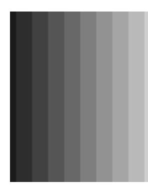
INF 2310 – Digital bildebehandling

FILTRERING I BILDE-DOMENET - II

Gradient-detektorer
Laplace-operatorer
LoG-operatoren
Canny-operatoren
GW Kap. 3.6 + Kap 10-10.2.7

Forrige uke: lavpass-filtere

- Slipper gjennom lave frekvenser.
- Glatter bildet og fjerner skarpe kanter/overganger.
- Konvolusjonsfiltere:
  - Middelverdifiltere
  - Gaussisk lavpass.



Original

5x5 middelverdi

Høypass-filtere

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

Høypass-filtre

- Et høypass-filter må ha positive vekter i midten, og negative vekter lenger ut. Et eksempel:

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

- 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, skaler $g(x,y)$ og legg til en konstant slik at vi får positive pikselverdier.

En liten test på konvolusjon

- Bildet:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}$$

Hva blir resultatet når filterkjernen
Plasseres rundt piksel:
1 (rødt)
2 (blått)
3 (grønt)

- Filter:

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

Punkt-deteksjon

- Anta at vi benytter en konvolusjonsmaske med vekter w_i gitt ved:
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
- Vi beregner $g(x,y) = \sum_{i=1}^{9} w_i f_i$

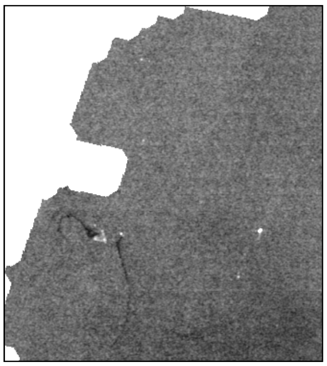
  der f_i er den gråtone-verdi vi finner under maske-verdien w_i.
- Et punkt (i,j) avviker fra sine omgivelser dersom

$$|g(i,j)| > T$$

  der T>0 er en terskel.
- Merk: samme maske som høypass-filtering, men nå bruker vi terskel T for å finne punkter som avviker tilstrekkelig fra omgivelsene.
- Hva får vi i homogene områder?
- Hva med en hellende gråtone-flate (for eksempel gradvis økende gråtone fra venstre mot høyre i bildet)?

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 ->
Det vi gjør kan ses på som en korrelasjon

”High-boost”-filtre

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

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

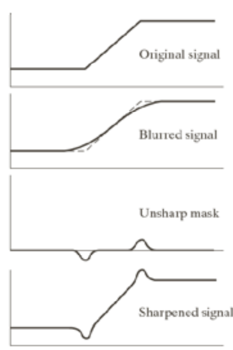
- Hvis vi adderer et høypass-bilde til original-bildet, får vi et ”High-boost” bilde

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

- Vekter vi de to bildene, får vi generelt

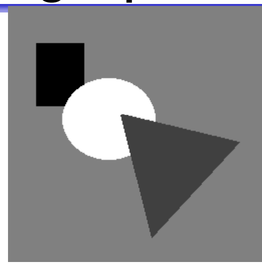
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & c & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Dette gjøres i øyet vårt, og i diverse elektronikk.

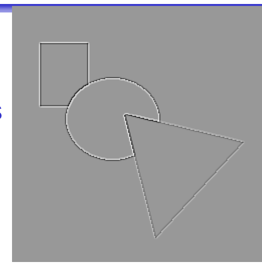


High-pass vs. high boost

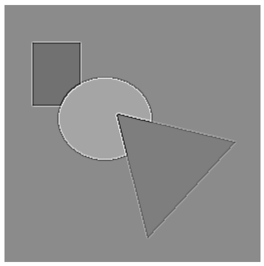
Original



High pass



High boost
c=9



High boost
c=30

Motivasjon for kant-deteksjon

- Repeterte fenomener i en sekvens bringer oss lite informasjon.
- Vi vet at dette utnyttes i kommunikasjon og i bilde-kompresjon.
- 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, ikke på f.eks. terskling.
- Slike systemer arbeider ofte både parallelt og sekvensielt:
  - Alle lokale omgivelser behandles uavhengig av hverandre (i parallell).
  - Lokale resultat kan være avhengig av tidligere resultater.

