cross-section where D_0 is radius of "circle" e.g., determines cur-off frequency

Introduction (2):

- ***Graphical Illustration of Low Pass Filtering***

Original image

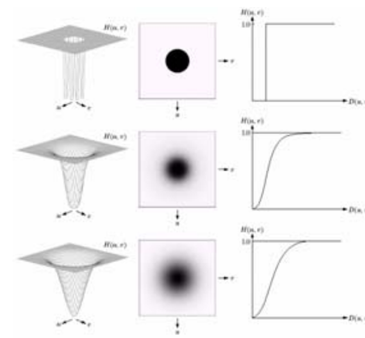


D_0 = 5 D_0 = 15 D_0 = 30 D_0 = 80

Sharpening Frequency Domain Filters

Introduction (1):

- **Graphical Illustration of High Pass Filtering**



Introduction (1):

- **What is a Sharpening Filter (Review)**
  - Removes (ideally) low frequency components of an image's Fourier representation (e.g., keeps frequency components above some cut-off frequency)
  - Basically, the reverse of the low pass filter and given mathematically by

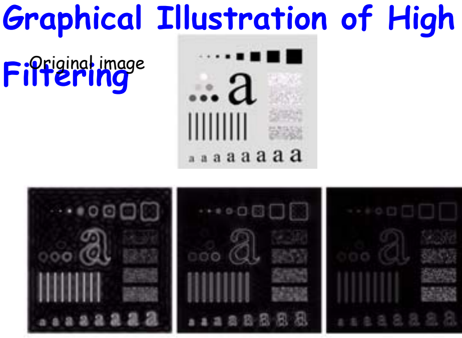
$$H_{hp}[u,v] = 1 - H_{lp}[u,v]$$

  - $H_{hp}[u,v] \rightarrow$ high pass filter
  - $H_{lp}[u,v] \rightarrow$ low pass filter

Introduction (2):

- **Graphical Illustration of High Pass Filtering**

Original image



D_0 = 15 D_0 = 30 D_0 = 80

Discontinuity Detection

Image Segmentation (2):

- **Image Segmentation Algorithms Generally Based on Two Basic Properties of Intensity**
  - Discontinuity
    - Partition image based on abrupt changes in intensity (e.g., edges where there is a large change in intensity between adjacent pixels)
  - Similarity
    - Partition image into regions that are similar based on some pre-defined criteria (e.g., intensity of pixels within a certain range)

Image Segmentation (1):

- **What is Image Segmentation ?**
  - Segmentation sub-divides an image into a number of regions or objects
  - How far this sub-division is carried out depends on the task
  - An extremely difficult yet important task
    - Its accuracy determines the eventual success or failure of any automated analysis procedure which rely on its output

Introduction (1):

- **Will Focus on Three Types of Discontinuities**
  1. Points
  2. Lines
  3. Edges

  - Regardless the type of discontinuity, most common approach to locating them is to "filter" the image with a 3 x 3 mask (e.g.,, convolution)
    - Mask coefficients are chosen depending on the type of discontinuity being searched for

Introduction (2):

- **Recall Spatial Domain Filtering with Mask**
  - Sum of products of coefficients with the gray levels in image encompassed by the mask

R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + … + w(0,0)f(x,y) +…+ w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

Example of a 3x3 template with its coefficients

Point Detection (2):

- **Basic Idea**
  - Isolated point (a point whose gray level is much different from its background) will be different from its surroundings and will be detected by the mask used
  - Examine mask coefficients
    - Sum of coefficients equals 0 → mask response will be zero in areas of constant gray level

Point Detection (1):

- ***In Principle, Straightforward***
  - Using the following mask, a point is detected at the location at which the mask is centered on if

$$|R| \geq T$$

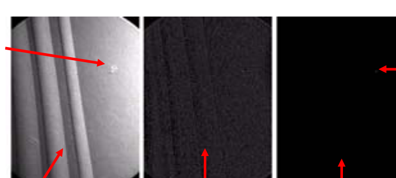
R → output of filtering operation (e.g., sum of filter coefficients multiplied by corresponding image intensities)

