ELIC 629

Digital Image Processing

Winter 2006

2D Discrete Fourier Transform and Frequency Domain Filtering

Bill Kapralos

Tuesday, April 11 2006

Overview (1):

- **Before We Begin**
  - Administrative details
  - Review → some questions to consider
- **The Two-Dimensional Fourier Transform**
  - Review
- **Filtering in the Frequency Domain**
  - Properties of the frequency domain
  - Gaussian filters
  - Smoothing filters
  - Sharpening filters

Before We Begin

Administrative Details (1):

- **Lab Seven Today**
  - Should be a straightforward lab to complete
  - Lab report required
    - Last lab report!
  - No camera required

Some Questions to Consider (1):

- What is a Fourier series ?
- What is a Fourier transform ?
- What is the discrete Fourier transform ?
- What determines the resolution of the DFT output ?
- What is the relationship between the size of the DFT and the size of the input ?
- Describe the symmetry property of the DFT

Introduction to the Two-Dimensional Fourier Transform

Introduction (1):

- **Straightforward to Extend One-Dimensional DFT to Two Dimensions**
  - Two-dimensional DFT of a function (image) f(x,y) of size M x N is given by

$$F[u,v] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f[x,y]e^{-j2\pi(ux/M+vy/N)}$$

  - Using Euler's relationship, we have the following

$$F[u,v] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f[x,y](\cos(-j2\pi(ux/M+vy/N)) + j\sin(-j2\pi(ux/M+vy/N)))$$

Introduction (2):

- **Straightforward to Extend One-Dimensional DFT to Two Dimensions (cont…)**
  - We can also easily extend the IDFT to two-dimensions as well. Given F[u,v], IDFT is

$$f[x,y] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F[u,v]e^{-j2\pi(ux/M+vy/N)}$$

  - Using Euler's relationship, we have the following

$$f[x,y] = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F[u,v](\cos(-j2\pi(ux/M+vy/N)) + j\sin(-j2\pi(ux/M+vy/N)))$$

Introduction (3):

- **Some Notes About 2D DFT**
  - x = 0, 1, 2, …, M-1 and y =0, 1, 2, …, N-1
  - Variables u and v are the *transform* or *frequency* variables and x, y are the *spatial* or *image* variables
  - As with 1D DFT, we can define the magnitude, phase and power spectrum in a similar manner
  - Magnitude

$$|F[u,v]| = \sqrt{R[u,v]^2 + I[u,v]^2}$$

Introduction (4):

- **Some Notes About 2D DFT (cont...)**
  - Phase φ[u,v]

$$|\phi[u,v]| = \tan^{-1}\left[\frac{I[u,v]}{R[u,v]}\right]$$

  - Power spectrum P[u,v]

$$P[u,v] = |F[u,v]|^2 = R[u,v]^2 + I[u,v]^2$$

  - where R[u,v] and I[u,v] are the real and imaginary components of the DFT F[u,v] respectively

Introduction (5):

- **Some Notes About 2D DFT (cont...)**
  - Typically we multiply input image by $(-1)^{x+y}$ (pixel-by-pixel multiplication) prior to computing the DFT
    - Shifts the origin of the DFT to frequency coordinates (M/2, N/2) → the center of the M x N 2D DFT
    - M, N → even integers
  - After the multiplication, the DFT becomes

$$F[u,v] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(f[x,y]e^{-j2\pi(ux/M+vy/N)})(-1)^{x+y}$$

Introduction (6):

- **Some Notes About 2D DFT (cont...)**
  - Which is equal to

$$F[u,v] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(f[x,y]e^{-j2\pi(ux/M+vy/N)})(-1)^{x+y} = F(u-M/2, v-N/2)$$

  - When we implement 2D DFT summations run from u = 0 to M-1 and v = 0 to N-1.
  - The center of the transform is at u = (M/2) + 1 and v = (N/2) +1

Introduction (7):

- **DC Component**
  - DFT at the origin (0,0) in the frequency domain is equal to the average gray level (intensity) of image f(x,y)

$$F[0,0] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

Some 2D DFT Relationships (2):

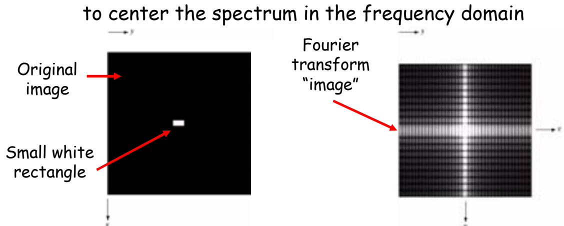
- **Frequency and Spatial Domains**

  $\Delta u = 1/(M\,\Delta x)$ and $\Delta u = 1/(N\,\Delta y)$

  - In other words → inverse relationship between spatial and frequency domain resolution

