ELIC 629
Digital Image Processing

Winter 2006

Image Enhancement in the Spatial Domain:
Basics of Spatial Filtering, Smoothing Spatial Filters, Order Statistics Filters

Bill Kapralos

Tuesday, February 14 2006

Overview (1):

- **Before We Begin**
  - Administrative details
  - Review → some questions to consider
- **Spatial Filtering**
  - Introduction → Review
  - Basics of 1D and 2D spatial filtering
- **Smoothing Spatial Filters**
  - Introduction
  - Smoothing linear filters

Before We Begin

Administrative Details (1):

- **Lab Four**
  - Should be a rather straightforward lab to complete
  - A lab report is required for this lab and is due in two weeks (e.g., February 28 2006)
- **Reading Week**
  - Next week → no lecture or lab
- **Test 1**
  - February 28 2006 → arrive on time as test will begin at 8:10am sharp!
    - No consideration given if you are late!

Some Questions to Consider (1):

- What is a linear operator and a non-linear operator ?
- How do we show an operator is linear or non-linear ?
- What is the spatial domain ?
- What is image enhancement in the spatial domain ?
- What is spatial filtering ?
- What is a smoothing spatial filter ?
- What is an averaging filter ?
- What is a histogram ?
- Can you think of how to enhance an image by using its histogram ?

Introduction to Spatial Domain Filtering

Introduction (1):

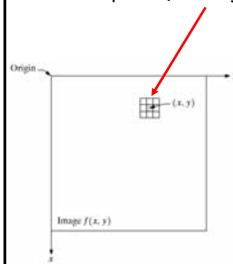
- **Spatial Domain**
  - The aggregate of pixels comprising an image
- **Spatial Domain Methods**
  - Procedures that operate directly on the image pixels
  - Denoted by the following expression

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

  - $f(x,y) \rightarrow$ input image
  - $g(x,y) \rightarrow$ output image
  - $T \rightarrow$ operator defined over some neighborhood of (x,y)

Introduction (2):

- **Defining a Neighborhood About x,y**
  - Square (rectangular) sub-image area centered at x,y



  - Center of sub-image is moved from pixel to pixel
  - Operator T is applied at each location x,y to yield output g
  - Operator T utilizes only pixels in area of image f spanned by neighborhood

Introduction (3):

- **Defining a Neighborhood About x,y (cont.)**
  - Square (rectangular) sub-image area centered at x,y
    - Also known as a mask, template, filter or window
    - Each entry of the sub-image has its own value → each value known as a coefficient
  - Mask does not always have to be two-dimensional
    - Can also have one-dimensional mask (more later…)

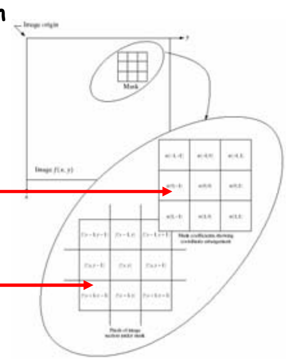
Basics of Spatial Filtering

Introduction – 2D Masks (1):

- **Graphical Illustration**

Mask (kernel, template…) with its coefficients

Pixels of sub-image under mask



Introduction – 2D Masks (2):

- **Template, Kernel, Mask**
  - Coefficients denoted by w
  - Origin is in the "middle"
  - Arbitrary size → dimensions do not need to be odd implying no "true" center (origin)

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

Introduction – 2D Masks (3):

- **"Mechanics" of Spatial Filtering**
  - Moving the template over each pixel of the image
    - At each pixel (x,y) the response (e.g., output value) is determined using some pre-defined relationship
    - For linear spatial filtering, response is given by a sum of the products of the filter coefficients (denoted by w) and the corresponding image pixels "under" the area of the template
    - Mathematically, response g at (x,y) given as

$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots + w(0,0)f(x,y) + \ldots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$

Introduction 2D – Masks (4):

- **"Mechanics" of Spatial Filtering (cont…)**
  - General filtering expression

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

  - m → row dimension
  - n → column dimension
  - m = 2a+1 and n = 2b+1 and a,b are non-negative integers or a = (m-1)/2 and b = (n-1)/2
  - But this gives output for one pixel location (x,y) only!

