7. Simple filter for image using convolution
7.1 Smoothing using Moving Average

$$g_{i,j} = \sum_{m,n} w_{m,n} f_{i+m,j+n} \qquad m,n \in -N,\dots,N, \quad N_m = 2N+1$$

size	Simple ave.	Square pyramid	circular	Gaussian like
3×3 $\begin{pmatrix} N_m = 3 \\ N = 1 \end{pmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times 1/9$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times 1/10$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \times 1/5$	$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \times 1/16$
5×5 $\begin{pmatrix} N_m = 5 \\ N = 2 \end{pmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times 1/25$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times 1/35$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times 1/13$	$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \times 1/256$

7.2 Smoothing using Non-linear Filter

Since the moving average losses higher frequency component, edges and corners in the original image are blurred.

$f_{i+m,j+n}$

0	0	9	9	9
0	0	9	9	9
0	0	9	9	9
0	0	18	18	18
0	0	18	18	18

$*$

$w_{m,n}$

1	1	1
1	1	1
1	1	1

$=$

$g_{i,j}$

0	3	6	9	9
0	3	6	9	9
0	4	8	12	12
0	5	10	15	15
0	6	12	18	18

- Median filter
  Reducible the blurring edges
- Adaptive local averaging filter
  Reducible the blurring corners

Note:

These filters are irreversible.

Median Filter

Replace the pixel value at the central pixel in partial region by the median in the region.

$f_{i,j}$

①	③	②	3	1
④	⑤	①	9	5
⑥	⑤	⑧	6	6
1	2	7	□5	□7
1	2	8	□4	□4

sort and select median

○ : 1, 1, 2, 3, 4, 5, 5, 6, 8
Median

□ : 4, 4, 5, 7
Median: $\frac{4+5}{2} = 4.5$

⇓

$g_{i,j}$

*	*	*	*	*
*	4	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	4.5

⇐

$g_{i,j}$

3.5	2.5	4	2.5	4
4.5	4	5	5	5.5
4.5	5	5	6	6
2	5	5	6	5.5
1.5	2	4.5	6	4.5

Reduction of blurring of edges.

Input:

0	0	9	9	9
0	0	9	9	9
0	0	9	9	9
0	0	18	18	18
0	0	18	18	18

Moving Average

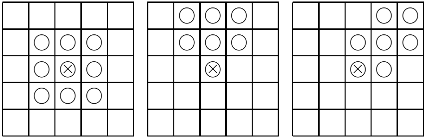
0	3	6	9	9
0	3	6	9	9
0	4	8	12	12
0	5	10	15	15
0	6	12	18	18

Median Filter

0	0	9	9	9
0	0	9	9	9
0	0	9	9	9
0	0	9	18	18
0	0	18	18	18

Adaptive Local Averaging Filter

Sampling patterns:



9 points (Symmetric($\times$1)) (C) | 7 points ($\times$ 4 patterns) (N,W,S,E) | 7 points ($\times$ 4 patterns) (NE,NW,SW,SE)

- Evaluate variance for each pattern.

$$\sigma^2_{(p)} \qquad (p \in \{\text{C, N,W,S,E, NE,NW,SW,SE}\})$$

- $g_{i,j}$ is taken as the average $\overline{f^{(p)}_{i,j}}$ with minimum variance $\sigma^2_{(p)}$.

Reduction of blurring of edges.

Input

0	0	9	9	9
0	0	9	9	9
0	0	9	9	9
0	0	18	18	18
0	0	18	18	18

Moving Average

0	3	6	9	9
0	3	6	9	9
0	4	8	12	12
0	5	10	15	15
0	6	12	18	18

Median Filter

0	0	9	9	9
0	0	9	9	9
0	0	9	9	9
0	0	9	18	18
0	0	18	18	18

Adapt. Local Ave.

0	0	9	9	9
0	0	9	9	9
0	0	9	9	9
0	0	18	18	18
0	0	18	18	18

7.3 Remove defect pixel

- Defect pixel : Pixels with $f_{i,j} = 0$ or $f_{i,j} = 255$(Max.)
- Reason : Difference of gain between pixels.
- Apply the Median filter to only the defect pixels.

Original

Defects

defects

Median for whole image

removed
smoothed

Median for defect pixel

removed
Not smoothed

7.4 Edge detection

Edge: Points having local maximum of the gradient ($|\nabla f|$).