2D DFT Example (1):

- **2D DFT of a "Simple" Image**
  - 20 x 40 rectangle superimposed on black background of size 512 x 512
  - Image multiplied by $(-1)^{x+y}$ prior to computing DFT to center the spectrum in the frequency domain



2D DFT Example (2):

- **Some Comments regarding the Example**
  - Inverse spatial vs. frequency relationship
    - Separation of "spectrum zeros" in u direction is twice separation in v direction → 1 to 2 size ratio of rectangle in the image
  - Spectrum was processed using log transform prior to displaying to enhance gray level
    - Recall, dynamic range of DFT is huge and if we didn't process it, little detail would be evident
    - Most DFT spectra are processed with the log transform prior to displaying

Filtering in the Frequency Domain

Properties Frequency Domain (1):

- **Usually No Direct Association Between Specific Components of Image and its DFT**
  - However some general statements can be made between frequency components of DFT and spatial characteristics
    - Frequency is directly related to rate of change so we can associate the frequency domain with patterns of intensity variation in image

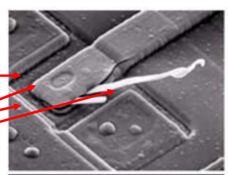
Properties Frequency Domain (2):

- **General Statements (cont…)**
  - DFT at origin (0,0) gives average intensity of image (e.g., DC component)
    - Moving away from origin → low frequencies correspond to slowly changing image components (e.g., in an image of a room, these may correspond to a smooth wall or floor)
    - As we move further away from the origin, higher frequencies → correspond to the greater gray level (intensity) changes (e.g., edges in an image corresponding to large variations in the image)

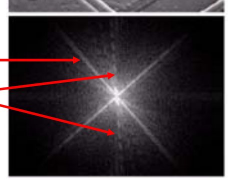
Properties Frequency Domain (3):

- **General Statements (cont…)**
  - Graphical illustration

Scanning electron microscope image of damaged IC magnified 2500 times.
Interesting features
→ strong edges at approx. +/- 45°
→ white oxide protrusions



DFT of image above
→ strong spectral features +/- 45°
→ vertical component just off to the left due to edges of oxide protrusion – notice the "zeros" in the spectrum – they correspond to the narrow vertical span of the protrusion



Properties Frequency Domain (4):

- **Basics of Filtering in Frequency Domain**
  - Consists of the following steps
    1. Multiply image (in spatial domain) by $(-1)^{x+y}$ to center the transform about the origin
    2. Compute DFT F[u,v] of image in step 1
    3. Multiply F[u,v] by desired filter function H[u,v]
    4. Compute inverse DFT of result in step 3
    5. Extract real part of the result in step 4
    6. Multiply result of step 5 by $(-1)^{x+y}$ to "shift back" the image

Properties Frequency Domain (5):

- **Basics of Filtering in Frequency Domain**
  - Details regarding the filter H[u,v]
    - Also referred to as a *filter transfer function*
    - Called a filter because it suppresses certain frequencies while leaving other un-touched (e.g., low pass and high pass filters from DSP course) → remember, no such thing as an "ideal" filter in reality!
    - In general, mathematically, the filtered DFT output is given by

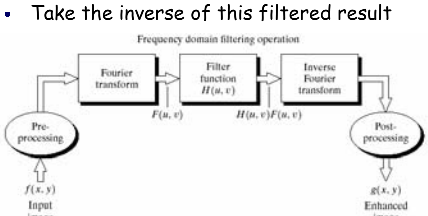
$$G[u,v] = H[u,v]F[u,v]$$

Properties Frequency Domain (6):

- **G[u,v] =H[u,v]F[u,v] in Detail**
  - Involves two-dimensional multiplication
    - Element-by-element basis
  - Zero-Phase filters
    - Elements of F[u,v] are typically complex values however, for our purpose, H[u,v] will typically be real
    - Multiply both real and imaginary parts of the corresponding components of F by the value of H[u,v] → since phase is not altered, it is called a zero-phase filter

Properties Frequency Domain (7):

- **Graphical Summary of DFT Filtering**
  - These steps may vary but basic idea is the same
    - Modify the transform of the image in some manner with some filtering function
    - Take the inverse of this filtered result



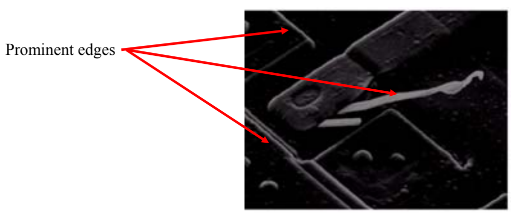
Basic Filters & their Properties (1):

