Image Processing

Pengwei Hao
p.hao@qmul.ac.uk

Topic 4: Filtering
ECS605U / ECS776P
School of EECS
Queen Mary University of London

Linear Operators

- Let H be an operator whose input and output are images.
- H is a linear operator, if for any two images f and g and two scalars a and b,

$$H(af + bg) = aH(f) + bH(g)$$

Linear and Nonlinear Operators

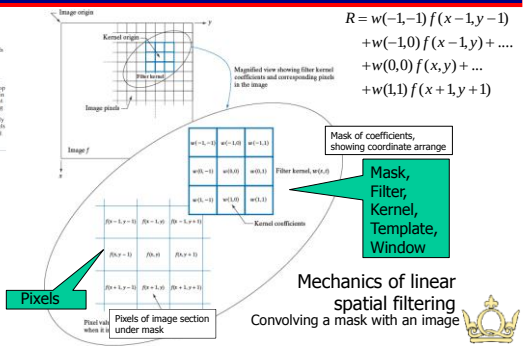
- Linear operators (e.g. box average filter)

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Nonlinear operators (median filter)
  - Replace the current point in the image by the median of the gray values in its neighborhood.

Image filtering/convolution

Basics of Spatial Filtering



Mechanics of linear spatial filtering
Convolving a mask with an image

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1						-1
0						-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9					-1
0						-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9				-1
0						-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1			-1
0						-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9		-1
0						-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0						-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3					-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1				-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9			-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9		-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9	1/9	-1
1						2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9	1/9	-1
1	11/9					2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9	1/9	-1
1	11/9	12/9				2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9	1/9	-1
1	11/9	4/3	10/9			2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9	1/9	-1
1	11/9	4/3	10/9	1		2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	7/9	11/9	1	4/9	-2/9	-1
0	2/3	1	8/9	4/9	1/9	-1
1	11/9	4/3	10/9	1	1	2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Computation of Filtering

1	2	2	3	1	0	0
-1	0	1	1	0	-1	-1
0	1	1	0	0	0	-1
1	1	2	2	1	1	2
2	2	1	2	1	2	3

Original image $f(x,y)$

1	2	2	3	1	0	0
-1	1	1	1	0	0	-1
0	1	1	1	0	0	-1
1	1	1	1	1	1	2
2	2	1	2	1	2	3

Filtered image $g(x,y)$

mask

1	1	1
1	1	1
1	1	1

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

Averaging / Smoothing

- The objective of smoothing is to blur and to reduce noise.
- The filters for blurring are called *averaging filters* or *lowpass filters*.
- Averaging filters have the undesirable side effect that they blur edges.

Smoothing Filter Masks

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

1	1	1
1	1	1
1	1	1

Box filter

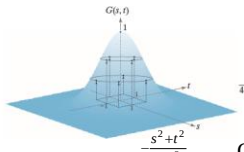
$\frac{1}{16}\times$

1	2	1
2	4	2
1	2	1

Weighted average

a b

FIGURE 3.35
(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.31(b)].



$$w(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

$\frac{1}{4.8976}\times$

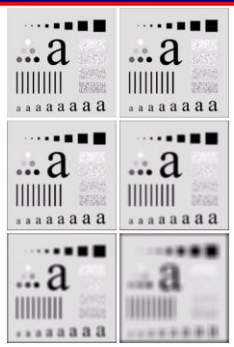
0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

Gaussian kernel

Box Filter Averaging

Original image

Black squares are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels



3x3 filtered

5x5 filtered

9x9 filtered

15x15 filtered

35x35 filtered

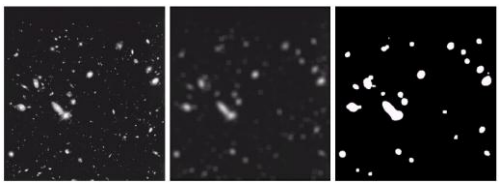
Gaussian Averaging

Original Gaussian of 21x21, σ=3.5 Gaussian of 43x43, σ=7

a b c

FIGURE 3.36 (a) A test pattern of size 1024 × 1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21 × 21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43 × 43, with $\sigma = 7$. This result is comparable to Fig. 3.33(d). We used $K = 1$ in all cases.