Introduction - 2D Masks (5):

- **"Mechanics" of Spatial Filtering (cont…)**
  - To generate complete output image, the above equation (process) must be applied to each pixel of input image e.g., for each x = 0 – M-1 and y = 0 – N-1 where M,N are the number of rows and columns of the input image
  - Similar to a frequency domain concept called convolution (more on this in the future…)
    - Hence sometimes referred to as "*convolving a mask with an image*" and the mask is often called a convolution mask

Introduction – 2D Masks (6):

- **"Mechanics" of Spatial Filtering (cont…)**
  - When we are not interested in processing entire image but rather only a particular pixel (x,y) we can use the following (shorter) mathematical definition:

$R = w_1z_1 + w_2z_2 + \ldots + w_{mn}z_{mn} +$

$$= \sum_{i=1}^{mn} w_i z_i$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

→ where, the w's are the filter coefficients and z's are the image gray-levels corresponding to the coefficients and mn is total number of coefficients in template

Introduction – 2D Masks (7):

- **Important Considerations**
  - What happens when kernel "placed over" a border pixel ?
  - Several approaches for dealing with this
    1. Ignore (don't handle) border pixels altogether or any pixels which lead to kernel (or portions of the kernel) "falling" out of image range
    2. "Wrap-around"
    3. "Zero-pad"
  - Keep in mind, some approaches may lead to an output image whose size is not equal to input image!

Introduction – 1D Masks (1):

- **Similar to 2D Case Except Now 1D**

w_1	w_2	w_3

Mask

  - Size of mask can be odd or even
    - If even → how do you define "center" ?
    - Can convert to odd by prepending a "zero" entry
    - Now origin is middle entry

1	2

→

0	1	2

Smoothing Spatial Filters

Introduction (1):

- **Purpose of Smoothing Spatial Filters**
  - Used for blurring, particularly in pre-processing
    - Removal of small detail prior to extraction of large object(s) in image
    - Bridging ("closing") of small gaps in lines or curves
  - Also used for noise reduction
    - Can be achieved with a linear or non-linear filter

Smoothing Linear Filters (1):

- **Essentially an Averaging Filter**
  - Output of smoothing filter is simply the average of pixels in the neighborhood of filter mask (kernel)
  - Also known as a low pass filter
    - Eliminates high frequency components (we will describe this further in later lectures)
  - Idea of smoothing filter
    - Random noise typically consists of sharp transitions in gray levels

Smoothing Linear Filters (2):

- Idea of smoothing filter (cont…)
  - By replacing the value of every pixel by the average of its neighbors, we essentially reduce the sharp transitions in gray levels
- Most obvious application of a smoothing filter is noise reduction
- However, beware!
  - Not all sharp transitions are bad and un-wanted!
  - Edges which are typically very important and wanted features of an image are also defined as sharp transitions in gray levels

Smoothing Linear Filters (3):

- However, beware! (cont…)
  - Since smoothing filter removes sharp transitions in gray level, averaging (smoothing) filters blur edges!
- Several other applications of smoothing (averaging) filters in addition to noise reduction
  - Smoothing of false contours which result when not using a large number of gray levels
  - Removing irrelevant details in an image → pixel regions that are small in comparison to the size of the filter kernel

Smoothing Linear Filters (4):

- **Smoothing Filter Kernels**
  - Example of a 3 x 3 smoothing filter

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

  - Above kernel (filter) produces average of the pixels under the mask
  - Notice that coefficients are equal to 1 and not 1/9!
    - One division instead of nine → more efficient!

Smoothing Linear Filters (5):

- **Smoothing Filter Kernels (cont…)**
  - Another example of a 3 x 3 smoothing filter

$$\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} \qquad R = \frac{1}{16}\sum_{i=1}^{9} z_i$$

  - This is an example of a weighted average filter
    - Coefficients are not all the same value!
    - Some pixels multiplied by higher values thus giving those pixels more importance in average

Smoothing Linear Filters (6):

- Another example of a 3 x 3 smoothing filter (cont…)

$$\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

  - Center coefficient is highest meaning center pixel is given most importance
- Other coefficients are reduced inversely as a function of distance from the center coefficient
  - Diagonal terms are further away than the "edge" neighbors and thus the corresponding pixels in image provide less importance to average

Smoothing Linear Filters (7):

- Another example of a 3 x 3 smoothing filter (cont…)
  - Many other types of coefficient masks are also available depending on application but typically try to keep the sum of the coefficients an integral power of 2 (e.g., 16 as in previous example)
- In general, hard to notice differences between images filtered by both these filter examples