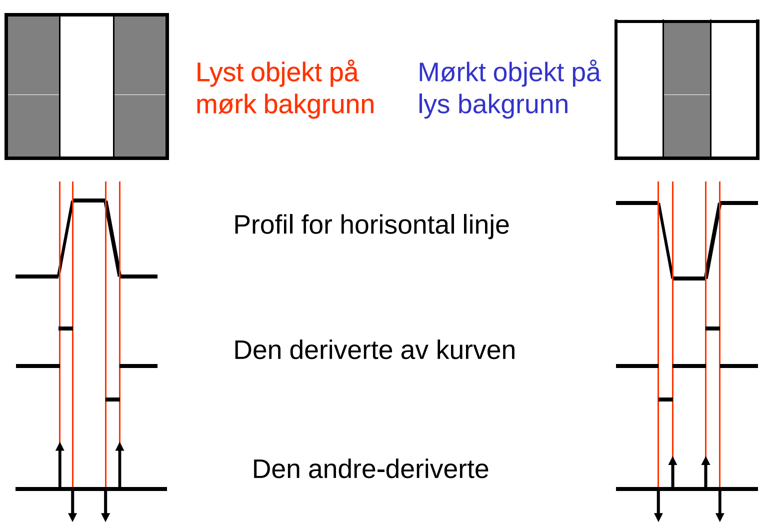
Digitale gradient-operatorer

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

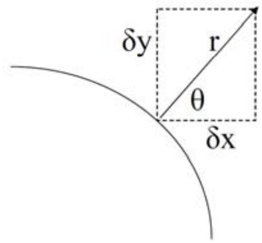
$$\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

- I våre digitale bilder setter vi h≥1.
- Vi får en ide om at pikseldifferanser i en retning kan si noe om gradienten. Mer om dette straks.
- 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 (kant-styrken).
- Andre gir også gradient-retningen.
- Gitt to digitale masker 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.

Illustrasjon av kant-deteksjon



Gradient i kontinuerlig bilde



- Gradienten til *F* langs *r* i retning θ

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

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 θ_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 δ*F*/δ*x* og δ*F*/δ*x* er bare de ortogonale gradient-komponentene i x- og y-retning i bildet.

Gradient i kontinuerlig bilde - II

- Vi gjentar: Gradienten er størst for den vinkelen θ_g der

$$\frac{\partial F}{\partial y}\cos\theta_g = \frac{\partial F}{\partial x}\sin\theta_g$$

- Dvs. når $$\frac{g_y}{g_x} = \frac{\sin\theta_g}{\cos\theta_g} = \tan\theta_g$$
- Derfor: Retningen til kanten relativt til x-aksen er gitt ved

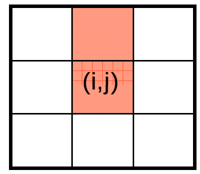
$$\boxed{\theta_g = \tan^{-1}\left(\frac{g_y}{g_x}\right)}$$

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

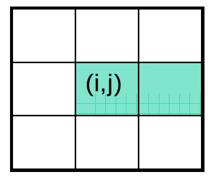
$$\boxed{\left(\frac{\partial F}{\partial r}\right)_{\max} = \left[g_x^2 + g_y^2\right]^{1/2}}$$

Digitale gradient-approksimasjoner

- Asymmetrisk 1D-operator:
  Kolonne: $G_y(x,y)=F(x,y)-F(x,y-1)$
  Rad: $G_x(x,y)=F(x,y)-F(x+1,y)$

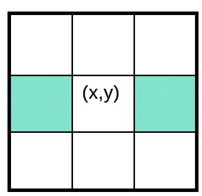


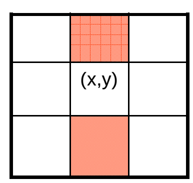
- Definisjonene er gitt slik at komponentene er positive for en kant der intensiteten øker fra venstre mot høyre og nedenfra og oppover i bildet.



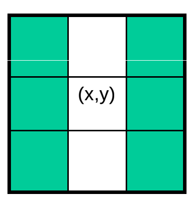
Digitale gradient-approksimasjoner

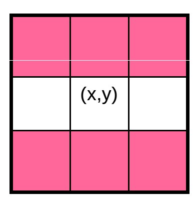
- Symmetrisk 1D-operator:
  $G_y(x,y)=F(x,y+1)-F(x,y-1)$
  $G_x(i,j)=F(x-1,y)-F(x+1,y)$





- Gradient-estimatene refererer nå til punktet (x,y).
- Problem: operatoren er veldig følsom for støy.
- Løsning: lavpassfilter i motsatt retning.





