

# Høypass-filtre

## INF 2310 – Digital bildebehandling

### FILTRERING I BILDE-DOMENET – II

Fritz Albregtsen

Høypass-filtre  
Gradient-operatorer  
Laplace-operatoren  
LoG-operatoren  
Canny-operatoren

G&W Kap. 3.6 + Kap. 10 – 10.2.7

F7 28.02.12

INF 2310

1

- Slipper gjennom høye frekvenser.
- Demper eller fjerner lav-frekvente variasjoner.
- Effekt:
  - Fjerner langsomt varierende bakgrunn
  - Framheve kanter, linjer og skarpe detaljer.

F7 28.02.12

INF 2310

2

# Høypass-filtre

- Et høypass-filter må ha positive vektorer i midten, og negative vektorer lenger ut.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Vi lar summen av vektene være null.
  - Hvorfor er dette lurt?
- Hvis vi lar middelverdien av ut-bildet bli null, må noen deler av ut-bildet være  $<0$ .
- Det er ingen god ide å benytte  $|g(x,y)|$ .
- For framvisning, skalér  $g(x,y)$  og legg til en konstant slik at vi får positive pikselverdier.

F7 28.02.12

INF 2310

3

# Punkt-deteksjon

- Anta at vi benytter en konvolusjonsmaske med vektor  $w_i$  gitt ved:

- Vi beregner  $g(x,y) = \sum_{i=1}^9 w_i f_i$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- der  $f_i$  er den gråtone-verdi vi finner under maske-verdien  $w_i$
- Et punkt  $(i,j)$  avviker fra sine omgivelser dersom

$$|g(x,y)| > T$$

- der  $T > 0$  er en terskel.
- Merk: samme filter som høypass-filtering, men her brukes terskel  $T$  for å finne punkter som avviker tilstrekkelig fra omgivelsene.
  - Hva får vi i homogene områder?
  - Hva med en hellende gråtone-flate?

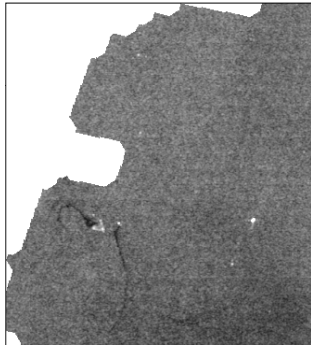
F7 28.02.12

INF 2310

4

## Eksempel: punkt-deteksjon

- Deteksjon av skip i radar-bilder over sjø.



$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 4 & 4 & 4 & -1 & -1 \\ -1 & -1 & 4 & 8 & 4 & -1 & -1 \\ -1 & -1 & 4 & 4 & 4 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Gir størst respons for de små lyse objektene (skipene).

Symmetrisk filter  
=> Vi beregner en korrelasjon

F7 28.02.12

INF 2310

5

## "High-boost" filtrering

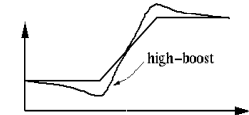
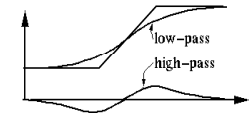
- Høypass-filtrering kan utføres ved  
Høypass = Original - Lavpass

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Adder høypass-resultatet til originalen, og få et "High-boost" bilde

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Legg forskjellige vekter på de to bildene, og få et "veiet high boost" bilde, ved:



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & c & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

F7 28.02.12

INF 2310

6

## "Unsharp masking"

- Middelvefilter => uskarpt bilde.
- Subtraher uskarpt bilde fra originalen.
- Adder differanse til originalen.
- Resultatet er skarpere enn originalen.



F7 28.02.12

INF 2310

7

## Motivasjon for kant-deteksjon

- Det meste av informasjonen i et bilde finnes ved kantene til objektene/regionene i bildet.
- "Kanter" brukes her om intensitets-kanter, farge-kanter, tekstur-kanter etc.
- Biologiske visuelle systemer er basert på kant-deteksjon.
- Slike systemer arbeider ofte både parallelt og sekvensielt:
  - Alle lokale omgivelser behandles uavhengig av hverandre.
  - Lokale resultat kan være avhengig av tidligere resultater.

