ELIC 629
Digital Image Processing

Fall 2005

Image Enhancement in the Spatial Domain & an Introduction to the Fourier Transform

Bill Kapralos

Monday, November 14 2005

Overview (1):

- **Before We Begin**
  - Administrative details
  - Review → some questions to consider
- **Introduction to Sharpening Filters (cont.)**
  - Second order derivative
  - Laplacian
- ***Combining Spatial Enhancement Features***
  - Introduction
  - Examples

Overview (2):

- **The Fourier Transform**
  - Introduction
  - Background
- **The One-Dimensional Fourier Transform**
  - Introduction
  - Properties

Before We Begin

Administrative Details (1):

- **Lab Six Today**
  - No assignment
  - Lab report required
  - This lab may take two weeks
    - We will see how it goes
  - Requires the use of Matlab
    - No camera required
  - Ideally, you will read and look over the lab before coming to the lab!

Administrative Details (2):

- **Mid-Term Exam**
  - Exams will be given back during the lab period
  - We will go over the exam solutions at a latter time definitely before the final exam!

Some Questions to Consider (1):

- What is an edge ?
- How do edges arise ?(three ways)
- What is a digital derivative ?
- What is the gradient operator ?
- What is a first-order derivative ?
- What is a second-order derivative ?
- What is a Sobel operator ?
- How do we apply the Sobel operator ?

Second Order Derivatives The Laplacian Operator

Introduction (1):

- **2D, Second Order Derivative Operator**
  - Basic approach
    - Define some discrete formulation for the second derivative
    - Using this formulation, define a filter mask (template etc.)
  - Isotropic filters → rotation invariant filters
    - Filter response independent of the direction of discontinuity
    - Rotating image and applying filter yields same results!

Introduction (2):

- **Simplest Isotropic Derivative Operator is the Laplacian, Defined for Image f(x,y) as**

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

  - Laplacian is a linear operator
    - Derivatives of any order are linear operators
  - Above expression is of course formulated in a continuous form
    - Must "convert" to discrete form if it is to be of any use for image processing

Defining the Discrete Laplacian (1):

- **Several Ways to Define a Discrete Laplacian Using Neighborhoods**
  - Must however satisfy the second order derivative properties previously described
  - Recall second order derivative previously given

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

  - We will basically "expand" on this formulation to account for both spatial variables x,y

Defining the Discrete Laplacian (2):

- **Defining a Discrete Laplacian (cont…)**
  - Partial second order (discrete) derivative in the "x" direction defined as

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

  - Partial second order (discrete) derivative in the "y" direction defined as

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

Defining the Discrete Laplacian (3):

- **Defining a Discrete Laplacian (cont…)**
  - By summing the x,y components, we obtain the digital implementation of the 2D Laplacian

$$\nabla^2 f = [f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]-4f(x,y)$$

  - Can be implemented using the following mask (kernel)
  - Isotropic results for rotations in multiples of 90° only!
    - In other words, diagonal directions ignored!

0	1	0
1	−4	1
0	1	0

Defining the Discrete Laplacian (4):

- **Defining a Discrete Laplacian (cont…)**
  - Diagonal directions can however be incorporated by adding two more terms, one for each of the two diagonal directions
  - Can be implemented using the following mask (kernel)
  - Isotropic results for rotations in multiples of 45° only!

1	1	1
1	−8	1
1	1	1

Defining the Discrete Laplacian (5):

- **Defining a Discrete Laplacian (cont…)**
  - A "negative version" of the Laplacian definition is also available in which the coefficients of the mask are negative of the ones given
    - Yields the same results

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Defining the Discrete Laplacian (6):

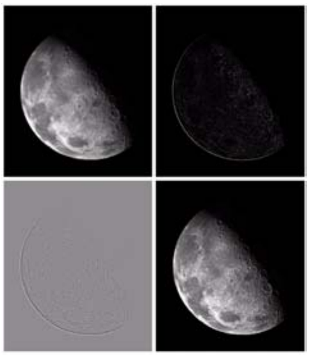
- **Laplacian in Practice**
  - Since it is a derivative operator, it highlights gray level discontinuities and deemphasizes regions with slow varying gray levels
    - Produces images that have grayish edge lines and other discontinuities on featureless background
  - Typically, add (or subtract if negative version of mask used) the Laplacian output image to the original input image
    - Recover the background
    - Preserve sharpening effect of the Laplacian

Defining the Discrete Laplacian (7):

- **Graphical Illustration of the Laplacian**

Original image – north pole of moon

Applying the Laplacian filter – contains both positive and negative values!