Gradient-operatorer

- ”Pixel difference”

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

- ”Separated pixel difference”

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

- Roberts-operatoren

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

Gradient-operatorer

- Prewitt-operatoren

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

- Sobel-operatoren

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

- Frei-Chen-operatoren

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

Forskjellen på G_x, G_y og gradient-operatoren

- For å finne horisontale kanter:
  - Beregn delresultatet $g_x(x,y)=H_x*f(x,y)$
  - Dvs. konvolver bildet med den horisontale filterkjernen H_x
- For å finne vertikale kanter:
  - Beregn delresultatet $g_y(x,y)=H_y*f(x,y)$
  - Dvs. konvolver bildet med den vertikale filterkjernen H_y
- Beregn så gradientoperatoren ved:

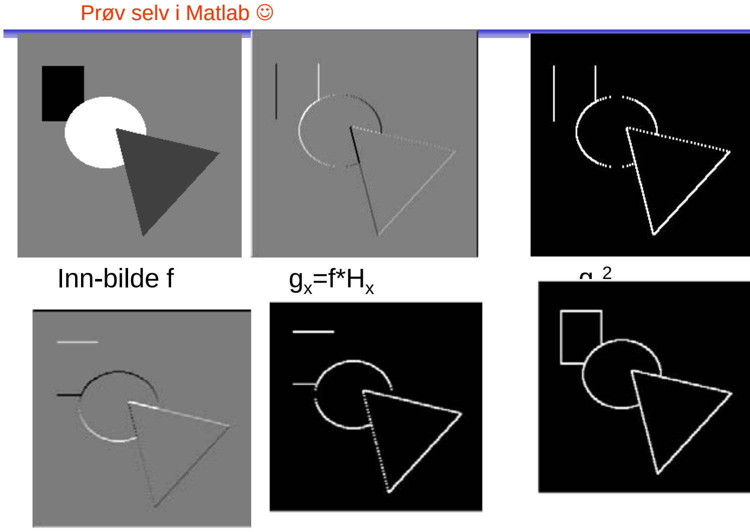
$$g(x,y)=\sqrt{g_x^2(x,y)+g_y^2(x,y)}$$ Gradient-magnitude (kant-styrke)

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

- Alternativt kan absoluttverdien brukes: $g(x,y)=|g_x(x,y)|+|g_y(x,y)|$
- Dette gjelder for alle operatorene (Sobel, Prewitt, Frei-Chen etc..)

Eksempel - Sobel

Vær obs på skalering av gx og gy i denne figuren (feks negative verdier)
Prøv selv i Matlab ☺



Inn-bilde f $g_x=f*H_x$ g_x^2

$g_y=f*H_y$ g_y^2 $(g_x^2+g_y^2)^{1/2}$

Symmetriske gradient-operatorer

- Operatoren gjøres mindre følsom for støy ved å midle i en retning og derivere i den ortogonale retningen.
- Eksempler: Prewitt, Sobel, Frei-Chen
- Frei-Chen gir samme gradient om kanten ligger langs aksene eller diagonalt.
- Prewitt er mer følsom for horisontale og vertikale enn for diagonale kanter.
- Det motsatte er tilfelle for Sobel.
- Prewitt-operatoren glatter ikke ut støy like effektivt som Sobel-operatoren.
- Større masker gir mindre støy-følsomhet.
- Alle de tre nevnte operatene er separable.

Separasjon av gradient-filtre

- Separasjon av Prewitt-operatoren:

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

- Separasjon av Sobel-operatoren:

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

- Separasjon av Frei-Chen:

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

Implementasjon 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

Mer om implementasjon

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

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

Krever 24 multiplikasjoner

- Eller vi kan skrive

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

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

Krever 32 multiplikasjoner

- Raskeste blir

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

Krever bare 20 multiplikasjoner

Om bruk av gradient-operatorer

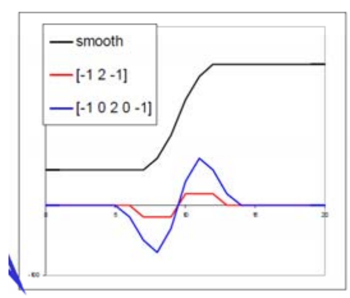
- Kantene blir ”brede”
  - Hvor bred avhenger av filterkjernen
- Hvordan skal vi finne eksakt hvor kanten går?
  - Terskle kantbildet?
  - Finn maksimum i derivert?
    - Bruke 2. derivert?

Laplace-operatoren

- Laplace-operatoren er definert 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 infleksjons-punkt/vendepunkt.
- Den gir ikke brede kanter.
- Vi finner bare magnitude, ikke retning.
- Merk at $|\nabla^2 f|$ har to ekstremverdier idet vi passerer en kant, mens $\nabla^2 f=0$ markerer nøyaktig kant-posisjon.
- Kantens eksakte posisjon finnes altså ved nullgjennomgangen.



1D Laplace-operator

- 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:
  $\nabla^2 f = -f(i-1)+2f(i)-f(i+1)$

Kontinuerlig	Digitalt
$f(x)$	$f(i)$
$f'(x)$	$f(i)-f(i-1)$
$f''(x)=\nabla^2 f$	$f'(i)-f'(i-1)=$ $[f(i)-f(i-1)]$ $-[f(i-1)-f(i-2)]$ $=f(i-2)-2f(i-1)+f(i)$

2D Laplace operator

- Anvender 1D Laplace i begge retninger og summerer:

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

- Dette får vi ved å konvolvere *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 naboskap.
- 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” $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
- 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 vs. Sobel

Sobel-filtrert

Laplace-filtrert

Implementasjon 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 2×25 operasjoner

er ikke separabel.

