

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Eksamen i : INF2310 — Digital bildebehandling

Eksamensdag : Tirsdag 19. mars 2013

Tid for eksamen : 15:00 – 19:00

Løsningsforslaget er på : **11 sider**

Vedlegg : **Ingen**

Tillatte hjelpemidler: **Ingen**

- Det er 7 oppgaver i dette oppgavesettet.
- Les gjennom hele oppgavesettet før du begynner å løse oppgavene. Kontroller at oppgavesettet er komplett før du begynner å besvare det. Dersom du savner opplysninger i en oppgave, kan du selv legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i såfall rede for forutsetningene og antagelsene du gjør.
- Det er tilsammen 21 delspørsmål, og det lønner seg å disponere tiden slik at man får besvart alle oppgavene. Hvis du står fast på enkeltoppgaver, gå videre slik at du får gitt et kort svar på alle oppgaver.
- **Alle svar skal begrunnes.** Gjør rede for bruken av eventuelle teoremer, prinsipper eller forutsetninger slik at en tredjeperson kan følge dine resonnerer.

## 1. Sampling og kvantisering

- a) Anta at et avbildningssystem gir en punktspredningsfunksjon der avstanden fra maksimum til første minimum er  $1/3 \mu\text{m}$  i bildeplanet. Vi skal altså kunne skille punktkilder som ligger  $1/3 \mu\text{m}$  fra hverandre i det analoge bildet. Hva er den minste samplingsraten (frekvensen) vi kan benytte ved digitaliseringen av dette bildet i følge samplingsteoremet, og hvor store kan detektorene være? Vær presis med benevningene!

Svar: Samplingsteoremet krever  $T_s < 1/2 T$ , der  $T$  her er  $1/3 \mu\text{m}$ .  
Følgelig må samplingsfrekvensen være  $f_s > 2 * 1/T = 6 \mu\text{m}^{-1}$ .  
Med en samplingsfrekvens på 6 detektorer per  $\mu\text{m}$ ,  
kan hver detektor maksimalt være  $1/6 \mu\text{m} = 0,167 \mu\text{m}$  hvis de ligger kant i kant.

- b) Hva mener vi med begrepene *aliasing*, *aliasing-frekvens* og *anti-aliasing*?

Svar: Romlig aliasing er en frekvensforvrengning som oppstår når man sampler med en lavere samplingsrate enn Nyquist-raten, dvs 2 ganger den høyeste romlige frekvensen som finnes i et bånd-begrenset bilde.

En aliasing-frekvens er en frekvens  $f_a$  som oppstår eller styrkes i det samlede bildet p.g.a. aliasing og er gitt ved  $f_a = f_s - f$ , når  $f < f_s < 2f$ , der  $f_s$  er samplingsfrekvensen og  $f$  er den sanne romlige frekvensen.

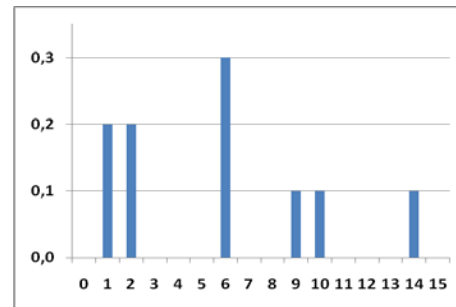
Anti-aliasing er teknikker for å dempe eller fjerne aliasing, for eksempel ved å filtrere bort høye frekvenser før sampling.

- c) I stedet for *bits* som kan lagre to verdier (0 og 1) kan vi ta i bruk *trits* som kan lagre tre verdier (-1, 0 og 1). På samme måte kan vi bruke en *tryte* = 6 *trits* istedenfor 1 *byte* = 8 *bits*. Hvis vi i utgangspunktet har en *tryte* per piksel i et bilde, og så halverer antall *trits* per piksel, hvor mange kvantiseringsnivåer vil vi da miste?

