1. Geometriske operasjoner

a) Vi har følgende sekvens av geometriske operasjoner som anvendes på et bilde:

1. Først anvender vi transformasjonen

$$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Deretter transformasjonen

$$\begin{bmatrix} 1 & s_y & 0 \\ s_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Og til slutt transformasjonen

$$\begin{bmatrix} \frac{1}{c_x} & 0 & 0 \\ 0 & \frac{1}{c_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Her er c_x, c_y, s_x og s_y vilkårlige tall som utgjør transformasjonskoeffisientene. Forklar hva denne sekvensen av transformasjoner gjør på et bilde.

Løsningsforslag: Her er det ikke viktig hva vi setter et bildets x- og y-akse til å være. Studenten skal derimot kunne forklare at de ulike transformasjonskoeffisienter anvendes på ulike akser til bildet.

1. *Først skaleres bildet opp med c_x langs x-aksen til bildet og c_y langs y-aksen.*
2. *Så anvendes det en "shear" transformasjon, der x-koordinatene adderes med $s_y y$ og y-koordinatene adderes med $s_x x$.*
3. *Til slutt gjøres en invers skalering av bildet vi får etter steg 2., altså en krymping av det, med $\frac{1}{c_x}$ langs x-aksen og med $\frac{1}{c_y}$ langs y-aksen.*

b) Istedenfor å ha en sekvens av transformasjoner, kan vi slå sammen transformasjonene til én transformasjon. Vis hvordan transformasjonene kan slås sammen og hvilken transformasjonsmatrise du får som resultat.

Løsningsforslag: Vi kan slå sammen transformasjonene ved å multiplisere matrisene sammen. Den første transformasjonen er lengst til høyre i matrisemultiplikasjonen, etterfulgt av den andre transformasjonen og til slutt den tredje transformasjonen:

$$\begin{bmatrix} \frac{1}{c_x} & 0 & 0 \\ 0 & \frac{1}{c_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & s_y & 0 \\ s_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{c_x} & 0 & 0 \\ 0 & \frac{1}{c_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x & s_y\,c_y & 0 \\ s_x c_x & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{s_y\,c_y}{c_x} & 0 \\ \frac{s_x\,c_x}{c_y} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformasjonsmatrisen blir altså

$$\begin{bmatrix} 1 & \frac{s_y\,c_y}{c_x} & 0 \\ \frac{s_x\,c_x}{c_y} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

c) Én eller fler av koeffisientene c_x, c_y, s_x og s_y kan ha verdi(er) som gjør at utbildet blir lik innbildet. Skriv koeffisienten(e) og tilhørende verdi(er) som gir dette utbildet.

Løsninsforslag: Fra deloppgave b, kan vi uttrykke transformasjonene som én matrise,

$$\begin{bmatrix} 1 & \frac{s_y\,c_y}{c_x} & 0 \\ \frac{s_x\,c_x}{c_y} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For at utbildet skal bli helt lik innbildet etter transformasjonene, må transformasjonsmatrisen bli identitetsmatrisen. Er transformasjonsmatrisen en identitetsmatrise, vil koordinatene til bildet bare mappes til seg selv. Det betyr at vi må sette verdier til transformasjonskoeffisientene slik at

$$\frac{s_y\,c_y}{c_x} = 0$$

og

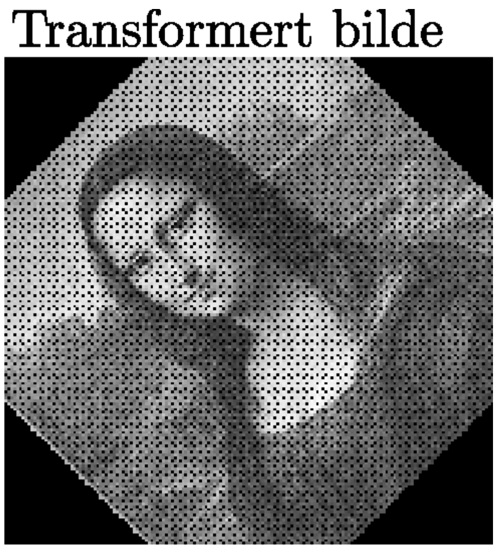
$$\frac{s_x\,c_x}{c_y} = 0$$

Vi kan ikke sette $c_x = 0$ eller $c_y = 0$ siden vi får 0 i nevner. Den eneste muligheten er å la $s_x = 0$ og $s_y = 0$.