Averaging: Application



Original Averaging Averaging+Thresholding

Border Extension – Padding

```
000000  111222  433443  121212
000000  111222  211221  343434
001200  111222  211221  121212
003400  333444  433443  343434
000000  333444  433443  121212
000000  333444  211221  343434
```

Zero Padding | Replicate Padding | Mirror Padding | Circular Padding

Border Extension

Zero padding Mirror padding Replicate padding

a b c

FIGURE 3.39 Result of filtering the test pattern in Fig. 3.36(a) using (a) zero padding, (b) mirror padding, and (c) replicate padding. A Gaussian kernel of size 187 × 187, with $K = 1$ and $\sigma = 31$ was used in all three cases.

Correlation and Convolution (1D)

correlation:

$$g(t) = w(t) \circ f(t) = \sum_s w(s) f(t+s)$$

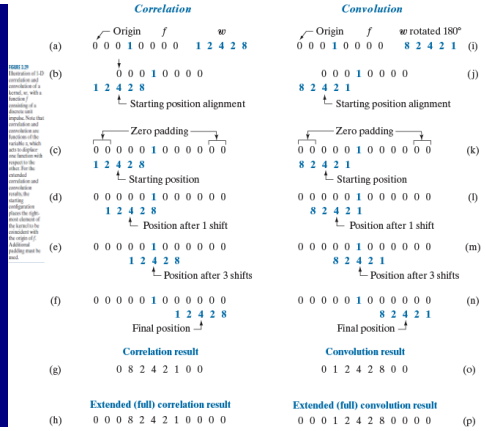
convolution:

$$g(t) = w(t) * f(t) = \sum_s w(s) f(t-s)$$

$$= \sum_s w(-s) f(t+s)$$

$$= w(-t) \circ f(t)$$

Correlation and Convolution (1D)



Correlation and Convolution (2D)

correlation:

$$g(x,y) = w(x,y) \circ f(x,y) = \sum_s \sum_t w(s,t) f(x+s, y+t)$$

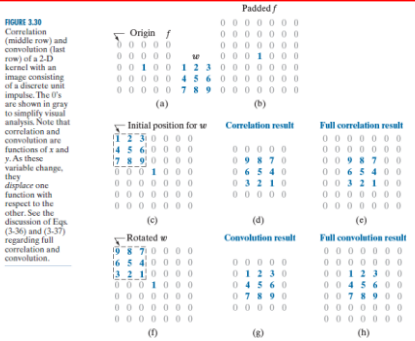
convolution:

$$g(x,y) = w(x,y) * f(x,y) = \sum_s \sum_t w(s,t) f(x-s, y-t)$$

$$= \sum_s \sum_t w(-s,-t) f(x+s, y+t)$$

$$= w(-x,-y) \circ f(x,y)$$

Correlation and Convolution (2D)



Correlation and Convolution (2D)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 4 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & -2 & 0 & 1 \\ 2 & -2 & -8 & -2 & 2 \\ 1 & 0 & -2 & 0 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

Coding: image correlation

```
// for Mask of size 3x3
for(int y=1; y<height-1; y++){
    for(int x=1; x<width-1; x++){
       r = 0;  g = 0;  b = 0;
       for(int s=-1; s<=1; s++){
          for(int t=-1; t<=1; t++){
             r = r + Mask[1+s][1+t]*ImageArray1[x+s][y+t][1];  //r
             g = g + Mask[1+s][1+t]*ImageArray1[x+s][y+t][2];  //g
             b = b + Mask[1+s][1+t]*ImageArray1[x+s][y+t][3];  //b
          }
       }
       ImageArray2[x][y][1] = r;  //r
       ImageArray2[x][y][2] = g;  //g
       ImageArray2[x][y][3] = b;  //b
   }
}
```