- Men den kan skrives som

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

som igjen kan skrives som

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

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

- Separasjon gir den raskeste implementasjonen: Krever 2×20 operasjoner

$$\boxed{\nabla^2 = -\left([1 \;\; 0 \;\; -2 \;\; 0 \;\; 1] * [1 \;\; 4 \;\; 6 \;\; 4 \;\; 1]^T + [1 \;\; 4 \;\; 6 \;\; 4 \;\; 1] * [1 \;\; 0 \;\; -2 \;\; 0 \;\; 1]^T \right)}$$

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=-∇^2G

- Gauss-funksjonen i 2D er gitt som:

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

- Vi deriverer denne:

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

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

Og tilsvarende får vi:

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

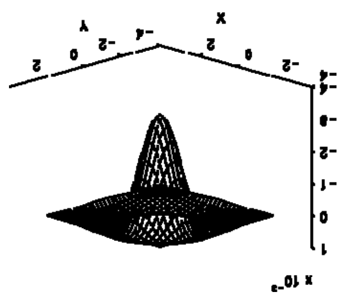
og summen av disse gir oss

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

Laplacian-of-Gaussian (LoG)

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

- σ er standard-avviket til G(x,y)
- w=2√2 σ er bredden av den positive toppen til LoG-operatoren.
- I de fleste tilfeller er størrelsen av operatoren ≈3w ≈8.5σ

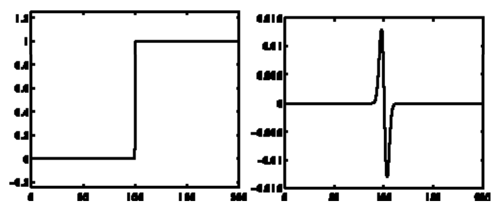


LoG

Bruk av LoG

- Laplace-operatoren (uten Gauss-en) 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.

Profil over en kant



LoG-respons for profilen

Et 7x7 LoG-filter

$$LoG_{7\times 7} = -\nabla^2_{5\times 5} * G_{3\times 3} = \begin{bmatrix} -2 & -8 & -14 & -16 & -14 & -8 & -2 \\ -8 & -24 & -24 & -16 & -24 & -24 & -8 \\ -14 & -24 & 30 & 80 & 30 & -24 & -14 \\ -16 & -16 & 80 & 160 & 80 & -16 & -16 \\ -14 & -24 & 30 & 80 & 30 & -24 & -14 \\ -8 & -24 & -24 & -16 & -24 & -24 & -8 \\ -2 & -8 & -14 & -16 & -14 & -8 & -2 \end{bmatrix}$$

- Dette kan splittes opp i

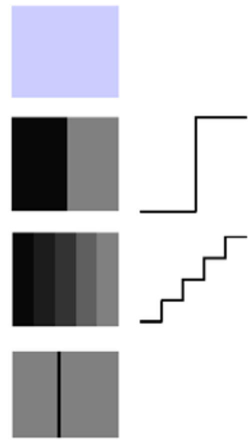
$$-\nabla^2_{5\times 5} = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}^T * \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix}$$

og

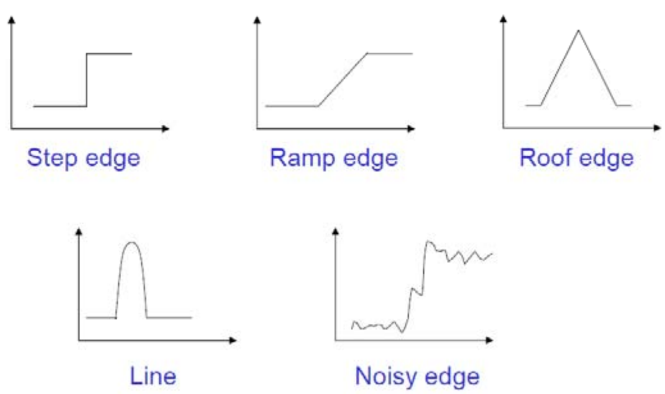
$$G_{3\times 3} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T$$

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 middelverdi.
  - 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.

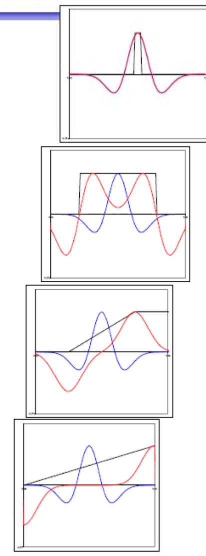


Kant-typer



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.



Oppsummering

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