F7 28.02.12

INF 2310

8

## Digitale gradient-operatorer

- Vi husker at den deriverte av  $f(x)$  er gitt ved

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

- I våre digitale bilder setter vi  $h \geq 1$ .
- Vi får da en gruppe av operator som approksimerer de ortogonale gradient-komponentene  $\delta F(x,y)/\delta x$  og  $\delta F(x,y)/\delta y$
- Noen operatorer gir bare estimat av gradient-magnituden.
- Andre gir også gradient-retningen.
- Gitt to digitale filtermasker  $H_x$  og  $H_y$ 
  - Disse to konvolveres med bildet  $F(i,j)$  og måler gradient komponentene  $g_x$  og  $g_y$  i en omegn om  $(i,j)$  i bildet.

## Gradient i et kontinuerlig bilde

- Gradienten til  $F$  langs  $r$  i retning  $\theta$

$$\frac{\partial F}{\partial r} = \frac{\partial F}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial F}{\partial y} \frac{\partial y}{\partial r}$$

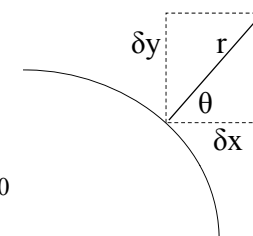
$$\text{dvs. } \frac{\partial F}{\partial r} = \frac{\partial F}{\partial x} \cos \theta + \frac{\partial F}{\partial y} \sin \theta$$

- Gradienten er størst når  $\frac{\partial}{\partial \theta} \left( \frac{\partial F}{\partial r} \right) = 0$

- Dvs. for den vinkelen  $\theta_g$  der

$$-\frac{\partial F}{\partial x} \sin \theta_g + \frac{\partial F}{\partial y} \cos \theta_g = 0 \Leftrightarrow \frac{\partial F}{\partial y} \cos \theta_g = \frac{\partial F}{\partial x} \sin \theta_g$$

- Men  $\delta F/\delta x$  og  $\delta F/\delta y$  er bare de ortogonale gradient-komponentene  $g_x$  og  $g_y$  i hhv. x- og y-retning i bildet.



## Gradient i kontinuerlig bilde - II

- Vi gjentar: Gradienten er størst for den vinkelen  $\theta_g$  der

$$\frac{\partial F}{\partial y} \cos \theta_g = \frac{\partial F}{\partial x} \sin \theta_g \Rightarrow \frac{\sin \theta_g}{\cos \theta_g} = \frac{\partial F / \partial y}{\partial F / \partial x} \Rightarrow \tan \theta_g = \frac{g_y}{g_x}$$

- Derfor: Retningen til kanten relativt til x-aksen er gitt ved

$$\theta_g = \tan^{-1} \left( \frac{g_y}{g_x} \right)$$

- Og gradient-magnituden er gitt ved kvadratroten av summen av kvadratene av de to gradient-komponentene:

$$\left( \frac{\partial F}{\partial r} \right) = [g_x^2 + g_y^2]^{1/2}$$

## Enkel gradient-deteksjon

- Derivasjon kan approksimeres med differanser,

$$\bullet \delta f/\delta x = |f(x+1,y) - f(x-1,y)| \quad \text{i x-retning}$$

$$\bullet \delta f/\delta y = |f(x,y+1) - f(x,y-1)| \quad \text{i y-retning}$$

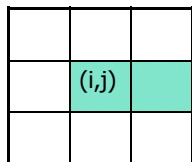
- Disse operatorene er følsomme for bilde-støy.
  - Vi finner noen falske kanter, ikke relatert til objekter.
- Fjerne støy først, og så gjøre kantdeteksjon ?
  - Eller kan vi gjøre det enklere, og bedre?

## Digitale gradient-approksimasjoner - I

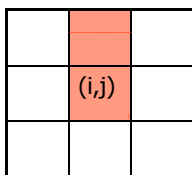
- Asymmetrisk 1D operator:

$$h_R(i,j) = f(i,j) - f(i,j-1)$$

$$h_C(i,j) = f(i,j) - f(i+1,j)$$



- Definisjonene er gitt slik at komponentene er positive for en kant der intensiteten øker fra venstre mot høyre og oppover i bildet.