Edge detection by gradient

- Derivative

$$\frac{df}{dx} = \lim_{\Delta\to 0} \frac{f(x+\frac{\Delta}{2}) - f(x-\frac{\Delta}{2})}{\Delta} = \lim_{\Delta\to 0} \frac{f(x+\Delta) - f(x-\Delta)}{2\Delta}$$

- Difference($\Delta \neq 0$)

$$\frac{\Delta f}{\Delta x} = \frac{f(x+\Delta) - f(x-\Delta)}{2\Delta} = \frac{1}{2}(f_{i+1} - f_{i-1})$$

$$\frac{df}{dx} \simeq \frac{\Delta f}{\Delta x}$$

$$g_{i,j} = \sum_{m,n} w_{m,n} f_{i+m,j+n} \quad \left(\begin{array}{c} m,n \in -N,\dots,N \\ N_m = 2N+1 \end{array} \right)$$

Example of $w_{m,n}$

- $\nabla f \cdot \boldsymbol{e}_x$

0	0	0
$\frac{-1}{2}$	0	$\frac{1}{2}$
0	0	0

- $\nabla f \cdot \boldsymbol{e}_y$

0	$\frac{1}{2}$	0
0	0	0
0	$\frac{-1}{2}$	0

- $\nabla f \cdot (\boldsymbol{e}_x + \boldsymbol{e}_y)$

0	0	$\frac{1}{2\sqrt{2}}$
0	0	0
$\frac{-1}{2\sqrt{2}}$	0	0

- $\nabla f \cdot (\boldsymbol{e}_x - \boldsymbol{e}_y)$

$\frac{-1}{2\sqrt{2}}$	0	0
0	0	0
0	0	$\frac{1}{2\sqrt{2}}$

Example of Edge detection

Original

$\nabla f \cdot \boldsymbol{e}_x$

$\nabla f \cdot \boldsymbol{e}_y$

$|\nabla f|$

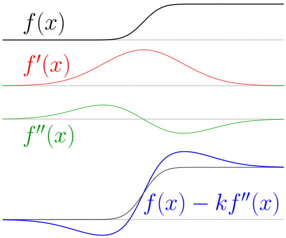
- $\nabla f \cdot \boldsymbol{e}_x$, $\nabla f \cdot \boldsymbol{e}_y$ embosses image.
- By $\nabla f \cdot \boldsymbol{e}_x$, the vertical edge can be detected, but horizontal one cannot.
- $|\nabla f|$ is useful for edge detection.

$|\nabla f|$ (inverted)

7.5 Enhancement of Edge

Using Laplacian

Consider curvature ($f''(x)$)



Enhancement of Edge :

$$g(\boldsymbol{r}) = f(\boldsymbol{r}) - k\nabla^2 f(\boldsymbol{r})$$

Second derivative

- 1-dim. $f''_i = f'_{i+1/2} - f'_{i-1/2}$
  $= f_{i+1} + f_{i-1} - 2f_i$
- 2-dim. Laplacian($\nabla^2 f$)
  $f''_{i,j} = f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j}$

In the case $k = 1$,

f

0	0	0
0	1	0
0	0	0

$-$ $(\nabla^2 f)$

0	1	0
1	−4	1
0	1	0

$=$ $f - (\nabla^2 f)$

0	−1	0
−1	5	−1
0	−1	0

Example of Edge Enhancement(using Laplacian)

Smoothed

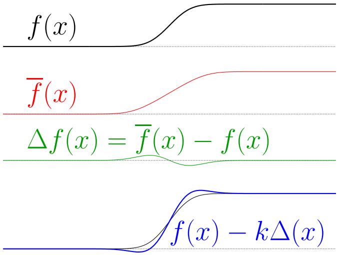
Enhance($k = 2$)

Enhance($k = 4$)

Original

Using diff. from Smoothed

Consider diff. from smoothed one $(\overline{f}(x) - f(x))$



Enhancement of Edge :

$$g(\boldsymbol{r}) = f(\boldsymbol{r}) - k\left(\overline{f(\boldsymbol{r})} - f(\boldsymbol{r})\right)$$
$$= (1+k)f(\boldsymbol{r}) - k\overline{f(\boldsymbol{r})}$$

In the case $k = 1$ and 3x3 simple ave.,

$$g = 2f - \overline{f}$$

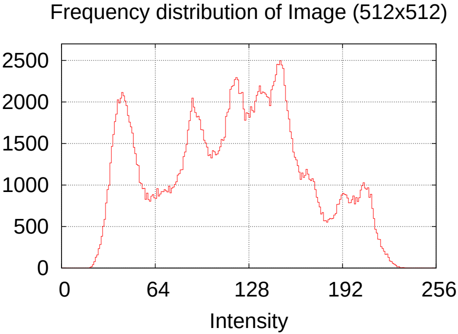
$$\underset{2f}{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}} - \underset{\overline{f}}{\begin{array}{|c|c|c|}\hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array}} = \underset{2f - \overline{f}}{\begin{array}{|c|c|c|}\hline \frac{-1}{9} & \frac{-1}{9} & \frac{-1}{9} \\ \hline \frac{-1}{9} & \frac{17}{9} & \frac{-1}{9} \\ \hline \frac{-1}{9} & \frac{-1}{9} & \frac{-1}{9} \\ \hline \end{array}}$$