Svar: Med  $T$  *trits* har vi  $3^T$  kvantiseringsnivåer. Etter halveringen av antall *trits* per piksel har vi  $3^{(T/2)}$ . Altså har vi  $3^T - 3^{(T/2)}$  færre nivåer. For  $T = 6$  vil dette si at vi mister  $3^6 - 3^3 = 729 - 27 = 702$  nivåer.

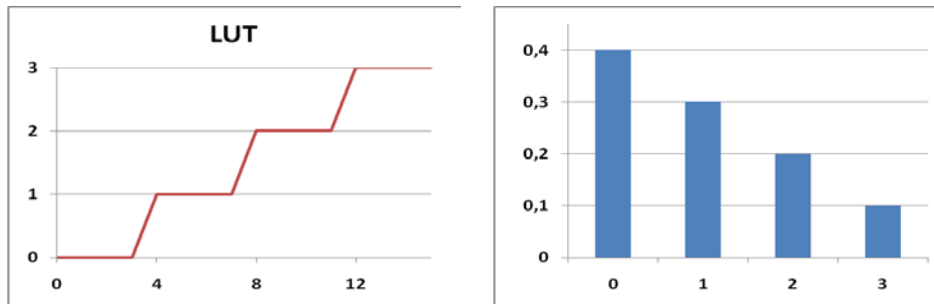
## 2. Kvantisering og histogram

Anta at du har et 4-bits gråtonebilde med normalisert histogram som skissert til høyre. Bildet inneholder en bakgrunn med to gråtoner og tre typer objekter.



- a) Vi ønsker å rekvantisere bildet til 2 bits per piksel, det vil si til et bilde med 4 verdier fra 0 til 3. Skisser den gråtonetransformen  $T(i)$  dette svarer til, og vis hvordan det normaliserte histogrammet til utbildet ville blitt.

Svar:

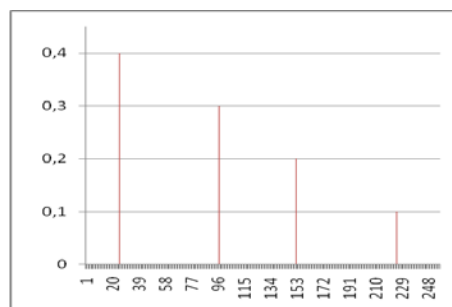


- b) Anta at originalbildet er  $1024 \times 1024$  piksler, og at vi ikke benytter kompresjon. Hvor stor lagerplass tar da dette bildet, uttrykt i MiB?

Svar: Veldig enkelt:  $4/8 = 1/2$  byte per piksel gir  $1/2$  MiB.

- c) Anta at du skal rekonstruere det rekvantiserte bildet til et 8 bits gråtonebilde. Hvilke verdier ville du brukt som rekonstruksjonsverdier for at bildets normaliserte histogram skal fylle gråtoneskalaen på omtrent samme måte som i original-bildet?

Svar: Det er en faktor 16 mellom gråtoneskalaen i originalbildet og gråtoneskalaen i det rekvantiserte bildet. Bruker vi  $1,5 \times 16 = 24$  som rekonstruksjonsnivå for 0;  $6 \times 16 = 96$  for 1;  $9,5 \times 16 = 152$  for 2 og  $14 \times 16 = 224$  for 3, så får vi histogrammet nedenfor.



### 3. Interpolasjon og filtrering

- a) Gitt pikselverdiene i fire nabopikslers er  $f(0,0)=1$ ,  $f(0,1)=3$ ,  $f(1,0)=3$ ,  $f(1,1)=9$ . Vi gjør bilinear interpolasjon for å finne en interpolert pikselverdi i punktet  $(x,y) = (0.25,0.25)$ . Hvilken pikselverdi får vi? Vis hvordan du går fram.

Svar: Ved interpolasjon finner man  $f(0.25,0.25) = 2.25$ , som rundes av til 2.