T → threshold (an intensity value, recall your labs)

–1	–1	–1
–1	8	–1
–1	–1	–1

Point Detection (1):

- ***Graphical Example***



Porosity with single black pixel embedded within it

Detected point

Original image → X-ray image of turbine blade with a porosity

Result of running mask over entire image

Result after thresholding → threshold set to 90% of highest pixel value in image

Line Detection (1):

- **More Difficult Than Point Detection**
  - Lines can be oriented in any manner (e.g., horizontally, vertically, +/-45^0, etc.)
    - Different mask to detect each line orientation

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

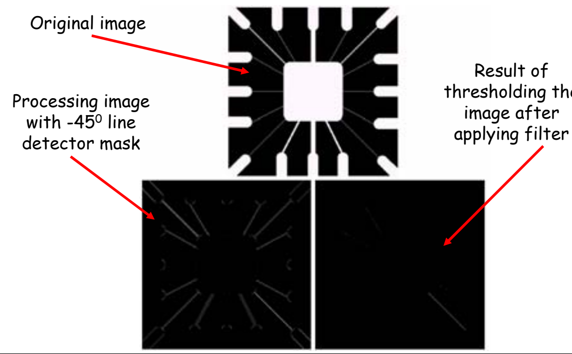
−45°

Line Detection (2):

- **Notes Regarding the Line Detection Masks**
  - Typically these masks detect lines 1 pixel thick
  - Preferred direction of each mask is weighted with a larger coefficient than the other possible directions (e.g., 2 instead of -1)
  - Coefficients sum to zero
    - Response will be equal to zero in areas of constant gray level

Line Detection (3):

- **Line Detection Graphical Example**

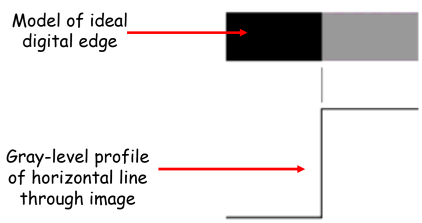


Edge Detection (1):

- **Basic Formulation**
  - What is an edge (review) → set of connected pixels that lie on a boundary between two regions
    - Different from a boundary → boundary is more of a "global" concept whereas edge is a more of a "local" concept
  - Modeling of an ideal edge
    - A set of connected pixels, each of which is located at an orthogonal step transition in gray level

Edge Detection (2):

- **Basic Formulation (cont…)**
  - Modeling of ideal edge – graphical illustration

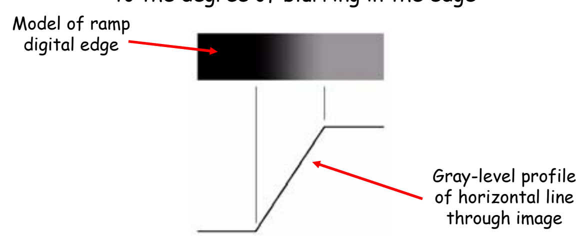


Edge Detection (3):

- **In Practice, Ideal Edges Do Not Exist!**
  - Sampling and the fact that sampling acquisition equipment etc. is far from perfect leads to edges that are blurred
  - Changing illumination (lighting conditions) will cause changes to edges & all parts of an image in general
    - Changes in lighting is actually a HUGE problem for vision/image processing tasks → many algorithms will not generalize across different lighting conditions
    - Color constancy → a big field in computer vision but still an un-solved problem!