Coding: image convolution

```
// for Mask of size 3x3
for(int y=1; y<height-1; y++){
    for(int x=1; x<width-1; x++){
       r = 0;  g = 0;  b = 0;
       for(int s=-1; s<=1; s++){
          for(int t=-1; t<=1; t++){
             r = r + Mask[1-s][1-t]*ImageArray1[x+s][y+t][1];  //r
             g = g + Mask[1-s][1-t]*ImageArray1[x+s][y+t][2];  //g
             b = b + Mask[1-s][1-t]*ImageArray1[x+s][y+t][3];  //b
          }
       }
       ImageArray2[x][y][1] = r;  //r
       ImageArray2[x][y][2] = g;  //g
       ImageArray2[x][y][3] = b;  //b
   }
}
```

Order-Statistics Filters

- Order-statistics filters are nonlinear filters whose response is based on ordering (ranking).
- The best-known example is *median filter*.
- Median filters are particularly effective in the presence of *impulse noise* (*salt-and-pepper noise*)

Median Filter

$$\hat{f}(x,y) = \underset{(x,y)\in S_{xy}}{median}\{g(s,t)\}$$

5 5 6
3 4 5 → (3,3,4,4,**5**,5,5,6,7) → 3 5 5
3 4 7 Sorting 3 4 7

Original Filtered

Min Filter

$$\hat{f}(x,y) = \min_{(x,y)\in S_{xy}}\{g(s,t)\}$$ Good for salt

5 5 6
3 4 5 → (**3**,3,4,4,5,5,5,6,7) → 3 3 5
3 4 7 Sorting 3 4 7

Original Filtered

Max Filter

$$\hat{f}(x,y) = \max_{(x,y)\in S_{xy}}\{g(s,t)\}$$ Good for pepper

5 5 6
3 4 5 → (3,3,4,4,5,5,5,6,**7**) → 3 7 5
3 4 7 Sorting 3 4 7

Original Filtered

Mid-Point Filter

$$\hat{f}(x,y) = \frac{1}{2}\left\{\max_{(x,y)\in S_{xy}}\{g(s,t)\} + \min_{(x,y)\in S_{xy}}\{g(s,t)\}\right\}$$

5 5 6
3 4 5 → (**3**,3,4,4,5,5,5,6,**7**) → 3 5 5 ($\frac{3+7}{2}=5$)
3 4 7 Sorting 3 4 7

Original Filtered

Alpha-trimmed Filter

d/2 lowest and d/2 highest values are deleted

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(x,y)\in S_{xy}} g_r(s,t)$$

5 5 6
3 4 5 → d=2 → (3,**3,4,4,5,5,5,6**,7) → 3 5 5 ($\frac{(3+4+4+5+5+5+6)/7}{=4.5714\approx 5}$)
3 4 7 Sorting 3 4 7

Original Filtered

Alpha-trimmed Filter

d/2 lowest and d/2 highest values are deleted

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(x,y)\in S_{xy}} g_r(s,t)$$

5 5 6		d=4	$\xrightarrow{(4+4+5)/5}$	5 5 6
3 4 5	→	(3,3,4,4,5,5,5,6,7)	$=4.6\approx 5$	3 5 5
3 4 7		Sorting		3 4 7
Original				Filtered

Alpha-trimmed Filter

d/2 lowest and d/2 highest values are deleted

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(x,y)\in S_{xy}} g_r(s,t)$$

5 5 6		d=8		5 5 6
3 4 5	→	(3,3,4,4,5,5,5,6,7)	→	3 5 5
3 4 7		Sorting		3 4 7
Original				Filtered

d = mn - 1

Median

Alpha-trimmed Filter

d/2 lowest and d/2 highest values are deleted

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(x,y)\in S_{xy}} g_r(s,t)$$

5 5 6		d=0	$\xrightarrow{(3+3+4+4+5+5+5+6+7)/9}$	5 5 6
3 4 5	→	(3,3,4,4,5,5,5,6,7)	$=4.6667\approx 5$	3 5 5
3 4 7		Sorting		3 4 7
Original				Filtered

d = 0

Arithmetic

Alpha-trimmed Filter

d/2 lowest and d/2 highest values are deleted

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(x,y)\in S_{xy}} g_r(s,t)$$

d = 0

Arithmetic

d = mn - 1

Median

Alpha-trimmed filter is a generalised filter.