Example of Edge Enhancement(using diff. from smoothed)

Smoothed | Enhance($k = 2$) | Enhance($k = 4$) | Original

7.6 Frequency distribution (Histogram)

Histogram → Used for binarizing or labeling



In this example the thresholds are about 64, 100, 126, 180.

7.7 Binarizing

Binarizing : Distinguish either true or false of a condition

1: True, 0: False

- Comparison with threshold

  e.g. $f(x,y) \geq f_{\mathrm{th}}$
- Solutions (Points) of equations

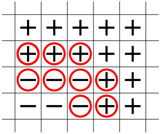
  e.g. $f(x,y) = a$

Solution of $f(x,y) = a$

$$(f(x,y) - a) \cdot (f(x',y') - a) \leq 0$$

($(x',y')$is adjacent pixels of (x,y))

In the case of 4 direction search:



$f \geq 128$

$f = 128$

$\left|\frac{\partial f}{\partial x}\right| > 5$

$\frac{\partial f}{\partial x} = 5$

7.8 Partitioning

Partitioning by Edge

Edge: Points with local max. of $|\nabla f|$.

$\rightarrow$ Solution of $\nabla^2 f = 0$?

$$\left(\begin{array}{l} \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ \quad \text{Even if } \frac{\partial^2 f}{\partial x^2} \neq 0 \text{ and } \frac{\partial^2 f}{\partial y^2} \neq 0 \text{ ,} \\ \quad \text{in the case of } \frac{\partial^2 f}{\partial x^2} = -\frac{\partial^2 f}{\partial y^2}, \\ \quad \nabla^2 f = 0. \quad \rightarrow \times \end{array} \right)$$

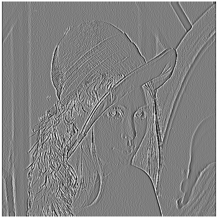
$$\therefore \boldsymbol{r} = \left\{ (x, y) \,\middle|\, \frac{\partial^2 f}{\partial x^2} = 0, \ \frac{\partial^2 f}{\partial y^2} = 0 \right\}$$

However, this is also inappropriate.

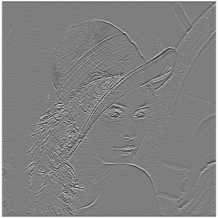
$\because$ The second derivatives have high frequency component, there are many zeros.

(Max: White, Min: Black)

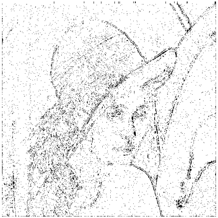
$\frac{\partial^2 f}{\partial x^2}$ [-20,20]



$\frac{\partial^2 f}{\partial y^2}$ [-20,20]



$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial y^2} = 0$



$\left(\begin{array}{l} = 0 : \text{ White} \\ \neq 0 : \text{ Black} \end{array} \right)$

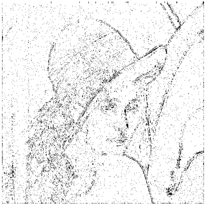
Improvement of the solution search algorithm

Methods to find the sol. of $\frac{\partial^2 f}{\partial x^2} = 0$.

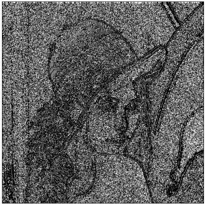
- Simple search
  $f''_i \cdot f''_{i+1} < 0$ or $f''_i \cdot f''_{i-1} \leq 0$
  $\rightarrow$ Broaden

- Use of average of half shifted points.
  $f''_{i+1/2} \cdot f''_{i-1/2} \leq 0 \rightarrow f''_i = 0$
  $\left(f''_{i\pm 1/2} = \frac{f''_i + f''_{i\pm 1}}{2}\right)$
  $\rightarrow$ This can avoid broadening.

Simple method



Ave. of Half shifted pix.



(White:Solution)

Solution is improved. However, it remains many solutions because of higher freq. components.
$\rightarrow$Coupling method with other methods is required.

Edge detection by $|\nabla f|$

Edge: Points having local maximum of the gradient ($|\nabla f|$).

$\Longrightarrow$ Points having large gradient.

