Spatial Filtering

CS 450: Introduction to Digital Signal and Image Processing

Slides created by Dr. Bryan Morse
BYU Computer Science

Neighborhood Operations

- Output is a function of a pixel's value *and its neighbors*
- Example (8-connected neighbors):

$$g(x,y) = \mathrm{Op}\begin{pmatrix} f(x-1,y-1) & , & f(x,y-1) & , & f(x+1,y-1) & , \\ f(x-1,y) & , & f(x,y) & , & f(x+1,y) & , \\ f(x-1,y+1) & , & f(x,y+1) & , & f(x+1,y+1) & \end{pmatrix}$$

- Possible operations: sum, weighted sum, average, weighted average, min, max, median, ...

Spatial Filtering

- The most common neighborhood operation is to multiply each of the pixels in the neighborhood by a weight and add them together.
- The local weights are sometimes called a *mask* or *kernel*.

Local Neighborhood

f(x-1,y-1)	f(x,y-1)	f(x+1,y-1)
f(x-1,y)	f(x,y)	f(x+1,y)
f(x-1,y+1)	f(x,y+1)	f(x+1,y+1)

Mask

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

$$g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s,t)\, f(x+s, y+t)$$

Spatial Filtering

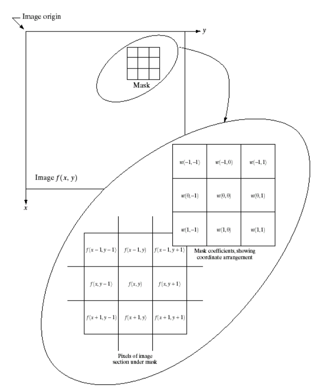


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Convolution

- Spatial filtering is often referred to as *convolution*.
- Or we say we *convolve* the image by a kernel or mask.
- Properly speaking, convolution uses a flipped kernel: (we'll go into why later)

Local Neighborhood

f(x-1,y-1)	f(x,y-1)	f(x+1,y-1)
f(x-1,y)	f(x,y)	f(x+1,y)
f(x-1,y+1)	f(x,y+1)	f(x+1,y+1)

Mask

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

$$g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s,t)\, f(x-s, y-t)$$

Spatial Filtering

Applications:

- Smoothing
- Sharpening
- Edge detection

⋮

Smoothing

- Already talked about averaging multiple images together to reduce the variance of the noise.
- With one image (signal), you can't average multiple images together.
- Why not average multiple *pixels* together?
- What does this assume?

Advantage: Less Noise
Disadvantage: Blurring

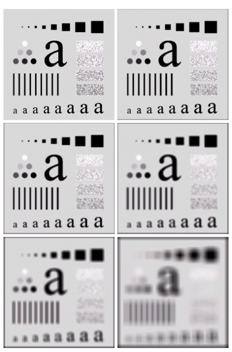
Smoothing

- Any kernel with all-positive weights causes *smoothing* or *blurring*.
- They are also called *low-pass filters* (more on this later)
  - Audio: diminishes high frequency sounds while allowing *low*-frequency sounds to *pass*
  - Images: reduces rapid changes/transitions
- To cause averaging (instead of just summation), have to normalize by the sum of the weights:

$$g(x, y) = \frac{\sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s, t)\, f(x + s, y + t)}{\sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s, t)}$$

Image Smoothing

Blurring with successively larger uniform kernels:



Smoothing Kernels

Changing the weights shapes the smoothing:

1	1	1
1	1	1
1	1	1

1	1	1
1	2	1
1	1	1

1	2	1
2	4	2
1	2	1

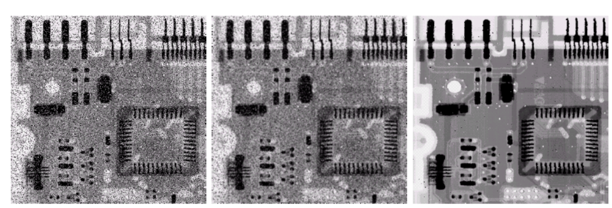
Can be any size:

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Nonlinear Smoothing

- Convolution is linear, but many neighborhood operators are not.
- One set of nonlinear smoothing operators are *order-statistic filters*:
  - Median
    - Unlike averaging, the result is a value from the original image
    - One method of (attempted) *edge-preserving smoothing*
    - Very good for "salt and pepper" noise (occasional dark, light pixels)
  - Trimmed-mean
    - Throw out the high and low values, average the rest
    - Less likely to be affected by outlying neighborhood values

Median Filtering



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)