Order-Statistics Filters

Median filter

$$\hat{f}(x,y)=\underset{(x,y)\in S_{xy}}{median}\{g(s,t)\}$$

Max and min filter

$$\hat{f}(x,y)=\max_{(x,y)\in S_{xy}}\{g(s,t)\}$$ Good for pepper

$$\hat{f}(x,y)=\min_{(x,y)\in S_{xy}}\{g(s,t)\}$$ Good for salt

Midpoint filter

$$\hat{f}(x,y)=\frac{1}{2}\left\{\max_{(x,y)\in S_{xy}}\{g(s,t)\}+\min_{(x,y)\in S_{xy}}\{g(s,t)\}\right\}$$

Alpha-trimmed mean filter

d/2 lowest and d/2 highest values are deleted

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(x,y)\in S_{xy}} g_r(s,t)$$

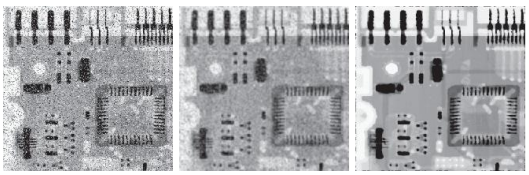
d = 0

Arithmetic

d = mn - 1

Median

Example of Median Filter



Original Gaussian 7x7 Median

a b c

FIGURE 3.43 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma=3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Coding: median filter

```
// for Window of size 3x3
for(int y=1; y<height-1; y++){
    for(int x=1; x<width-1; x++){
        k = 0;
        for(int s=-1; s<=1; s++){
           for(int t=-1; t<=1; t++){
              rWindow[k] = ImageArray1[x+s][y+t][1];  //r
              gWindow[k] = ImageArray1[x+s][y+t][2];  //g
              bWindow[k] = ImageArray1[x+s][y+t][3];  //b
              k++;
           }
        }
        Arrays.sort(rWindow);
        Arrays.sort(gWindow);
        Arrays.sort(bWindow);
        ImageArray2[x][y][1] = rWindow[4];  //r
        ImageArray2[x][y][2] = gWindow[4];  //g
        ImageArray2[x][y][3] = bWindow[4];  //b
    }
}
```

Sharpening

- The objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred.
- Blurring could be accomplished by averaging (integration).
- Sharpening could be accomplished by difference (differentiation)

Derivatives

- Some detail sharpening filters are based on first- and second-order derivatives.
- The derivatives of a digital function are defined in terms of differences.
  - First-order derivative

$$\left.\begin{aligned} G_x &= \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \\ G_y &= \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y) \end{aligned}\right\} \Rightarrow \left\{\begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} -1 & 1 \end{bmatrix}\right. \text{ or } \left\{\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}\right.$$

Derivatives

  - Second-order derivative

$$\frac{\partial^2 f(x,y)}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f(x,y)}{\partial x}\right)$$

$$= \{f(x+1,y) - f(x,y)\} - \{f(x,y) - f(x-1,y)\}$$

$$= f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{\partial^2 f(x,y)}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial f(x,y)}{\partial y}\right)$$

$$= \{f(x,y+1) - f(x,y)\} - \{f(x,y) - f(x,y-1)\}$$

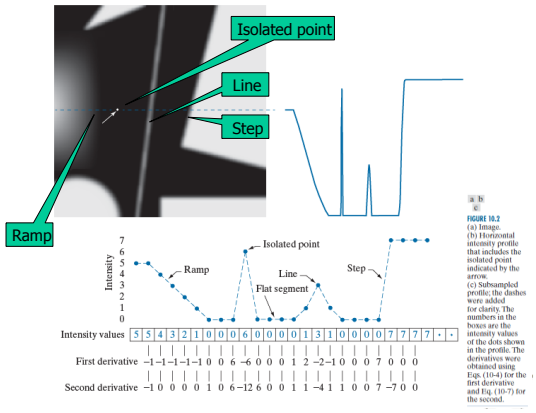
$$= f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

