Digital Image Processing: Practical Foundation in Spatial Domain
CSC621-821
Biomedical Imaging and Analysis
Dr. Kazunori Okada

Adopted from slides for DT228/4 Digital Image Processing by Dr. Brian MacNamee at Dublin Institute of Technology
http://www.comp.dit.ie/bmacnamee/gaip.htm

Try java applets in
http://homepages.inf.ed.ac.uk/rbf/HIPR2/

Midterm #1 (Next week)

• Open Notes: Handwritten Notes Only
• Also Bring: Function calculator, Pen/pencil/eraser, A few scratch papers.
• CLOSED BOOK & CLOSED XEROX LECTURE NOTE COPIES
• 100 minutes: the rest of the time will be a lecture on DIP
• Five questions, 20 point each, Partial credits (Time is Very Tight)
  – Lec2: one problem (10 min)
  – Lec3: one problem (10 min)
  – Lec4: one problem (30 min)
  – Lec5: two problems (50 min)
• Lec 2 & 3: understand the big pictures of various imaging modalities. Focus on Difference/Pros/Cons
• Lec 4 & 5: study and understand various image processing processes and analyses you learned so that you can derive basic formulae on your own and solve a small scale problem by hand with actual data with your calculator.
• 621: Only top 4 scores counted, 821: All scores counted
Overview

• Last lecture on how to manage data
  – Data Structure: Annotated Lattice, matrix, function
  – Image File Formats: DICOM/Analyze
  – Image Visualization: MRP/MIP/Volume rendering

• Today’s lecture
  – Practical Foundation of Digital Image Processing I
    – Spatial Domain Analysis
    – Image Enhancement
    – Point Processing: Intensity Transformations
    – Neighbor Processing: Spatial Smoothing Filtering

Digital Image Processing: Definition

• DIP: Manipulation of Digital Images for
  – Improving pictorial information for human interpretation
  – Processing image data for storage, transmission and representation for autonomous machine perception

Characteristics:
  • Mountain
  • Snow
  • Cheery tree
  • Blue sky
Digital Image Processing: Overview

• What manipulation we talk about???
  – Smoothing, Sharpening, Denoising, Restoration, Compression, Edge detection, Shape Morphology, Segmentation, Registration, Feature extraction, Recognition, Classification

<table>
<thead>
<tr>
<th>Low Level Process</th>
<th>Mid Level Process</th>
<th>High Level Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: Image</td>
<td>Input: Image</td>
<td>Input: Image/Attributes</td>
</tr>
<tr>
<td>Output: Image</td>
<td>Output: Attributes</td>
<td>Output: Understanding</td>
</tr>
<tr>
<td>Examples: Noise removal, image sharpening</td>
<td>Examples: Object recognition, segmentation</td>
<td>Examples: Scene understanding, autonomous navigation</td>
</tr>
</tbody>
</table>

Image Processing Computer Vision AI

Image Filtering: Basic Framework

• A general-purpose image manipulation
• Image-In and Image-Out

\[ g(x, y) = T[f(x, y)] \]

\( f(x, y) \): input image
\( g(x, y) \): output image

\( T \): an operator defined over a neighbourhood of \( (x, y) \)

In 3D Image?
**Convolucion**

- Filtering Operation is basically to compute a sum of a product of two functions (an image \( f \) and filter-op \( w \)) along raster scan.

\[
g(x, y) = \sum_{s=-d}^{d} \sum_{t=-d}^{d} w(s, t) f(x + s, y + t)
\]

- This operation can be mathematically formulated by **convolution**.

\[
g(x, y) = w \ast f(x, y) = \int_{\mathbb{R}^2} w(s, t) f(x - s, y - t) ds dt
\]

**Point Processing**

\[
s = T ( r )
\]
Neighbor Processing

\[ s = T ( r, a, b, c, d, e, f, g, h ) \]

**In 3D Image and larger window?**

Signal Frequency: Basic Idea

- Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient
  - a **Fourier series**
Two broad categories of image filtering techniques

• Spatial domain techniques
  – Direct manipulation of image pixels

• Frequency domain techniques
  1. Reorganize an image in the frequency space by applying **Fourier transform**
  2. Then manipulate the image in the Fourier space

• This and next lectures focus on the spatial domain techniques. Frequency domain techniques will be explained later
What is **Image Enhancement**?