- Problem:

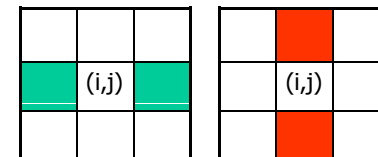
- Gradient-estimatet refererer til et punkt midt mellom to piksler
- X- og Y-estimatet refererer ikke til samme sted i bildet.

## Digitale gradient-approksimasjoner - II

- Symmetrisk 1D operator:

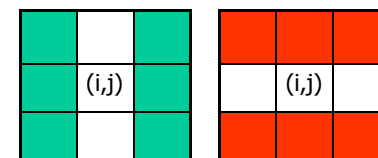
$$h_R(i,j) = f(i,j+1) - f(i,j-1)$$

$$h_C(i,j) = f(i-1,j) - f(i+1,j)$$



- Gradient-estimatene refererer nå til (i,j).
- Problem: Operatoren er veldig følsom for støy.
- Løsning: Gjør det samme for tre symmetriske par

- Gradient-estimatene blir mer robuste mot støy i bildet.



## Noen gradient-operatorer - I

- "Pixel difference"

$$H_x(i,j) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad H_y(i,j) = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- "Separated pixel difference"

$$H_x(i,j) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad H_y(i,j) = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

- Roberts-operatoren

$$H_x(i,j) = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad H_y(i,j) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

## Noen gradient-operatorer - II

- Prewitt-operatoren

$$H_x(i,j) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad H_y(i,j) = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Sobel-operatoren

$$H_x(i,j) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad H_y(i,j) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Frei-Chen-operatoren

$$H_x(i,j) = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \quad H_y(i,j) = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

## Symmetriske gradient-operatorer

- Operatoren gjøres mindre følsom for støy ved å midle i en retning og derivere i den ortogonale retningen.
 
$$\begin{matrix} \text{Prewitt} \\ \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \end{matrix}$$
- Eksempler: **Prewitt, Sobel, Frei-Chen**
- Prewitt er mer følsom for horisontale og vertikale enn for diagonale kanter.
 
$$\begin{matrix} \text{Sobel} \\ \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \end{matrix}$$
- Det motsatte er tilfelle for Sobel.
- Frei-Chen gir samme gradient om kanten ligger langs aksene eller diagonalt.
 
$$\begin{matrix} \text{Frei-Chen} \\ \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \end{matrix}$$
- Alle de tre nevnte operatene er separable.

## Separasjon av gradient-filtre

- Separasjon av Prewitt-operatoren:
 
$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = [1 \ 0 \ -1] * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \ 1 \ 1]$$
- Separasjon av Sobel-operatoren:
 
$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = [1 \ 0 \ -1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \ 2 \ 1]$$
- Separasjon av Frei-Chen:
 
$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} = [1 \ 0 \ -1] * \begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \ \sqrt{2} \ 1]$$

## $g_x$ , $g_y$ og gradient-magnituden, $G$

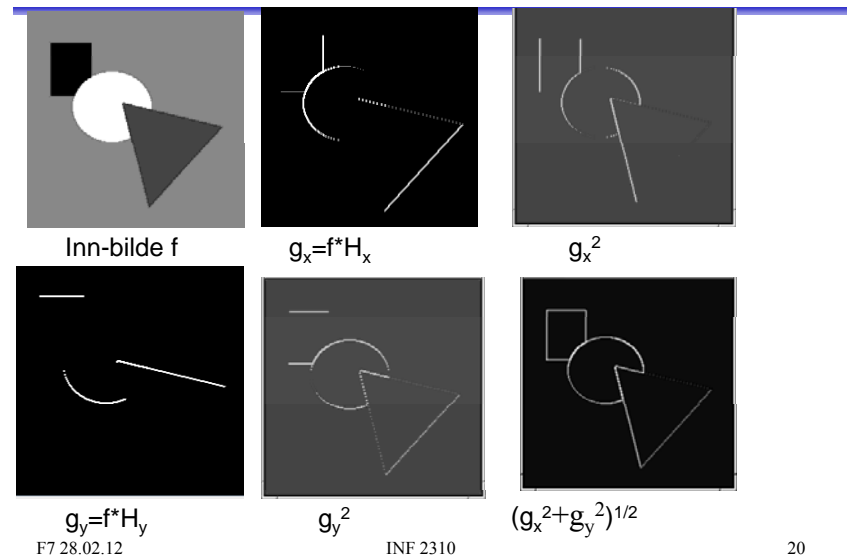
- Vi finner de horisontale kantene:
  - Beregn  $g_x(x,y) = H_x * f(x,y)$
- Vi finner de vertikale kantene:
  - Beregn  $g_y(x,y) = H_y * f(x,y)$
- Beregn gradient-magnitude og retning:

$$G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \quad \text{Gradient-magnitude}$$

$$\theta(x, y) = \tan^{-1} \left( \frac{g_y(x, y)}{g_x(x, y)} \right) \quad \text{Gradient retning}$$

## Eksempel - Sobel

Vær obs på skalering av  $g_x$  og  $g_y$  i denne figuren



## Implementasjoner av en gradient-operator

- Større filtre gir mindre følsomhet for støy.
- Sobel (og de andre operatorene) kan utvides til større enn 3×3.
- Her er en 5×5 Sobel operator:

$$h_x = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Krever 50  
multiplikasjoner

- Denne kan implementeres som

$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Krever 36  
multiplikasjoner

## Implementasjoner av en gradient-operator II

- De to 3×3 filtrene er separable, og kan skrives

$$h_x = [1 \ 0 \ -1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * [1 \ 2 \ 1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad h_y = [1 \ 2 \ 1] * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \ 2 \ 1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Krever 24  
multiplikasjoner

- Eller vi kan skrive

$$h_x = [1 \ 1] * [1 \ -1] * \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} * [1 \ 1] * [1 \ 1] * \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Krever 32  
multiplikasjoner

$$h_y = [1 \ 1] * [1 \ 1] * \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 \\ 1 \end{bmatrix} * [1 \ 1] * [1 \ 1] * \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- Raskeste blir

$$h_x = [1 \ 2 \ 0 \ -2 \ -1] * \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad h_y = [1 \ 4 \ 6 \ 4 \ 1] * \begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix}$$

Krever bare 20  
multiplikasjoner

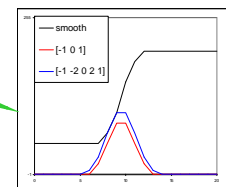
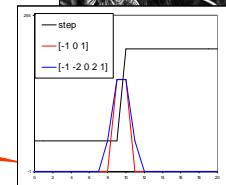
## Om bruk av gradient-operatorer

- Kantene blir "brede"
  - Hvor bred kanten blir avhenger av filterkjernen
- Hvordan skal vi finne eksakt hvor kanten går?
  - Terskle kantbildet?
  - Tynne kantbildet?
  - Bruke 2. deriverte?
- Følg med, følg med!



## Gradient-operatorer

- Gradient-operatorer gir en "bred respons"
  - Hvor bred responsen blir vil avhenge av både kant-profilen og filterstørrelsen.
- En "step edge":
  - Bredden på responsen er avhengig av størrelsen på filteret.
- En bred kant (glattet med [1 2 3 2 1]):
  - Bredden på responsen er avhengig av bredden på kanten.
- Finn eksakt hvor kanten går:
  - Finn maksimum i derivert?
  - Bruke 2. derivert?



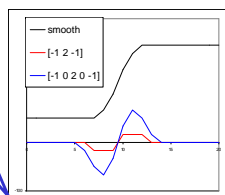
# Laplace-operatoren



- Laplace-operatoren er gitt ved:

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

- Den endrer fortegn der  $f(x,y)$  har et infleksjonspunkt / vendepunkt.
- $|\nabla^2 f|$  har to ekstremverdier idet vi passerer en kant
- $\nabla^2 f = 0$  markerer kant-posisjon.



- Kantens eksakte posisjon finnes ved nullgjennomgangen.
- Dette gir ikke brede kanter.
- Vi finner bare magnitudo, ikke retning.