Scaled by taking absolute value of previous image to eliminate negative values – not really "correct"!

Laplacian and original image added together

Defining the Discrete Laplacian (8):

- **Laplacian in Practice – Simplifications**
  - Can incorporate the two steps of performing the Laplacian and adding results to the original image using a single mask

0	−1	0
−1	5	−1
0	−1	0

Diagonal directions ignored

−1	−1	−1
−1	9	−1
−1	−1	−1

Diagonal directions emphasized

Combining Spatial Enhancement Features

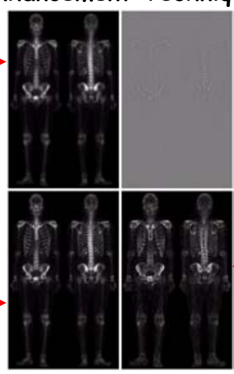
Introduction (1):

- **Frequently, Many Enhancement Approaches are Combined to Produce Desired Result**
  - Best way to illustrate this is by an example → consider a bone scan used to detect diseases such as bone infections and tumors
    - Goal → enhance original image by sharpening it to bring out more skeletal detail
    - But original image gray level dynamic range is low and contains high noise

Introduction (2):

- **Multiple Enhancement Techniques Example**

A. Original image



B. Laplacian applied to original image – notice the skeletal outline

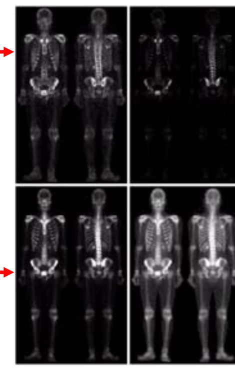
C. Sharpened version obtained by summing original with Laplacian

D. Sobel applied to original image

Introduction (3):

- **Multiple Enhancement Techniques Example**

E. Sobel image processed with a 5x5 averaging filter



F. Mask image formed by product of image C and E

G. Sharpened version obtained by summing A and F

H. Power law transformation applied to image G to increase contrast

Introduction (4):

- **Multiple Enhancement Techniques**
  - Keep in mind that performing these multiple operations can become computationally very expensive!
    - Typically not done in real-time!
    - In many cases, real-time not required → medical imaging etc. do not necessarily need to be real-time. Can be processed afterwards and results can typically be made available after a day or more

The Fourier Transform

Background (1):

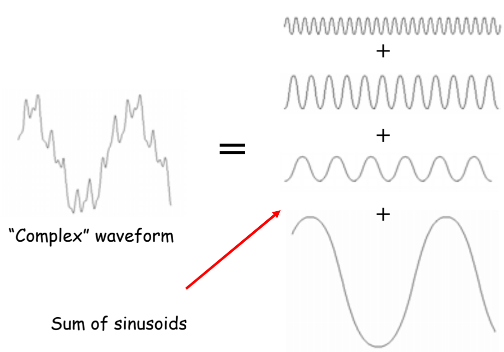
- **Fourier Domain Processing is Fundamental to Image Processing**
  - To fully understand image processing at the very least, a basic understanding of Fourier processing is needed!
    - Perform a Fourier transform on (spatial domain) image to obtain its spectral components
    - Perform some operation on this spectral representation
    - Perform inverse Fourier operation to get back the spatial representation

Background (2):

- **Introduced by the French mathematician Jean Baptiste Fourier in 1807**
  - Published his theory in a book titled "The Theory of Heat" (1822)
  - Fourier's theory (Fourier series) → any function that periodically repeats itself (infinitely) can be expressed as a sum of sines and/or cosines of different frequencies, each multiplied by different coefficient
    - Doesn't matter how complicated the function is, as long as it repeats itself!