First- and Second-Derivatives

Type	1st	2nd
Edge	thicker	
Fine detail (thin line and isolated points)		stronger
Step	stronger	double response
		Point>Line>Step

- 1st derivative is better for edge detection,
  2nd derivative is better for image enhancement



Laplacian

- The Laplacian is defined as

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

- The Laplacian is rotation invariant.
- Filter mask:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Derivation of Laplacian mask

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

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

$$\nabla^2 f = \{f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)\} - 4f(x,y)$$
$$= 4 \times \{\bar{f}(x,y) - f(x,y)\}$$

$$\bar{f}(x,y) = \frac{1}{4}\{f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)\}$$

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

Laplacian

$$\nabla^2 f = 4 \times \{\bar{f}(x,y) - f(x,y)\}$$

$$\bar{f}(x,y) = \frac{1}{4}\{f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)\}$$

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

0	1	0
1	-4	1
0	1	0

Laplacian Masks

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

4-neighborhood

0	−1	0
−1	4	−1
0	−1	0

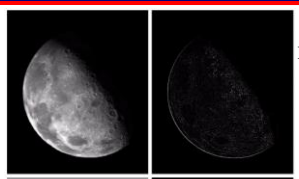
−1	−1	−1
−1	8	−1
−1	−1	−1

8-neighborhood

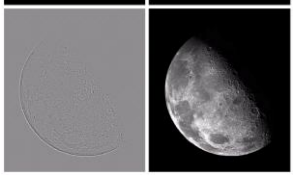
FIGURE 3.45 (a) Laplacian kernel used to implement Eq. (3-53). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

Example of Laplacian

Original



Laplacian



Laplacian+Scale

$f(x,y) - \nabla^2 f$

Simplification Using Laplacian

$$\nabla^2 f = \{f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)\} - 4f(x,y)$$
$$= 4 \times \{\bar{f}(x,y) - f(x,y)\}$$

Simplification for image enhancement using Laplacian:

$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$
$$= f(x,y) - 4\{\bar{f}(x,y) - f(x,y)\}$$
$$= 5f(x,y) - 4 \times \bar{f}(x,y)$$

Enhancement Example

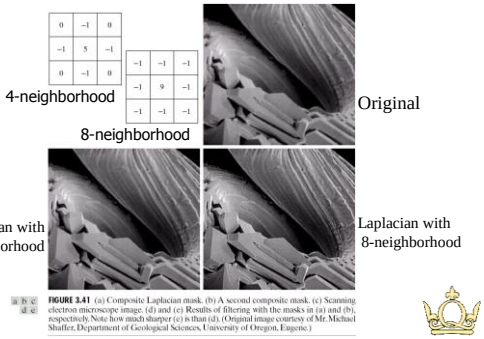


FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Unsharp Masking and High-Boost Filtering for Enhancement

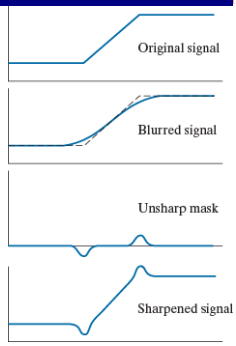
- Unsharp masking

$$f_{unsharp}(x,y) = f(x,y) - f_{blur}(x,y)$$

where $f_{blur}(x,y) = \bar{f}(x,y)$

- High-boost filtering

$$f_{hb}(x,y) = f(x,y) + kf_{unsharp}(x,y)$$
$$= f(x,y) + k\left[f(x,y) - \bar{f}(x,y)\right]$$
$$= (k+1)f(x,y) - k\bar{f}(x,y)$$



Unsharp Masking and High-Boost Filtering

- Unsharp masking

$$f_{unsharp}(x,y) = f(x,y) - f_{blur}(x,y)$$

where $f_{blur}(x,y) = \bar{f}(x,y)$

- High-boost filtering

$$f_{hb}(x,y) = f(x,y) + kf_{unsharp}(x,y)$$
$$= f(x,y) + k\left[f(x,y) - \bar{f}(x,y)\right]$$
$$= (k+1)f(x,y) - k\bar{f}(x,y)$$