Problem :

- Appropriate threshold

  $$|\nabla f| > |\nabla f|_0$$

- Broaden edges
  - Use with the second derivative.
  - Use of variance of gradient.

Variance of gradient

$$\overline{\nabla f} = (\overline{f'_x}, \overline{f'_y})$$

$$\sigma^2_{\nabla f} = \sigma^2_{f'_x} + \sigma^2_{f'_y}, \quad \widehat{\sigma}_{\nabla f} = \frac{\sigma_{\nabla f}}{|\nabla f|}$$

Condition of edge:

- $|\nabla f| > |\nabla f|_0$
- $\widehat{\sigma}_{\nabla f} < \widehat{\sigma}_{\nabla f\,0}$
- $\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial y^2} = 0$

Example of Edge detection by $|\nabla f|$

$|\nabla f| \geq 8$,
$f_{\prime x^\prime x} = f_{\prime y^\prime y} = 0$

$|\nabla f| \geq 8$,
$\sigma_{\nabla f} / |\nabla f| \leq 0.7$

$\sigma_{\nabla f} / |\nabla f| \leq 0.7$,
$f_{\prime x^\prime x} = f_{\prime y^\prime y} = 0$

$|\nabla f| \geq 8$,
$\sigma_{\nabla f} / |\nabla f| \leq 0.7$,
$f_{\prime x^\prime x} = f_{\prime y^\prime y} = 0$

$|\nabla f| \geq 8$,
$\sigma_{\nabla f} / |\nabla f| \leq 0.7$,
$f_{\prime x^\prime x} = f_{\prime y^\prime y} = 0$

Disconnected edge

Expansion and Contraction of Binary image

- Expansion

$$g_{i,j} = \begin{cases} 1 & \text{One of neighbors has 1.} \\ 0 & \text{otherwise} \end{cases}$$

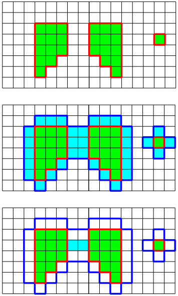
Connect regions

- Contraction

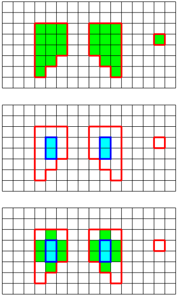
$$g_{i,j} = \begin{cases} 0 & \text{One of neighbors has 0.} \\ 1 & \text{otherwise} \end{cases}$$

Disconnect a region,
Remove isolated point.

Original
⇓
Expansion
⇓
Contraction

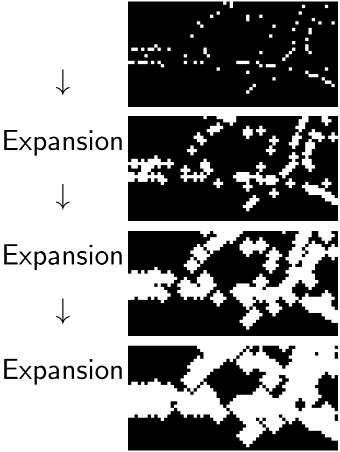


Original
⇓
Contraction
⇓
Expansion



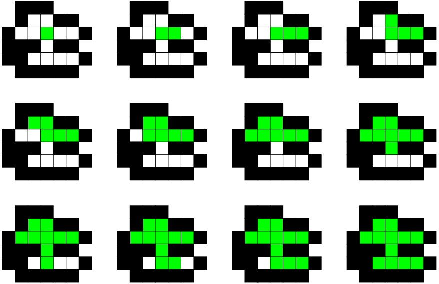
Expansion$\neq$ Inv{Contraction}

Connection of Edges by Expansion



7.9 Painting (Labeling)

Until finding border set mark



begin function `flood_fill`(i, j)
  if flag $L_{i,j}$ is not marked **then**
    set flag $L_{i,j} = D$ as `internal`
    call `flood_fill`$(i+1, j)$
    call `flood_fill`$(i, j+1)$
    call `flood_fill`$(i-1, j)$
    call `flood_fill`$(i, j-1)$
  end if
end function

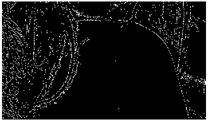
begin main
  set flag $L_{i,j} = B$ for `boundary`
  select initial point (i_0, j_0)
  `flood_fill`(i_0, j_0)
end main

Recursive coding

Example of Painting

Original

Edge detection



Connect broken edge
(Exp edge 3times)

Flood fill

Widen region
(Expand region 3times)

Paint isolated
(Expand region 3times
Contract reg. 3times)

merged
(with detected Edge)

This method requires several try to tune parameters.
Unfortunately, there are no automatic methods.