Background (3):

- **Graphical Illustration**



Background (4):

- **Can Even Represent Non-Periodic, Finite Functions as the Integral of Sines and/or Cosine Functions**
  - Provided area under resulting curve of the function is finite
  - This formulation is known as a Fourier transform as opposed to a Fourier series
  - Even more useful when considering practical problems → many times functions (signals) in "real-life" are not periodic and are finite

Background (5):

- **Important Characteristics of Both Fourier Transform and Fourier Series**
  - Can completely recover (reconstruct) the original (spatial representation) function with NO loss of information
    - Can work in the Fourier Domain and then return back to spatial domain → many problems are easier solved in the Fourier domain

Background (6):

- **The Functions (Images) we are Dealing with Are Finite in Duration**
  - We are therefore primarily interested and will be dealing with, is the Fourier transform
- **Many Image Enhancement Techniques in the Fourier Domain**
  - Extremely useful
  - Can be easier to understand what exactly is happening and how the operations work

The One-Dimensional Fourier Transform

Introduction (1):

- **Originally, Fourier Transform was Formulated with Continuous Time Signals**
  - We are dealing with sampled images
  - Finite intensity values and finite in duration
  - In other words, we are dealing with a discrete signal → remember, an image itself is a signal as in your DSP course, except we are now dealing with a two-dimensional signal as opposed to a one-dimensional signal you are familiar with
  - Discrete Fourier Transform (DFT) introduced to handle discrete signals

Discrete Fourier Transform (1):

- **One of the Most Common and Powerful Procedures Encountered in the Field of Digital Signal Processing in General**
  - Enables us to analyze, manipulate and synthesize signals in ways not possible with continuous (analog) signal processing
  - Used in every field of engineering
  - A solid understanding of the DFT is extremely important!

Discrete Fourier Transform (2):

- **What is the Discrete Fourier Transform ?**
  - A mathematical procedure used to determine the frequency (or harmonic) content of a discrete signal
    - Remember → discrete signal obtained by periodically sampling a continuous time signal in the time domain
  - Based on the Continuous Fourier Transform (CFT), denoted by X(f) (or F(u))

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

Discrete Fourier Transform (3):

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$

- **Lets Analyze This Expression:**
  - f → frequency (spectral component)
  - x(t) → continuous time domain signal
  - $e^{-j2\pi ft}$ → a sinusoid (sine wave) of frequency f
  - In words → Fourier Transform of frequency component f is a correlation of the infinite input signal at each time step with a sine wave of frequency f → X(f) tells us "how much" of the sine wave of frequency f the signal contains

Discrete Fourier Transform (4):

- **Discrete Fourier Transform (DFT) Mathematically**

$$X[m] = \frac{1}{M}\sum_{n=0}^{M-1} x[n]e^{-j2\pi nm/N}$$

  - Using Euler's Relationship $e^{-j\theta} = \cos(\theta) - j\sin(\theta)$ we obtain:

$$X[m] = \frac{1}{M}\sum_{n=0}^{M-1} x[n](\cos(2\pi nm/N) - j\sin(2\pi nm/N))$$

Discrete Fourier Transform (5):

$$X[m] = \sum_{n=0}^{M-1} x[n](\cos(2\pi nm/N) - j\sin(2\pi nm/N))$$

- X[m] → mth DFT output e.g., X[0], X[1] … X[M-1]
- m → index of the DFT output in frequency domain (m = 0, 1, 2, … M-1)
- x[n] → sequence of input (discrete) samples (x[0], x[1], x[2] … X[n-1])
- n → (discrete) time domain index of input samples
- j = sqrt(-1) (remember, complex numbers!)
- N → number of samples (same for input and DFT)

Discrete Fourier Transform (6):

- **Some Notes Regarding the DFT**
  - Indices for input samples and DFT output samples always go from 0 to N-1
    - With N input time domain samples, the DFT determines the spectral content of the input at N equally spaced frequency points
    - N is an important parameter and determines
      1. How many input samples are needed
      2. Resolution of the frequency domain results
      3. Amount of processing time required to calculate an N-point DFT

Discrete Fourier Transform (7):

- **Some Notes Regarding the DFT (cont…)**
  - In words:
    - Each X[m] DFT output is the sum of a *point for point* product between an input sequence of input values and a complex sinusoid of the form cos(θ) – jsin(θ)
    - Exact frequencies of the of the different sinusoids depend on sample rate f_s and number of samples N
    - Fundamental frequency of the sinusoids is f_s /N and all other X[m] analysis frequencies are integer multiples of the fundamental!