# 1D Laplace-operator

**Kontinuerlig**   **Digitalt**

$f(x)$

$f(i)$

$f'(x)$

$f(i) - f(i-1)$

$f''(x) = \nabla^2 f$

$$\begin{aligned} f'(i) - f'(i-1) \\ = [f(i) - f(i-1)] \\ - [f(i-1) - f(i-2)] \\ = f(i-2) - 2f(i-1) + f(i) \end{aligned}$$

- I 1D er  $\nabla^2 f$  ekvivalent med 2. deriverte
- Av symmetri-hensyn flytter vi operatoren slik at den er sentrert om  $i$ .
- Dessuten bytter vi fortegn:  
 $\nabla^2 f = -f(i-1) + 2f(i) - f(i+1)$

# 2D Laplace-operator

- Anvend 1D Laplace i begge retninger:

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

$$\begin{aligned} \approx -f(i-1, j) + 2f(i, j) - f(i+1, j) \\ - f(i, j-1) + 2f(i, j) - f(i, j+1) \end{aligned}$$

- Dette oppnår vi ved å konvolvare  $f(i,j)$  med

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

# Flere Laplace-operatorer

- Merk at Laplace-operatorene kan uttrykkes som senter-verdi minus et (veiet) middel over et lokalt nabolik.
- 1D  $\nabla^2 f(i) = -f(i-1, j) + 2f(i, j) - f(i+1, j) = 3f(i) - \sum_{j=i-1}^{i+1} f(j)$

• 2D "pluss"  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

• 2D "kvadrat"  $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$



# Laplace på et 2.grads polynom

- La intensiteten omkring  $(x,y)$  være gitt ved polynomet  $f(x,y) = k_1+k_2x+k_3y+k_4x^2+k_5xy+k_6y^2$
- I et 3x3 område rundt  $(x,y)$  har vi da intensitetene

$k_1-k_2-k_3+k_4+k_5+k_6$	$k_1-k_3+k_6$	$k_1+k_2-k_3+k_4-k_5+k_6$
$k_1-k_2+k_4$	$k_1$	$k_1+k_2+k_4$
$k_1-k_2+k_3+k_4-k_5+k_6$	$k_1+k_3+k_6$	$k_1+k_2+k_3+k_4+k_5+k_6$

- Den korrekte Laplace-verdien er gitt ved

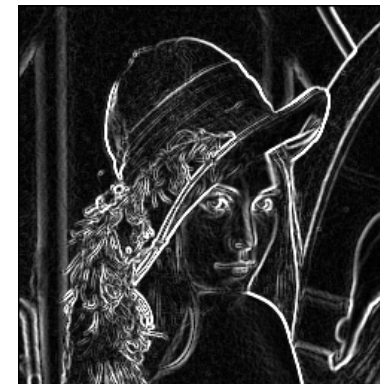
$$-\nabla^2(f(x,y)) = -\left(\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right) = -2(k_4+k_6)$$

- Både 4-nabo og 8-nabo masken gir korrekt estimat!

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

$$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

# Laplace vs. Sobel



Sobel-filtrert  
=> bred kant



Laplace-filtrert  
=> dobbelt-kant

# "Edges & step-edge lines"

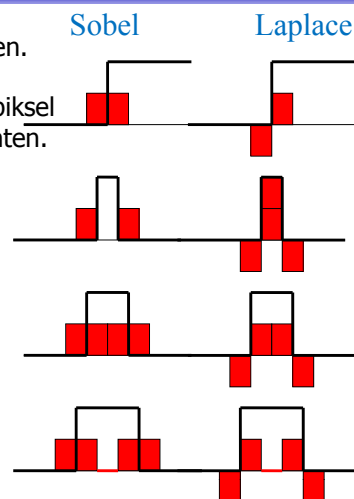
- Kantpixel = 1. pixel innenfor kanten.

- Sobel gir samme respons på første pixel utenfor og første pixel innenfor kanten.

- Laplace gir motsatte responser, og riktig null-gjennomgang.

- En linje har to kanter.

- 1-pixels linje gir dobbel respons – utenfor linja.



# Implementasjoner av en 5x5 Laplace-operator

- Laplace-operatoren

$$\nabla^2 = \begin{bmatrix} 2 & 4 & 4 & 4 & 2 \\ 4 & 0 & -8 & 0 & 4 \\ 4 & -8 & -24 & -8 & 4 \\ 4 & 0 & -8 & 0 & 4 \\ 2 & 4 & 4 & 4 & 2 \end{bmatrix}$$

Krever 2x25 operasjoner

er ikke separabel.