Dette kan finnes på minst to måter, begge hentet fra en forelesningsfoil:

- 1) Interpoler først i x-retning. Interpoler deretter i y-retning. Altså:

$$\begin{aligned} f(x,y) &\approx (1-y)f(x,0) + yf(x,1) \\ \text{der } f(x,0) &\approx (1-x)f(0,0) + xf(1,0) \\ \text{og } f(x,1) &\approx (1-x)f(0,1) + xf(1,1) \\ \Rightarrow f(x,y) &\approx (1-x)(1-y)f(0,0) + x(1-y)f(1,0) + (1-x)yf(0,1) + xyf(1,1) \end{aligned}$$

- 2) Eller i matrisenotasjon:

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

- b) En gitt baklengs geometrisk transform forskyver bildet  $\frac{1}{2}$  piksel i horisontal og vertikal retning, og bilinear interpolasjon benyttes til å finne nye pikselverdier. Deretter forskyves bildet tilbake, og igjen benyttes bilinear interpolasjon.

Hvilket konvolusjons-filter anvendt på det opprinnelige bildet gir samme resultat som denne fram-og-tilbake transformen med to bilineære interpolasjoner? Forklar!

(Du kan se bort fra problemer nær kanten av bildet).

Svar: Bilinear interpolasjon til midtpunktet mellom fire pikslers gir middelverdien av de fire pikselverdiene. Altså lavpassfilteret

$$\frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Ved tilbake-forskyvningen skjer det samme en gang til, og vi får 3x3-filteret

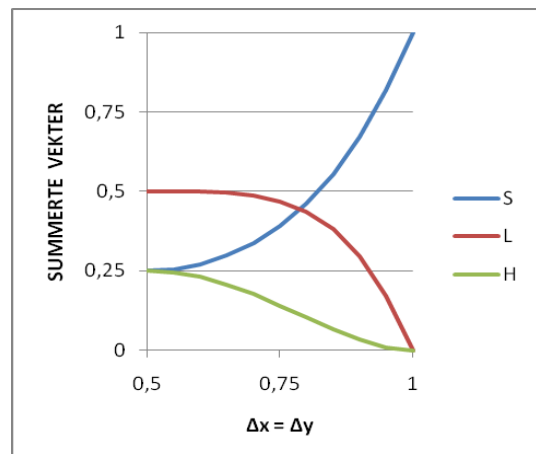
$$\frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- c) Anta at forskyvningen fram og tilbake er gitt ved  $\Delta x = \Delta y = N + k$ , der N er et heltall  $\geq 0$ , og  $\frac{1}{2} < k \leq 1$ . Vil resultatbildet da bli skarpere eller mer uskarpt enn resultatet av de to forskyvningene i deloppgave b)? Forklar!

Svar: Man kan kanskje tro at større forskyvning gir mer uskarpt bilde, men mest uskarpt resultat får vi når  $k = 1/2$ , som gir en lavpassfiltrering med det filteret som er løsning på deloppgave b. N påvirker ikke resultatet, bortsett fra de problemene som oppstår nær kanten av bildet.

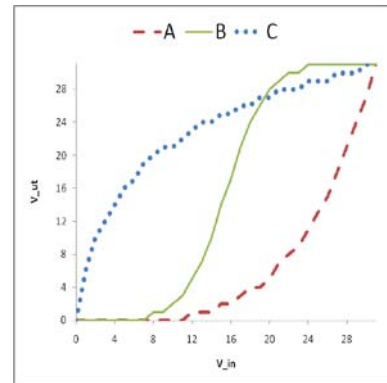
For  $\frac{1}{2} < k \leq 1$  vil vi komme nærmere det motstående pikslet i et 2x2 utsnitt av bildet, og dette motstående pikslet får større vekt dess nærmere k kommer til 1, som vist i figuren til høyre, og bildet blir like skarpt som originalen.

Filtervektene gitt for sentrum av 3x3 filteret (S), sum av langsiden (L), og sum av hjørnene (H).



## 4. Gråtonetransformer

I figuren til høyre er det gitt tre forskjellige gråtonetransformer, A, B og C. Horizontal og vertikal akse i figuren er hhv gråtone i innbildet (i) og gråtone i utbildet (s). Vi regner her med 5 bits gråtonebilder.