- **Notch Filter**
  - "Zero DC filter" → suppose we want to set average intensity value to zero
    - Set this term to zero (e.g., F[0,0] = 0) in the frequency domain
    - Take inverse DFT of resulting transform → now average intensity value of image is zero!
    - This simple filter can be accomplished by

$$H[u,v] = \begin{cases} 0 & if \quad (u,v) = (M/2, N/2) \\ 1 & otherwise \end{cases}$$

Basic Filters & their Properties (2):

- **Notch Filter (cont…)**
  - Graphical example of notch filter → notice the overall decrease of gray level and notice that prominent edges now stand out



Basic Filters & their Properties (3):

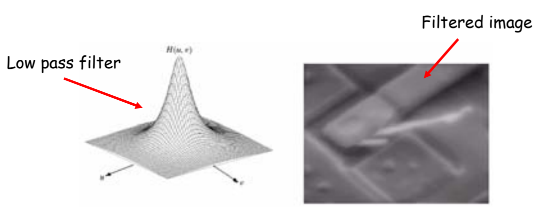
- **Notch Filter (cont…)**
  - Remember
    - In reality, average of image cannot be equal to zero because image needs to have negative values for an average gray level to be zero and displays can't handle negative values
    - We basically have to modify the image to display it → e.g., one way is to assign negative values a new value of zero (black) and all other values up from that (as done in previous example)

Basic Filters & their Properties (4):

- **Low-Pass Filter**
  - Low frequencies result from general gray level appearance of image (e.g., smooth areas)
  - A low pass filter will ideally eliminate high frequencies while completely leaving low frequencies un-touched
    - In reality of course, high frequencies are not entirely eliminated but rather attenuated
    - Low pass filtered image will have less sharp details than original since high frequencies which are responsible for sharp transitions are attenuated

Basic Filters & their Properties (5):

- **Low-Pass Filter (cont…)**
  - Graphical illustration (example) of low pass filter and resulting image after it has been filtered
    - Notice the blurred results since high edges etc. are removed



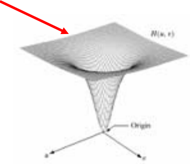
Basic Filters & their Properties (6):

- **High-Pass Filter**
  - High frequencies are responsible for details in image such as edges and noise
  - A high-pass filter will ideally eliminate low frequencies while completely leaving high frequencies un-touched
    - In reality of course, low frequencies are not entirely eliminated but rather attenuated
    - High pass filtered image will have less gray level variation in smooth areas while transitional gray level detail will be emphasized making image appear sharper

Basic Filters & their Properties (7):

- **High-Pass Filter (cont…)**
  - Graphical illustration (example) of high-pass filter and resulting image after it has been filtered
    - Sharp, with little gray level detail
    - Usually, constant is added to filter so it doesn't remove F[0,0] completely

Low pass filter



Filtered image

Frequency vs. Spatial Domain (1):

- **Convolution Theorem**
  - Establishes most fundamental relationship between frequency and spatial domains
    - Remember filtering in the spatial domain ?
    - Formally, convolution of two functions denoted by f(x,y) * h(x,y) is defined by

$$f(x,y)*h(x,y)=\frac{1}{MN}\sum_{0}^{M-1}\sum_{0}^{N-1}f(m,n)h(x-m,y-n)$$

  - Minus sign in h(x-m, y-n) means that the function h is mirrored about the origin
    - Inherent in the definition of convolution

Frequency vs. Spatial Domain (2):

- **Convolution Theorem (cont…)**

$$f(x,y)*h(x,y)=\frac{1}{MN}\sum_{0}^{M-1}\sum_{0}^{N-1}f(m,n)h(x-m,y-n)$$

  - Basically, above equation states the following
    1. Flipping one function about the origin
    2. Shifting that function with respect to the other by changing the values of (x, y)
    3. Computing a sum of products over all values of m and n for each displacement (x, y) → displacements (x, y) are integer increments that stop when the functions no longer overlap

Frequency vs. Spatial Domain (3):

- **Convolution Theorem (cont…)**
  - Consider the following definitions
    - F[u, v] → Fourier transform of f[x,y]
    - H[u, v] → Fourier transform of h[x,y]
  - One half of the convolution theorem states that f(x,y)*h(x,y) and F[u,v]H[u,v] comprise a Fourier transform pair. Mathematically,

$$f(x,y)*h(x,y) \Leftrightarrow F[u,v]H[u,v]$$

  - In words, convolution in the spatial domain is equal to multiplication in the frequency domain

Frequency vs. Spatial Domain (4):

- **Convolution Theorem (cont…)**
  - Other half of the convolution theorem states that f(x,y)h(x,y) and F[u,v]*H[u,v] comprise a Fourier transform pair. Mathematically,

$$f(x,y)h(x,y) \Leftrightarrow F[u,v]*H[u,v]$$

  - In words, multiplication in the spatial domain is equal to convolution in the frequency domain