- Men den kan skrives som

$$\nabla^2 = -\left( \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \right)$$

Vi ser at

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

kan finnes ved 2 ganger derivasjon med Sobel – filtrene

som igjen kan skrives som

$$\nabla^2 = -\left( \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 4 & 0 & -8 \\ 1 & 0 & -1 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \right)$$

- Separasjon gir den raskeste implementasjonen: Krever 2x20 operasjoner

$$\nabla^2 = -\left( \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}^T + \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}^T \right)$$



## Fra Laplace til LoG

- Vi gjorde gradient-operatorene støy-robuste
  - ved å bygge inn en lavpassfiltrering. Eksempel: Sobel-operator

$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = [1 \ 0 \ -1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \ 2 \ 1]$$

- Vi kan gjøre det samme med Laplace-operatoren
  - Vi bruker et Gauss-filter G
  - Og siden konvolusjon er kommutativ, får vi

$$\nabla^2 * (f * G) = (\nabla^2 * G) * f = LoG * f$$

- Der "LoG" er resultatet av å anvende Laplace-operatoren på en Gauss-funksjon.

## LoG = -∇<sup>2</sup>G

Gauss – funksjonen i 2D:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Deriverer mhp x:

$$\frac{\partial G}{\partial x} = -\frac{1}{2\pi\sigma^4} \left[ x e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right]$$

Deriverer mhp y:

$$\frac{\partial G}{\partial y} = -\frac{1}{2\pi\sigma^4} \left[ y e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right]$$

2. derivert mht x:

$$\frac{\partial^2 G}{\partial x^2} = -\frac{1}{2\pi\sigma^4} \left( 1 - \frac{x^2}{\sigma^2} \right) \left[ e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right]$$

2. derivert mht y:

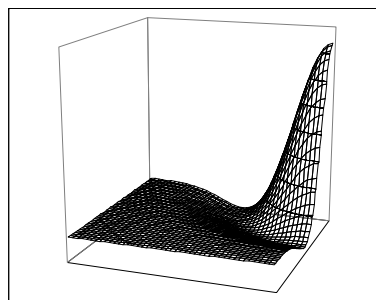
$$\frac{\partial^2 G}{\partial y^2} = -\frac{1}{2\pi\sigma^4} \left( 1 - \frac{y^2}{\sigma^2} \right) \left[ e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right]$$

Summen av disse gir oss

$$-\nabla^2 G = \frac{1}{2\pi\sigma^4} \left( 2 - \frac{x^2 + y^2}{\sigma^2} \right) \left[ e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right]$$

## Laplacian-of-Gaussian (LoG)

$$-\nabla^2 G = \frac{1}{2\pi\sigma^4} \left( 2 - \frac{x^2 + y^2}{\sigma^2} \right) \left[ e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right]$$



- $\sigma$  er standard-avviket til  $G(x,y)$
- $w = 2\sqrt{2} \sigma$  er bredden av den positive toppen til LoG-operatoren.
- I de fleste tilfeller er størrelsen av operatoren  $\approx 3w \approx 8.5\sigma$
- Da er  $\sum (LoG) \approx 0$

En kvadrant av et LoG-filter

## Et 11x11 LoG-filter

Laplace - operatoren

$$-\nabla^2 G = \frac{1}{2\pi\sigma^4} \left( 2 - \frac{x^2 + y^2}{\sigma^2} \right) \left[ e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right] \text{ kan implementeres som en sampling av}$$

$$A \left( 1 - k \frac{x^2 + y^2}{\sigma^2} \right) \left[ e^{-\frac{(x^2+y^2)}{2\sigma^2}} \right] \text{ ved heltalls verdier av } x \text{ og } y.$$

Bruk en verdi av  $k$  slik at

summen innenfor masken blir 0.