Ligningene for de tre transformene er i vilkårlig rekkefølge:

$$L-1: \quad s = k \log(i+1); \quad k = (2^b - 1) / b$$

$$L-2: \quad s = a i^\gamma; \quad a = (2^b - 1)^{-3}, \quad \gamma = 4$$

$$L-3: \quad s = (2^b - 1) [1 + \tanh(d(i - T))] / 2; \quad d = \frac{1}{4}, \quad T = (2^b - 1) / 2$$

der b er antall bits og  $\tanh(x)$  er en anti-symmetrisk sigmoid-funksjon som er lik 0 for  $x = 0$ , og som går mot -1 for negative x og +1 for positive x.

a) Hvilken effekt har transformene A, B og C på et gitt innbilde ?

Forklar resonnementene!

Svar: A vil minske kontrasten i de mørke delene av et bilde, og øke kontrasten i de lyse delene av bildet. Utbildet blir mørkere enn innbildet. B vil minske kontrasten både i de lyse og de mørke delene av bildet, og øke kontrasten omkring midten av gråtoneskalaen. Lysheten endres ikke for innbilder der alle gråtonene forekommer like ofte eller, mer generelt, der innbildets histogram er symmetrisk om T. Generelt vil utbildet kunne være enten lysere eller mørkere enn innbildet avhengig av hvordan gråtonene i innbildet er. C vil øke kontrasten i de mørke delene av bildet, og minske den i de lyse delene. Utbildet blir lysere enn innbildet.

b) Hvilken ligning svarer til hvilken av transformene A, B og C ?

Forklar resonnementene!

Svar:

- C er en logaritmisk gråtonetransform gitt ved ligning L-1. I argumentet har vi (i+1) for at vi skal unngå problemer med logaritmen av 0. Dermed får vi  $s[i=0]=0$ , mens faktoren k sikrer at  $s[2^5 - 1] = 2^5 - 1$ .
- A er en eksponensiell gråtonetransform gitt ved ligning L-2 med  $\gamma=4$ . Skaleringsfaktoren a skal bare sørge for at  $s[2^5 - 1] = 2^5 - 1$ .
- B er ligning L-3.  $[1 + \tanh(x)]$  sikrer at resultatet ligger mellom 0 og 1. Her er argumentet flyttet til midt på gråtoneskalaen med parameteren T, og skalert med parameteren d. Til slutt er s skalert slik at  $s[2^5 - 1] = 2^5 - 1$

c) Hva blir effekten av å endre på parametrene  $\gamma$ , T og d ?

Forklar resonnementene!

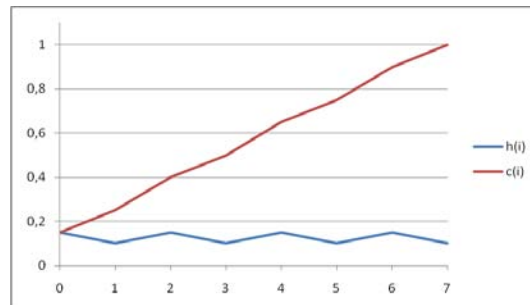
Svar:  $\gamma > 4$  ville gitt enda høyere kontrast i de lyse delene av bildet, og enda lavere kontrast i de mørke delene av bildet. Lavere  $\gamma$  vil gi mindre kontrastforsterkning i de lyse delene av bildet, ned til  $\gamma=1$  som er en identitetsmapping.  $\gamma < 1$  gir motsatt effekt; økt kontrast i mørke deler av bildet og minsket kontrast i lyse deler av bildet. T bestemmer hvor på gråtoneskalaen vi vil sentrere kontrastforsterkningen, og d bestemmer hvor bratt sigmois-kurven skal være; høy d gir et lite intervall, lav d gir et bredt intervall.

## 5. Histogramtransformer

Anta at vi har følgende 4x5 gråtonebilde med en 3 bits gråtoneskala.