Edge Detection (4):

- **In Practice, Ideal Edges Dont Exist! (cont..)**
  - In reality, edges have a more "ramp-like" profile
    - The slope of the ramp is inversely proportional to the degree of blurring in the edge

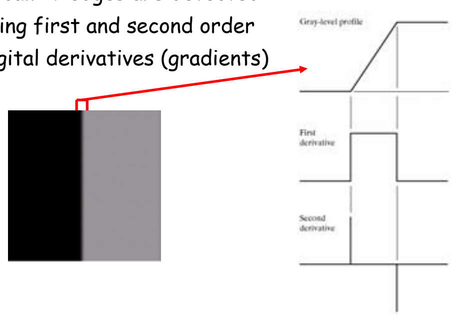


Edge Detection (5):

- **In Practice, Ideal Edges Dont Exist! (cont..)**
  - Edge is no longer a one-pixel thick path
    - An edge point is now any point contained in the ramp and an edge would be a set of such points which are connected
    - Thickness of edge is given length of ramp which is determined determined by the slope which itself is determined by the amount of blurring
    - Blurred edges are typically thicker e.g., the greater the blurring → the thicker the edge

Edge Detection (6):

- **Detecting Edges**
  - Recall → edges are detected using first and second order digital derivatives (gradients)



Edge Detection (7):

- **Detecting Edges (cont…)**
  - Remember
    - First derivative → positive at points of transition into and out of ramp (moving from left to right) & zero in constant gray-level areas
    - Second derivative → positive at transition associated with the "dark" side of edge, negative at light side of edge and zero along ramp & in areas of constant gray level

Edge Detection (8):

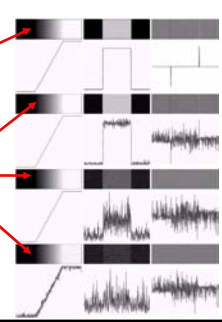
- **Detecting Edges (cont…)**
  - Some conclusions regarding derivatives & edges
    - Magnitude of first order derivative can be used to detect presence of edge at point
    - Sign of second order derivative can be used to determine whether edge pixel itself lies on dark or bright side of edge
    - Second order derivative produces two values for every edge & therefore zero-crossing
    - Zero-crossing → imaginary straight line drawn from positive to negative value would cross zero near midpoint of the edge

Edge Detection (9):

- **Edge Detection Example**
  - Entire transition from left to right is single edge

No noise in image

Gaussian noise added to image above



1st column → first derivative

3rd column → second derivative

Edge Detection (10):

- **Edge Detection Example**
  - Conclusions we can draw from previous examples
    - To be classified as edge point, gray-level transition must be significantly stronger than background
    - Threshold used to determine whether it is different from background → e.g., will be classified as edge only if derivative is greater than some but thresholds have their own problems!
    - The set of all these points greater than the threshold and connected comprise the edge

Introduction (1):

- ***Central to Image Processing/Computer Vision***
  - Essentially, thresholding basically involves performing a check at each pixel location
  - This should be familiar from your labs!

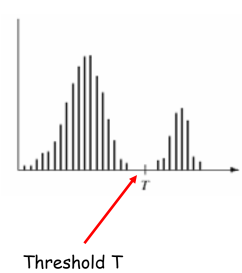
For each pixel (x,y) in image

1. Obtain pixel intensity p_i
2. Compare p_i with pre-defined threshold value T
   - if $p_i \geq T$ then $p_i = 1$ (p_i is an object point)
   - if $p_i < T$ then $p_i = 0$ (p_i is background point)

Thresholding

Introduction (2):

- ***Graphical Example***
  - Histogram of image with light object and dark background
  - After performing thresholding of image with threshold T, pixels corresponding to object will be highlighted (e.g., set to 1) while background pixels will be set to zero



Introduction (3):

- **Multi-Level Thresholding**
  - Can be used to locate (detect) multiple objects where each object is within some range of intensities
    - Multiple thresholds and therefore multiple checks per pixel
    - For example, two objects, two threshold T_1, T_2
    - Pixel belongs to one object if $T_1 < f(x,y) \le T_2$
    - Pixel belongs to other object if $f(x,y) > T_2$
    - Pixel belongs to background if $f(x,y) \le T_1$

Introduction (4):

- ***Graphical Example***
  - Multi-level thresholding