A skaleres verdiene

slik at vi kan bruke heltall.

$$LoG_{11 \times 11} = \begin{bmatrix} -24 & -21 & -17 & -13 & -10 & -10 & -10 & -13 & -17 & -21 & -24 \\ -21 & -16 & -10 & -4 & 0 & 2 & 0 & -4 & -10 & -16 & -21 \\ -17 & -10 & -1 & 6 & 11 & 13 & 11 & 6 & -1 & -10 & -17 \\ -13 & -4 & 6 & 15 & 22 & 24 & 22 & 15 & 6 & -4 & -13 \\ -10 & 0 & 11 & 22 & 29 & 31 & 29 & 22 & 11 & 0 & -10 \\ -10 & 2 & 13 & 24 & 31 & 34 & 31 & 24 & 13 & 2 & -10 \\ -10 & 0 & 11 & 22 & 29 & 31 & 29 & 22 & 11 & 0 & -10 \\ -13 & -4 & 6 & 15 & 22 & 24 & 22 & 15 & 6 & -4 & -13 \\ -17 & -10 & -1 & 6 & 11 & 13 & 11 & 6 & -1 & -10 & -17 \\ -21 & -16 & -10 & -4 & 0 & 3 & 0 & -4 & -10 & -16 & -21 \\ -24 & -21 & -17 & -13 & -10 & -10 & -10 & -13 & -17 & -21 & -24 \end{bmatrix}$$

## 7x7 LoG-filter

$$\text{LoG}_{7 \times 7} = -\nabla_{3 \times 3}^2 * G_{3 \times 3} = \begin{bmatrix} -2 & -8 & -14 & -16 & -14 & -8 & -2 \\ -8 & -24 & -24 & -16 & -24 & -24 & -8 \\ -14 & -24 & 30 & 80 & 30 & -24 & -14 \\ -16 & -24 & 80 & 160 & 80 & -24 & -16 \\ -14 & -24 & 30 & 80 & 30 & -24 & -14 \\ -8 & -24 & -24 & -16 & -24 & -24 & -8 \\ -2 & -8 & -14 & -16 & -14 & -8 & -2 \end{bmatrix}$$

- Dette kan splittes opp i

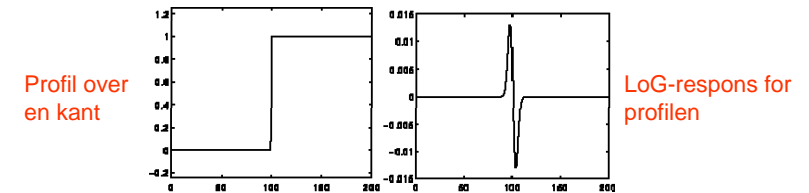
$$-\nabla_{3 \times 3}^2 = [1 \ 4 \ 6 \ 4 \ 1] * [1 \ 0 \ -2 \ 0 \ 1]^T + [1 \ 4 \ 6 \ 4 \ 1]^T * [1 \ 0 \ -2 \ 0 \ 1]$$

og

$$G_{3 \times 3} = [1 \ 2 \ 1] * [1 \ 2 \ 1]^T$$

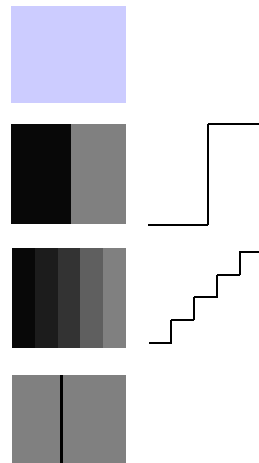
## Bruk av LoG-operator

- Laplace-operatoren detekterer kanter, men også støy i bildet.
- Ofte må man glatte støy før Laplace-utføres.
- LoG-operatoren gjør begge disse operasjonene i ett.
- I homogene områder vil LoG-operatoren gi respons 0.
- Den vil ha et positivt utslag på den ene siden av kanten, 0 i selve kanten, og et negativt utslag på den andre siden.