Frequency vs. Spatial Domain (5):

- **Filters in the Spatial and Frequency Domain Form a Fourier Transform Pair**
  - Given filter in frequency domain to obtain filter in spatial domain
    - Take inverse DFT of the frequency domain representation of the filter
  - Given filter in spatial domain to obtain filter in frequency domain
    - Take DFT of the spatial domain representation of the filter

Frequency vs. Spatial Domain (6):

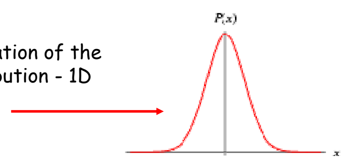
- **Some Notes**
  - All functions previously described are of size M x N (e.g., images and frequency domain representation)
    - Given same size filters in both spatial and frequency domains, typically more computationally efficient to filter in frequency domain
    - But not always worth taking DFT of spatial domain function to get frequency domain rep.

Gaussian Filters (1):

- **What is a Gaussian Function ?**
  - Normal distribution with mean μ and variance σ^2
  - Defined by the following distribution function

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

Graphical illustration of the Gaussian distribution - 1D



Gaussian Filters (2):

- **Gaussian Function as a Filter**
  - Very Useful and important
    - Their shape is easily specified
    - Both DFT and IDFT of a Gaussian is also a Gaussian
    - An averaging ("blurring") filter

Gaussian Filters (3):

- **Mathematically → Fourier Transform Pair**
  - Let H[u] denote frequency domain Gaussian filter given by

$$H[u] = Ae^{-u^2/2\sigma^2}$$

  - Corresponding filter in the spatial domain is given by

$$h[x] = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$

  - Both functions above comprise a Fourier transform pair → both Gaussian and real valued (e.g., no complex numbers!

Gaussian Filters (4):

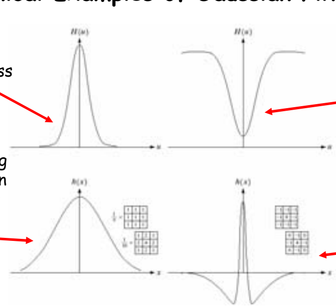
- **Fourier Transform Pair (cont…)**
  - Both functions behave reciprocally to each other
    - When H[u] has a broad profile (and therefore large σ) → h[x] will have a narrow profile
    - When h[x] has a broad profile (and therefore large σ) → H[u] will have a narrow profile

Gaussian Filters (5):

- **Graphical Examples of Gaussian Filters**

Gaussian frequency domain low pass filter

Gaussian frequency domain high pass filter



Corresponding spatial domain Gaussian low pass filter

Corresponding spatial domain Gaussian high pass filter

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):

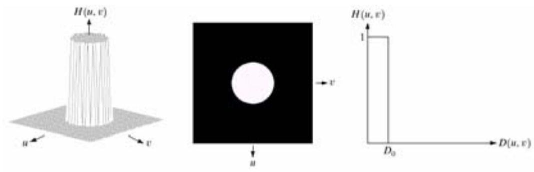
- **Mathematically**

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

  - 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

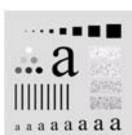
low pass filter displayed as an image

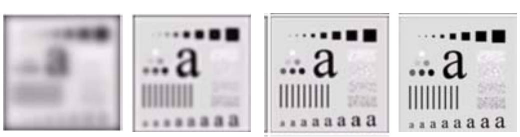
Filter radial cross-section where D_0 is radius of "circle" e.g., determines cur-off frequency

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):

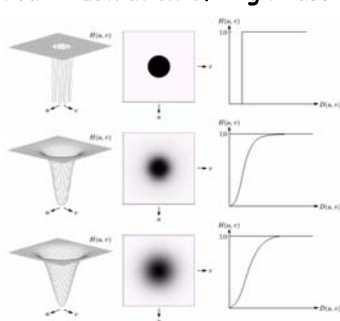
- **What is a Sharpening Filter (Review)**
  - Removes (ideally) low frequency components of an image's Fourier representation (e.g., keeps frequency components above some cut-off frequency)
  - Basically, the reverse of the low pass filter and given mathematically by

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

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

Introduction (1):

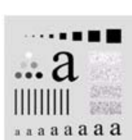
- **Graphical Illustration of High Pass Filtering**

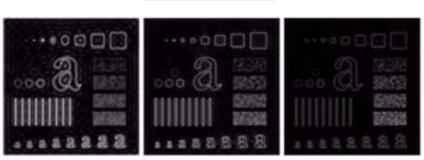


Introduction (2):

- **Graphical Illustration of High Pass Filtering**

Original image





$D_0 = 15$ $D_0 = 30$ $D_0 = 80$