5	7	2	0	6
0	4	1	6	3
3	4	2	5	1
7	4	2	6	0

- a) Finn det normaliserte histogrammet og det normaliserte kumulative histogrammet.  $[3/20, 2/20, 3/20, 2/20, 3/20, 2/20, 3/20, 2/20]$ ;  $[3/20, 5/20, 8/20, 10/20, 13/20, 15/20, 18/20, 1]$



- b) Vis hvordan du går fram for å utføre en histogramutjevning av dette bildet til et utbilde med bare 4 gråtoner fra gråtone 0 til gråtone 3. Vis også resultatbildet.

Sett inn verdier i transform-arrayet

$T[i] = \text{Round}((L-1)*c[i]+k)$  med  $L=4$  og  $k=0$  for  $i = 0, 1, \dots, G-1$ , der  $G=8$ .

i	p(i)	c(i)	T(i)
0	0.15	0.15	0
1	0.1	0.25	1
2	0.15	0.40	1
3	0.1	0.50	2
4	0.15	0.65	2
5	0.1	0.75	2
6	0.15	0.90	3
7	0.1	1.00	3

(Tabellen er en del av løsningsforslaget)

Gå deretter gjennom bildet piksel for piksel, og sett  $g(x,y) = T[i(x,y)]$ .

Resultatet blir da

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

(Tabellen er en del av løsningsforslaget)

- c) Beskriv en alternativ metode som gir et resultatbilde med flatt histogram for akkurat dette innbildet. Begrunn valget av metode. Vis resultatbilde og histogram, og sammenlign med resultatet av histogramutjevningen.

Det spesielle med dette innbildet er at det har et tilnærmet flatt histogram.

Dessuten har vi bedt om en reduksjon fra 8 til 4 gråtoner.

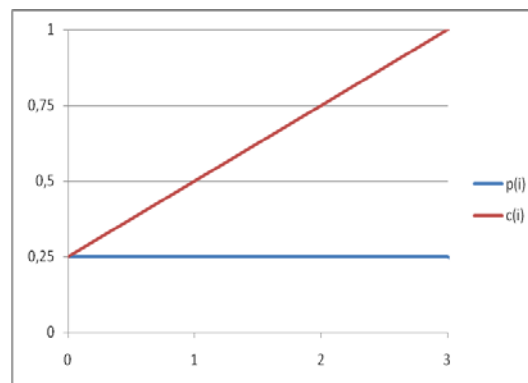
Vi kan altså ganske enkelt redusere antall bits fra 3 til 2 i hver gråtone.

Vi får da følgende LUT

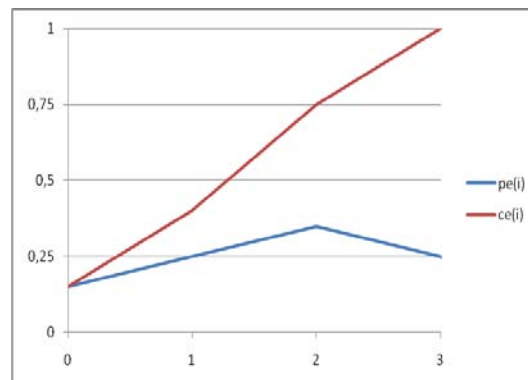
i	LUT
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	3

Og resultatbilde med histogrammer:

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



Som faktisk er litt "bedre" enn resultatet av histogramutjevningen:



## 6. Kantdeteksjon med LoG-filtrering

Merk: Deloppgave a) og b) ber deg beregne konvolusjoner og majoriteten av poengene som gis til disse deloppgavene ligger i å utføre disse korrekt.

I denne oppgaven skal du finne kantskinner i følgende én-dimensjonale bilder:

$$f_1 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$f_2 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 5 & 5 & 5 & 5 & 5 \\ \hline \end{array}$$

Det er to kantskinner for  $f_1$  og ett kantskille for  $f_2$ . Alle tre kantskillene ligger midt mellom to piksler og er markert med ekstra fet cellekant i bildene over.