Discrete Fourier Transform (8):

- **Some Notes Regarding the DFT (cont…)**
  - The N separate DFT analysis frequencies are

$$f_{analysis}(m) = \frac{mf_s}{N}$$

  - So, X[0] gives us magnitude of an 0Hz ("DC") component contained in the signal, X[1] gives us magnitude of the fundamental component, X[2] gives us magnitude of 2 x fundamental component contained in signal etc.
  - Finally, keep in mind, we are dealing with complex sinusoids → magnitude and phase!

Discrete Fourier Transform (9):

- **Determining the Magnitude and Phase Contained in each X[m] Term**
  - We can represent an arbitrary DFT output value X[m] by its real and imaginary parts

$$X[m]=X_{real}[m]+jX_{imag}[m]=X_{mag}[m] \text{ at angle of } X_{\theta}[m]$$

  - The magnitude of X[m] is

$$X_{mag}[m]=|X[m]|=\sqrt{X_{real}[m]^2+X_{imag}[m]^2}$$

Discrete Fourier Transform (10):

- **Determining the Magnitude and Phase Contained in each X[m] Term (cont…)**
  - The phase angle of X[m], $X_{\theta}[m]$ is

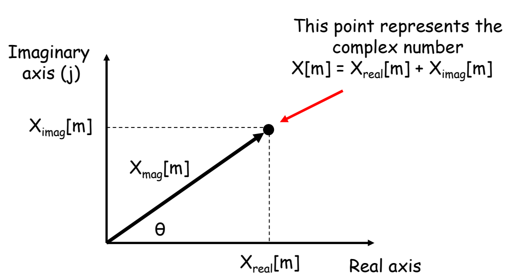
$$X_{\Theta}[m]=\tan^{-1}\left(\frac{X_{imag}[m]}{X_{real}[m]}\right)$$

  - The power of X[m], known as the power spectrum or spectral power is the magnitude squared

$$X_{PS}[m]=X_{mag}[m]^2=X_{real}[m]^2+X_{imag}[m]^2$$

Discrete Fourier Transform (11):

- ***Graphical Illustration of Phase and Magnitude (Complex Plane)***



Some Properties of the 1D DFT

DFT Symmetry (1):

- **Symmetry in DFT Output is Obvious!**
  - Standard DFT is designed to accept complex input but most physical DFT inputs are "real" inputs
    - Non-zero real sample values
    - Imaginary values are assumed to be zero
  - With "real" input x[n] the complex DFT outputs for n = 1 to n = (N/2) – 1 are redundant with frequency output values for m > (N/2)
    - mth DFT output will have the same value as the (N-m)th DFT output
    - the phase angle of the mth output is the negative of the (N-m)th DFT output

DFT Symmetry (2):

- **Symmetry in DFT Output is Obvious! (cont…)**
  - What does this symmetry mean?
    - If we perform an N-point DFT on a real input sequence, we get N separate complex DFT output terms but only the first N/2 terms are independent
    - To obtain DFT of x[n], we need only compute the first N/2 values of X[m] where $0 \leq m \leq (N/2)-1$
    - The X[N/2] to X[N-1] DFT output terms provide no additional information about the spectrum of the real sequence x[n]

DFT Linearity (3):

- **DFT is Linear**
  - The DFT of the sum of two signals is equal to the sum of the transforms of each signal
    - Let $x_1[n]$ and $x_2[n]$ be two discrete input signals with DFT $X_1[m]$ and $X_2[n]$ respectively
    - Consider the sum of these two signals

$$x_{sum}[n] = x_1[n] + x_2[n]$$

    - The DFT of $x_{sum}[n]$ is

$$X_{sum}[m] = X_1[m] + X_2[n]$$

DFT Linearity (4):

- **DFT is Linear (cont…)**
  - Exercise:
    - Mathematically prove this linearity property for the DFT

Inverse DFT

Inverse DFT – IDFT (1):

- **Reverse the DFT Process**
  - DFT transforms time-domain data into frequency domain representation
  - With inverse DFT, we transform frequency domain representation into time-domain representation
    - Perform IDFT on X[m] frequency domain values

$$x[n] \quad = \quad \sum_{n=0}^{M-1} X[m](\cos(2\pi nm/N) - j\sin(2\pi nm/N))$$