One of the major goals of image processing

- Process of making images more useful for...
  - Highlighting interesting detail in images
  - Removing noise from images
  - Making images more visually appealing

- Important for biomedical image analysis
  - Radiologist’s diagnosis by visual inspection
  - Helps to find small/obscure problems
  - Helps to correctly measure the problems

Image Enhancement Example 1
Image Quality

• Basic concepts and types of general digital image quality issues
  – Dynamic Range
  – Quantization Resolution
  – Quantization Error
  – Signal to Noise Ratio
  – Contrast and Noise
Quantization: Review

- Discretizing analog intensity value into a set of finite discrete levels
- Intensity Resolution: # of intensity levels (in bits)

Quantization Error & Dynamic Range

- Quantization Error (QE):
  - Difference between the original analog and quantized digital signals
- Dynamic Range (DR):
  - Ratio between maximum and minimum attainable intensities
  - Given (n+1) levels, \( I_0, \ldots, I_n \), \( DR = I_n / I_0 \)
  - dB: logarithmic unit of logDR:
    \[ 1dB = 20 \log_{10}(DR) \]
- QE limits DR of digital signal
- Maximum achievable DR with Q-bit uniform quantization is

\[
\log DR \leq 20 \log_{10} \left( \frac{2^Q}{1} \right) \approx 6.02QdB
\]
**How Much Resolution is Enough?**

- Human cannot distinguish subsequent intensities $I_{j+1}$ and $I_j$ if the difference is less than 1%
- DR = 100: $n > 463$
- DR = 1000: $n > 694$
- 12bit: $n = 4096$

\[
\begin{align*}
I_{n} & \leq 1.01^n I_0 \\
\implies \frac{I_n}{I_0} & \leq 1.01^n \\
\therefore \log_{1.01}(\frac{I_n}{I_0}) & = n \leq \log_{1.01}(DR)
\end{align*}
\]

**Signal to Noise Ratio**

- Ratio of Signal and Noise

\[
\text{SNR} = \frac{P_{signal}}{P_{noise}} = \left(\frac{\text{averageintensity}}{\text{averagenoise}}\right)^2
\]

- SNR is also limited by dynamic range/quantization error

\[
\text{SNR} \leq (1.76 + 6.02Q)\text{dB}
\]

- For images, SNR can be estimated by ratio of mean $\mu$ and standard deviation $\sigma$ of pixel/voxel values

\[
\text{SNR} = \frac{\mu}{\sigma}
\]

- Rose Criterion: SNR must be at least 5 dB for identifying image details at 100% certainty
Contrast

Difference in maximum and minimum intensity values in an image

- Low contrast
  - Uses only a small part of the available quantized levels
  - Make images appear washed out

Saturation and Noise

- Saturation
  - Analog intensities that were cut off by the maximum quantized intensity range available
  - E.g. highlights

- Noise
  - Signals other than the observed targets caused by some aberrations of imaging process
  - Gaussian noise
  - Salt-n-pepper noise
Noise Example

Salt and Pepper  Gaussian

Point Processing: Overview

- Simplest spatial domain operations
- Neighborhood is simply the pixel itself
- Point processing operations:

\[ s = T \left( r \right) \]

- \( s \): the processed image pixel value
- \( r \): the original image pixel value
- \( T \): intensity transformation function
Intensity Transformation

- Reconfigure the original intensity range into a new output range
- For image enhancement
  - Contrast improvement
  - Identification of target
- Intensity Transformation Function
  - Basic gray level mapping
  - There are many different kinds of grey level transformations
  - Linear, Piecewise Linear, Log, Power Law

Negative Image (Linear)

- Flipping black and white of gray level range
- Useful for enhancing white or grey detail embedded in dark regions of an image
  - Note how much clearer the tissue is in the negative image of the mammogram below
Intensity Transformation Function

Identity

\[ s = r \]

Negative Intensity

\[ s = I_{\text{max}} - r = L - 1 - r \]

Thresholding (Piecewise Linear)

- **Binarization** of Images: single parameter (TH)
- Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

\[ s = \begin{cases} I_{\text{max}} & r > \text{threshold} \\ 0.0 & r \leq \text{threshold} \end{cases} \]
Intensity Transformation Function

\[ s = \begin{cases} 
1.0 & r > k \text{ (threshold)} \\
0.0 & r \leq k \text{ (threshold)} 
\end{cases} \]