Til å finne kantskillene skal du bruke følgende én-dimensjonale Laplacian-of-Gaussian-filtre (LoG-filtre), også kalt LoG-operatorer:

$$h_1 = \begin{array}{|c|c|c|c|c|} \hline -1 & -2 & 6 & -2 & -1 \\ \hline \end{array}$$

$$h_2 = \begin{array}{|c|c|c|c|c|c|c|} \hline -1 & -2 & -1 & 8 & -1 & -2 & -1 \\ \hline \end{array}$$

$$h_3 = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline -1 & -1 & -2 & 1 & 6 & 1 & -2 & -1 & -1 \\ \hline \end{array}$$

Alle filtre er sentrert, d.v.s. at filterets senterpiksel er origo.

Når vi i denne oppgaven ber deg beregne en konvolusjon, så trenger du bare å beregne responsen for pikslene der bildet og filteret overlapper i alle posisjoner, d.v.s. de pikselene der hele filteret ligger innenfor bildet når filterets origo er plassert i pikselet man ønsker å beregne responsen for.

Når vi i denne oppgaven snakker om en *nullgjennomgang* i resultatet av en LoG-filtrering, så mener vi punktet midt mellom to nabo-piksler som har motsatt fortegn i resultatet av LoG-filtreringen og der begge LoG-responsene er ulik 0.



- a) Beregn konvolusjonen av  $f_1$  og hvert av filtrene  $h_1$ ,  $h_2$  og  $h_3$ , d.v.s.  $f_1 * h_1$ ,  $f_1 * h_2$  og  $f_1 * h_3$ , og angi nullgjennomgangene i hvert resultat av de tre filtreringene.  
Siden bildet består av bare 0-ere utenom én posisjon der det er 1, så vil konvolusjonen av bildet og et filter være filteret selv. Vi får dermed at:

$$f_1 * h_1 = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & -1 & -2 & \mathbf{6} & -2 & -1 & 0 & 0 \\ \hline \end{array}$$

$$f_1 * h_2 = \begin{array}{|c|c|c|c|c|c|c|} \hline -1 & -2 & -1 & \mathbf{8} & -1 & -2 & -1 \\ \hline \end{array}$$

$$f_1 * h_3 = \begin{array}{|c|c|c|c|c|} \hline -2 & \mathbf{1} & 6 & \mathbf{1} & -2 \\ \hline \end{array}$$

når vi begrenser størrelsen av responsen til posisjonene med full overlapp. Nullgjennomgangene i hvert filtreringsresultat er markert med ekstra fet cellekant.

- b) Beregn konvolusjonen av  $f_2$  og hvert av filtrene  $h_1$  og  $h_2$ , d.v.s.  $f_2 * h_1$  og  $f_2 * h_2$ , og angi nullgjennomgangene i hvert resultat av de to filtreringene.

$$f_2 * h_1 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline -1 & -4 & -1 & 0 & 0 & 1 & 4 & 1 \\ \hline \end{array}$$

Ingen nullgjennomgang i dette filtreringsresultatet (bare ett nullplata).

$$f_2 * h_2 = \begin{array}{|c|c|c|c|c|c|} \hline -8 & -4 & -1 & \mathbf{1} & 4 & 8 \\ \hline \end{array}$$

Nullgjennomgangen i dette filtreringsresultatet er markert med ekstra fet cellekant.

- c) Drøft hvordan standardavviket til Gauss-funksjonen i et LoG-filter, som gir bredden av LoG-kjernen og antyder størrelsen av LoG-filteret, generelt sett bør velges for at nullgjennomgangene i resultatet av LoG-filtreringen skal gi alle og korrekte kantskinner for strukturer og ramper.

Bruk gjerne filtreringene fra deloppgave a) og b) som eksempler, men ikke begrensn drøftingen til bare disse eksemplene.