Smoothing Linear Filters (8):

- **General Filter Implementation**
  - Recall the filtering expression for filtering M x N image with weighted averaging filter of size m x n

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

  - The general expression equation can now be stated as (again, given an N x M image with m x n filter where m,n are odd)

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

Smoothing Linear Filters (9):

- **General Filter Implementation (cont…)**

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

  - Complete filtered image is obtained by applying above equation for each x = 0,1,2, …, M-1 and y = 0,1,2, …, N-1
  - Denominator is sum of the mask (kernel) coefficients and is constant (e.g., computed once!)
    - Typically division applied once to output image rather than at each stage (pixel output)

Smoothing Linear Filters (10):

- **Graphical Examples**

Original image

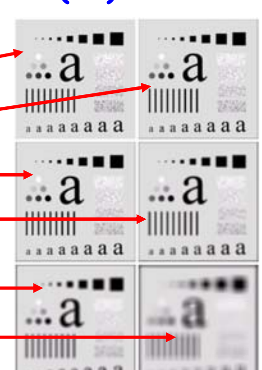
3 x 3 filter

5 x 5 filter

9 x 9 filter

15 x 15 filter

35 x 35 filter



Smoothing Linear Filters (11):

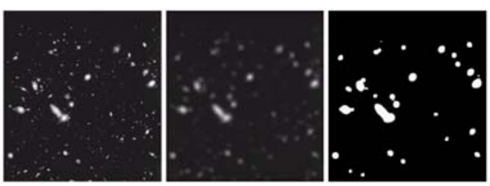
- **Some Notes Regarding Averaging Filters**
  - For small filter (e.g., 3 x 3) a slight general blurring of entire image occurs but details that are about same size of filter are affected considerably
  - Noise is less pronounced
  - Jagged borders are "pleasantly" smoothed
  - As filter size increases, blurring is more pronounced

Smoothing Linear Filters (12):

- **Image Blurring**
  - Important application of averaging filter is blurring to get "gross representation" of objects of interest
    - Smaller objects blend into the background
    - Larger objects become more "blob-like" and easy to detect
    - Size of mask (kernel) determines size of objects to be blended into background → smaller mask, smaller objects blended to background

Smoothing Linear Filters (13):

- **Image Blurring Graphical Example**
  - Image obtained with Hubble space telescope



Original image — Filtered with 15 x 15 averaging mask — Threshold image – small objects have disappeared

Order-Statistics Filters (1):

- **Non-linear Spatial Filters**
  - Response of filter is based on ordering (ranking) the pixels contained in image area encompassed by filter and then replacing center pixel with value determined by ranking result
  - One of the most known examples is the median filter

Order-Statistics Filters (2):

- **Median Filter**
  - Replaces the value of a pixel with median of the gray levels in the neighborhood of that pixel
  - Can provide excellent noise reduction with less blurring than averaging filters
    - Particularly effective for impulsive noise also known as salt-and-pepper noise due to its appearance as white and black dots in the image
  - What is the median ?
    - Given a set of values, the median ξ of the set is chosen such that half the values of set are less than ξ and half are more

Order-Statistics Filters (3):

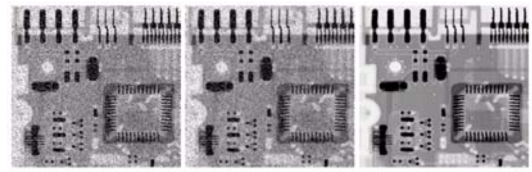
- **Median Filter (cont…)**
  - To perform median filtering:
    1. Sort the values of the pixel in question at spatial location (x,y) and its neighbors
    2. Determine the median value ξ
    3. Assign intensity value at location (x,y) the median value
  - Given 3 x 3 mask → median value is the 5^{th} largest value, 5 x 5 mask → median is 13^{th} largest value

Order-Statistics Filters (4):

- **Median Filter (cont…)**
  - Principle function of median filter is to force points with distinct gray levels to be more like their neighbors
    - Isolated clusters of pixels that are light or dark with respect to their neighbors are eliminated

Order-Statistics Filters (5):

- **Median Filter Graphical Example**
  - Comparison between averaging filter



Original image corrupted by salt-and-pepper noise | Image filtered with 3 x 3 average filter | Image filtered with 3 x 3 median filter