\( I_{\text{max}} \) can be 1.0

Contrast Stretching (Piecewise Linear)

- Stretch an input’s narrow contrast range into an entire intensity range available
- User-defined function w/ two parameters \((th_1, th_2)\)
- **Windowing** in CT scan is the same

\[ s = \begin{cases} 
I_{\text{max}} & r > th_2 \\
(a(r-th_1)) & th_1 < r \leq th_2 \\
0.0 & r \leq th_1 
\end{cases} \]
Intensity Transformation Function

\[ s = \begin{cases} 
  L-1 & r > k_2 \\
  a(r-k_1) & k_1 < r \leq k_2 \\
  0 & r \leq k_1 
\end{cases} \]

Window width: \( w = k_2 - k_1 \)

\[ a = \frac{L-1}{w} \]

Slicing (Piecewise Linear)

- Highlights a specific range of grey levels
  - Similar to contrast stretching (3 to 4 parameters)
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image
Intensity Transformation Function

\[ s = \begin{cases} s_t & A < r \leq B \\ s_b & \text{otherwise} \end{cases} \]

\[ s = \begin{cases} s_t & A < r \leq B \\ r & \text{otherwise} \end{cases} \]

Power Law Transformation

Why don’t we use a well defined mathematical function?
- Suppose we normalize intensity range to [0, 1]
- Then consider:
  \[ S = C \cdot r^\gamma \]
  - Map a narrow range of dark input values into a wider range of output values or vice versa
  - Varying \( \gamma \) gives a whole family of curves
Spine MRI Example

- $c$ is set to 1

$$s = r^\gamma$$

Power Law Transform: Example

$\gamma = 0.6$
Power Law Transform: Example

\[ \gamma = 0.4 \]

\[ \gamma = 0.3 \]
**Histogram Concept**

- The histogram of an image shows us the distribution of gray levels in the image.
- Massively useful in image processing, especially in segmentation.

---

**Histogram Computation**

1. Make an array whose length is equal to the number of all available intensity levels.
2. Initialize all the array items by zeros.
3. For each pixel of an image:
   a. Choose an array item using the pixel value as an index.
   b. Increment the item by one.
4. You may divide all items by the total number of pixels for normalization if necessary.
   - 3D Image?

---

**Counts**

- r

---

**Frequencies**

- Gray Levels
Histogram Example with Images

Example 1
Example 2

Example 3
Example 4

Histogram Equalization: Idea

- How to improve contrast of dark/washed-out images?
- Try to find an intensity transformation function that spreads the squashed histogram derived from the original image
- Estimate an intensity transformation function from an image directly!
- Turns out that there is a simple way to compute the function from the histogram of the original image!!!
Cumulative Distribution Function

- Histogram equalization formula
  - \( r_k \): input intensity \([0.0, 1.0]\)
  - \( s_k \): output intensity \([0.0, 1.0]\)
  - \( k \): the intensity index within a range of \([0.0, 1.0]\)
  - \( n_j \): the count of intensity \( j \)
  - \( n \): the sum of all counts
- CDF: cumulative distribution func.

\[
s_k = T(r_k) = \sum_{j=1}^{k} p_r(r_j) = \sum_{j=1}^{k} \frac{n_j}{n}
\]

Histogram Equalization: Process

1. Compute CDF
2. Transfer Intensity
3. Do Hist Eq.
Example with Previous Images

[Image with IN and OUT marked]
Neighborhood Processing: Overview

- Common spatial domain operations
- Neighborhood is a set of surrounding pixels
- Neighborhood processing operations:

\[ s = T(r, r_1, \ldots, r_n) \]

- \( s \): the processed image pixel value
- \( r \): the original image pixel value at center
- \( r_1, \ldots, r_n \): the pixel values in a neighborhood
- \( n \): the number of pixels in the neighborhood
Neighborhoods & Connectivity

- Neighbourhoods are mostly a rectangle around a center pixel
- Any size rectangle and any shape filter are possible
- Standard symmetric neighborhoods are:
  - In 2D: 4-connected and 8-connected
  - In 3D: 6-connected and 26-connected

Simple Spatial Filters

Some simple neighbourhood operations include:

- **Min Filter**: Set the pixel value to the minimum in the neighbourhood
  \[ S = \min (r_1, r_2, \ldots, r_n) \]
- **Max Filter**: Set the pixel value to the maximum in the neighbourhood
  \[ S = \max (r_1, r_2, \ldots, r_n) \]
- **Median Filter**: The median value of a set of numbers is the midpoint value in that set (e.g. from the set \{1, 7, 15, 18, 24\} 15 is the median). Sometimes the median works better than the average
  \[ S = \text{median} (r_1, r_2, \ldots, r_n) \]
Simple Filters Exercise

**General Spatial Filtering Process**

The above is repeated for every pixel in the original image to generate the filtered image.
General Spatial Filtering Formula

\[ g(x, y) = \sum_{s=-d}^{a} \sum_{t=-b}^{b} w(s, t) f(x+s, y+t) \]

Filtering can be given in equation form as shown above.
Notations are based on the image shown to the left.

Smoothing Filters by Average

One of the simplest spatial filtering operations we can perform is a smoothing operation:

- Average all of the pixels in a neighbourhood around a central value.
- Especially useful in removing noise from images.
- Also useful for highlighting gross detail.

\[ S = \text{average}(r, r_1, \ldots, r_n) \]

In 3D?

Why 1/9?

Simple averaging filter.

\[ \frac{1}{9} \]

\[ \frac{1}{9} \]

\[ \frac{1}{9} \]

\[ \frac{1}{9} \]

\[ \frac{1}{9} \]
**Smoothing Filter Process**

The above is repeated for every pixel in the original image to generate the smoothed image.

**Average Filter Example**

- The image at the top left is an original image of size 500*500 pixels.
- The subsequent images show the image after filtering with an averaging filter of increasing sizes – 3, 5, 9, 15 and 35.
- Notice how detail begins to disappear.
Weighted Average Filter

- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to the central pixel are more important
  - Often referred to as a weighted averaging

![Weighted averaging filter](image)

In 3D?

Average vs Median

- These filters are often used to remove noise from images
- Median filter often works better than an averaging filter for noise removal!!!

![Original Image](image) ![Image After Averaging Filter](image) ![Image After Median Filter](image)
Problem is.....

- At the edges of an image we are missing pixels to form a neighbourhood
How to Handle Edge Problems?

• There are a few approaches to dealing with missing edge pixels:
  1. Omit missing pixels
     a. Only works with some filters
     b. Can add extra code and slow down processing
  2. Pad the image
     a. Typically with either all white or all black pixels
  3. Replicate border pixels
  4. Truncate the edge of image
     a. Lose a lot of image when filter size is large
  5. Allow pixels wrap around the image
     a. Can cause some strange image artefacts

Combination of Multiple Filters

• By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding
Filtering, Correlation, Convolution

- The filtering we have been talking about so far is referred to as **correlation** with the filter itself referred to as the **correlation kernel**
- **Convolution** is a similar operation, with just one subtle difference

\[
\begin{array}{ccc}
  a & b & c \\
  d & e & e \\
  f & g & h \\
\end{array} \quad \begin{array}{ccc}
  r & s & t \\
  u & v & w \\
  x & y & z \\
\end{array} \quad e_{\text{processed}} = v*{e} + \\
\quad \quad \quad \quad z*a + y*b + x*c + \\
\quad \quad \quad \quad w*d + u*e + \\
\quad \quad \quad \quad t*f + s*g + r*h
\]

- For symmetric filters it makes no difference

Proof: Convolution Cond.

- Difference between filtering formula and convolution

\[
g(x, y) = w * f(x, y) = \int \int w(s, t)f(x-s, y-t)dsdt
\]

\[
g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)f(x+s, y+t)
\]

- Symmetric Filters: \(w(s, t) = w(-s, -t)\)

\[
g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)f(x+s, y+t)
\]

\[
= \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(-s, -t)f(x-s, y-t) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)f(x-s, y-t) = [w*f](x, y)
\]

- Proof: Convolution Cond.
## Summary

**Foundation of Digital Image Processing I**
- Spatial Domain Techniques
- Point Processing for Image Enhancement
- Neighborhood Processing for Image Smoothing

**Next week:**
- Midterm #1: Lectures 2-5, 100 min, open hand-written notes, calculator, pen/pensile/eraser, blank papers
- Be prepared: **Would not finish if you are not!!!**
  - Try doing filtering exercises in today’s lecture’s slides
  - Make a good handwritten notes to bring in with you.

**Foundation of Digital Image Processing II**
- Spatial Domain Techniques Cond.
- Neighborhood Processing for Image Sharpening