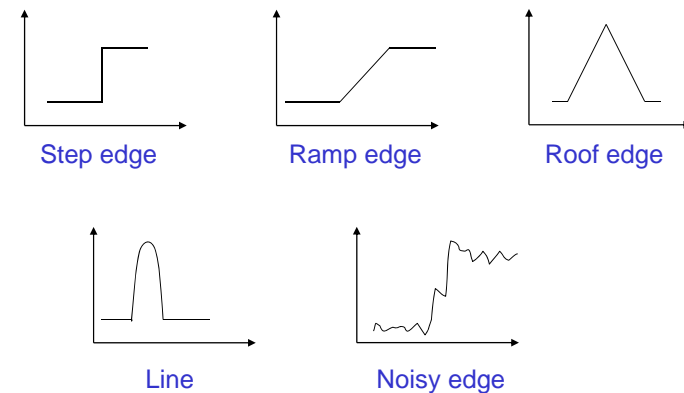


## Flater, kanter og linjer

- En **homogen flate** er et område der alle pikselverdiene er omtrent like.
- En **kant** definerer vi som overgangen mellom to områder med forskjellig middelvei.
  - Kanten er første piksel innenfor overgangen.
- En **rampe** er et område der gradienten er konstant.
  - Kanten er første piksel innenfor midtpunktet på rampen.
- Merk at en **linje** består av **to** kanter.

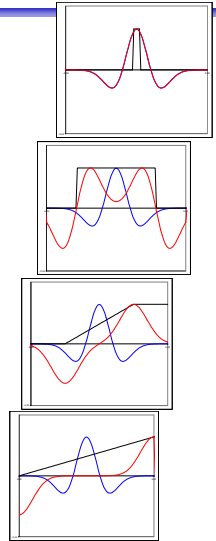


## Kant-typer



## Legg merke til at ...

- En struktur som er smalere enn LoG-kjernen, gir to nullgjennomganger.
- Kjernen bestemmer avstanden mellom dem.
- Hvis strukturen er bredere enn kjernen, men smalere enn filteret, blir kantene riktig detektert.
- På en rampe som er bredere enn kjernen, men smalere enn filteret, finner LoG en nullgjennomgang midt på.
- På ramper som er bredere enn filteret, finner ikke LoG noen nullgjennomgang, bare et null-platå.



## Robust kant-deteksjon

Vanligvis tre steg i robust kant-deteksjon:

1. **Støy-reduksjon**, der vi forsøker å fjerne så mye støy som mulig uten å smøre ut kanten (lavpassfilter).
2. **Kant-filtrering**, der vi anvender høypass-filtre, gradient-operatorer e.l.
3. **Kant-lokalisering**, der vi etterbehandler resultatet fra kant-filtreringen for å finne eksakte kant-posisjoner.

## Hva er en god kant-detektor?

- Bare de relevante kantene bør finnes.
- Posisjonen til detektert kant bør samsvare med der kanten faktisk finnes i bildet.
- En kant skal kun gi en enkelt respons.
- Den bør være robust for litt støy.

## Canny's kant-detektor

- En "optimal" kant-detektor som gir oss et best mulig kompromiss mellom støyreduksjon og kant-lokalisering.
- Den er optimal i forhold til tre kriterier:
  - God kant-lokalisering
  - En enkelt respons
  - Best mulig deteksjon

## Canny's algoritme

---

- Lavpass-filtrering med Gauss-funksjon (gitt  $\sigma$ ).
- Finn gradient-magnitude og retning.
- Tynning av gradient-magnitude ortogonalt på kant:
  - Hvis et piksel har en nabo med høyere pikselverdi, settes pikselverdien ned.
- Hysteresetterskling (to terskler  $T_h$  og  $T_l$ ) :
  1. Merk alle piksler der  $G > T_h$
  2. Scan alle piksler der  $G \in [T_l, T_h]$
  3. Hvis et slikt piksel er nabo til et merket piksel, så merkes dette pikselet også.
  4. Gjenta fra trinn 2 til konvergens.

## Oppsummering

---

- Vi har utledet enkle kant-deteksjonsoperatorer.
- Gradient-operatorene glatter i den ene retningen og gjør kantdeteksjon i den andre retningen.
- Gradient-operatorer gir både kant-styrke og retning.
- Laplace-operatoren gir presis lokalisering av kanten, men forsterker støy.
- LoG-operatoren er en mer robust versjon av Laplace som inkluderer glatting.
- Filterets størrelse må passe til oppgaven!
- Canny gir et kompromiss mellom støyreduksjon og kantlokalisering.