Upresist sagt gir nullgjennomgangene korrekte kantskinner dersom LoG-kjernen er smalere enn strukturen. Mer presist:

- Dersom en struktur er mindre enn halvparten av LoG-kjernen er vi garantert at nullgjennomgangene er lenger ute enn de korrekte kantskillene. Dette skjedde da vi konvolverte  $f_1$  med  $h_3$ .
- Dersom en struktur er større enn halvparten av LoG-filteret er vi garantert at nullgjennomgangene gir de korrekte kantskillene.
- Dersom en struktur er større enn halvparten av LoG-kjernen, men mindre enn halvparten av filteret, så vil det avhenge av diskretiseringen og tilnærmingen av LoG-filteret om nullgjennomgangene gir korrekte kantskinner. Konvolusjonene  $f_1 * h_1$  og  $f_1 * h_2$  faller begge innenfor dette tilfellet. Etter akkurat disse to konvolusjonene endte vi opp med korrekte kantskinner.

For at LoG-filtreringen skal inneholde en nullgjennomgang for ramper så må LoG-filteret være større enn rampen. Siden  $h_1$  er én piksel mindre enn rampen i  $f_2$  så inneholdt ikke  $f_2 * h_1$  noen nullgjennomgang, men siden  $h_2$  er én piksel større enn rampen i  $f_2$  så inneholdt  $f_2 * h_2$  en nullgjennomgang (og denne blir lokalisert korrekt).

Standardavviket til Gauss-funksjonen i et LoG-filter må altså velges lite nok for å gi korrekte kantskinner for smale strukturer og samtidig stort nok for å gi nullgjennomgang (og dermed et kantskille) for brede ramper.

## 7. Filtrering for deteksjon av horisontale kanter

- a) Anta vi ønsker å fremheve horisontale kanter i et bilde ved å bruke et konvolusjonsfilter som tilnærmer den deriverte i vertikal retning, d.v.s. tilnærmer den partiell-deriverte med hensyn på variabelen til den vertikale akse,  $x$ . Oppgi et slik konvolusjonsfilter og forklar hvordan det tilnærmer den deriverte i vertikal retning når det konvolveres med et bilde.

Du kan oppgi et vilkårlig konvolusjonsfilter som tilnærmer den deriverte i vertikal retning, f.eks.:

- Fra asymmetrisk 1D-operator:  $h_x(i, j) = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$  eller  $h_x(i, j) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

- Fra symmetrisk 1D-operator:  $h_x(i, j) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$  eller  $h_x(i, j) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$

- Fra Prewitt-operatoren:  $h_x(i, j) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

- Fra Sobel-operatoren:  $h_x(i, j) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

- Fra Frei-Chen-operatoren:  $h_x(i, j) = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$

Forklaring: Den deriverte i vertikal retning er gitt som:

$$\frac{\partial f}{\partial x}(x, y) = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h}$$

Konvolusjonsfilteret tilnærmer dette ved å beregne differansen i vertikal retning av nærliggende piksler, noe som tilsvarer å sette  $h \geq 1$  i likningen over.

- b) Beregn en mer støyrobust versjon av konvolusjonsfilteret du oppga i deloppgave a). Konvolusjonsfilter fra deloppgave a) skal inngå i beregningen og det mer støyrobuste filteret skal også være et konvolusjonsfilter.

Her skal du konvolvare filteret du oppga i deloppgave a) med et lavpassfilter. Hvis du oppga filteret fra den symmetriske 1D-operatoren i deloppgave a), kan du nå konvolvare det med f.eks. en 1x3-tilnærming av et Gauss-filter:

$$h_2(i, j) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * [1 \ 2 \ 1] = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- c) Hvordan har vi lært at et Laplace-filter, også kalt en Laplace-operator, kan gjøres mer støyrobust? Hva kaller vi det resulterende konvolusjonsfilteret?

Et Laplace-filter kan gjøres mer støyrobust ved å konvolvare filteret med et Gauss-filter. Det resulterende konvolusjonsfilteret kalles et Laplacian-of-Gaussian-filter (LoG-filter) og en LoG-operator.

***TAKK FOR OPPMERKSOMHETEN!***