Original

Gaussian blurred

Unsharp masking

High-boost with k=1

High-boost with k=4.5

Gradient

- $\nabla \mathbf{f}(x,y)$: Gradient of an image $f(x,y)$ at location (x,y)
- $\nabla f(x,y)$: Gradient magnitude
- $\alpha(x,y)$: Gradient direction

$$\nabla \mathbf{f}(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\nabla f(x,y) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \qquad \nabla f(x,y) \approx |G_x| + |G_y|$$

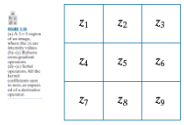
$$\theta(x,y) \approx \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

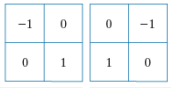
Gradient Masks

Roberts:

$$\nabla f = \sqrt{\left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]}$$

$$\nabla f = |z_9 - z_5| + |z_8 - z_6|$$



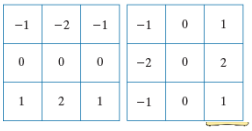


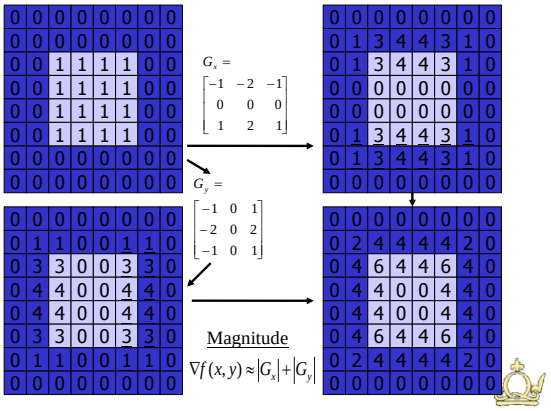
Sobel:

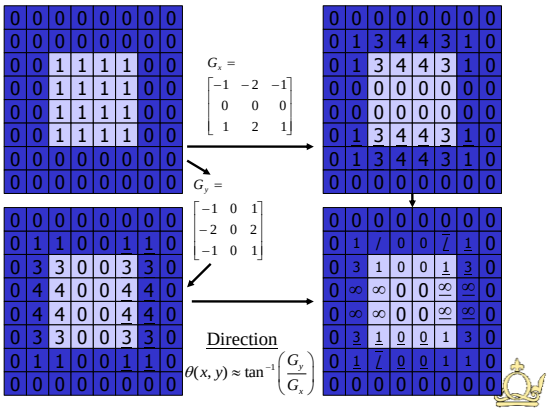
$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

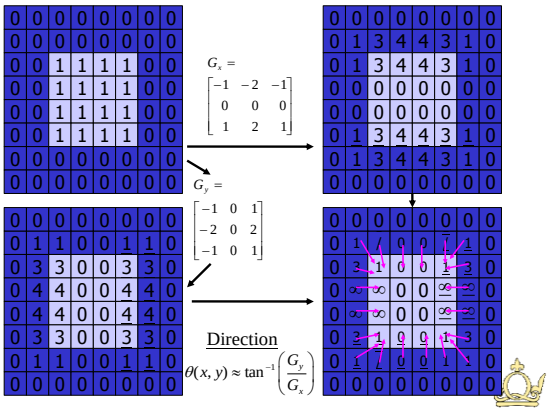
$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

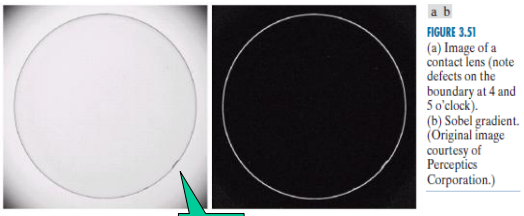








Gradient Enhancement



a b

FIGURE 3.51
(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Perceptics Corporation.)

Original Defect Sobel

Combining Enhancement Methods

- An enhancement task may require application of several complementary techniques.
- Enhancement is **problem oriented**.
- A certain amount of **trial and error** is required before an image enhancement approach is selected.