Denne deloppgaven kan også løses uten å ha gjort deloppgave b. Siden første transformasjon er en skalering, og siste transformasjon er en krymping av bildet med samme faktorer langs bildets akser, er de nok å ikke gjøre noe "shearing" på bildet for å få et utbilde som er lik innbildet.

2. Anvendelse av geometriske operasjoner

a) Gitt et bilde der vi har anvendt en geometrisk operasjon på det:



Hva kan være årsaken til at vi får slike "hull" i bildet, altså sorte piksler inni selve bildet, etter å ha anvendt den geometriske operasjonen? Forklar.

Løsningsforslag: Slike hull i utbildet oppstår når vi bruker forlengsmapping til å anvende geometriske transformasjoner på et bilde. I baklengsmapping, mapper vi koordinatene til innbildets piksler. Det kan hende at etter mappingen blir koordinatene lik andre koordinater som allerede har fått pikselverdier eller havner utenfor bildet. Vi er derfor ikke garantert at alle utbildets piksler får en verdi med forlengsmapping, som gir opphav til disse "hullene".

b) Forklar hvordan og hvorfor vi kan unngå slike hull som i bildet fra deloppgave a når vi anvender en geometrisk transformasjon.

Løsningsforslag: Utfordringen med forlengsmappig, er at alle utbildets piksler ikke nødvendigvis får en verdi. Dette prøver baklengsmapping å ordne. I baklengsmapping iterer vi heller gjennom utbildets piksel-koordinater og tar en invers mapping av dem. Den inverse mappingen sørger for at vi får et koordinatpar fra innbildet som vi kan hente en pikselverdi fra eller approksimere verdien ved hjelp av interpolasjon.

3. Histogram

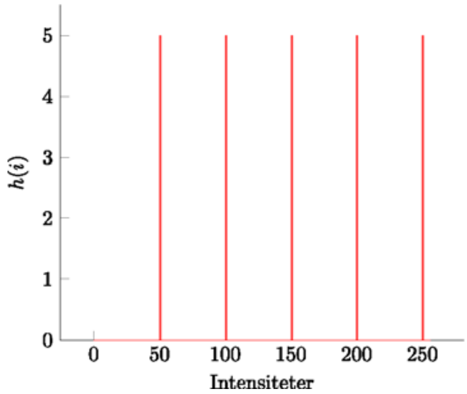
Vi har to 8-bit bilder sammen med deres histogram, Bilde 1

50	100	150	200	250
50	100	150	200	250
50	100	150	200	250
50	100	150	200	250
50	100	150	200	250

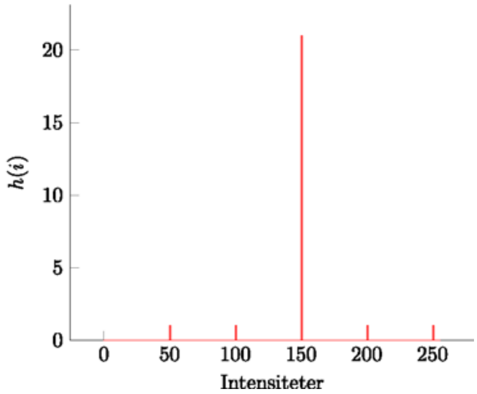
Bilde 2

150	150	150	150	150
150	150	150	150	150
50	100	150	200	250
150	150	150	150	150
150	150	150	150	150

Histogram til bilde 1



Histogram til bilde 2



a) Regn ut bildenes middelverdi og varians og vis hvordan du går frem.

Løsningsforslag: Vi har at middelverdien μ og variansen σ^2til et 8-bit bilde med normalisert histogram p(i) kan regnes ut ved

$$\mu = \sum_{i=0}^{255} ip(i)$$

$$\sigma^2 = \sum_{i=0}^{255} i^2 p(i) - \mu^2$$

La middelverdien og variansen til bilde 1 være henholdsvis μ_1 og σ_1^2 middelverdien og varians til bilde 2 være henholdsvis μ_2 og σ_2^2. Vi kan normalisere de gitte histogrammene ved å dele dem på totalt antall piksler og få at

$$\begin{aligned}\mu_1 &= 50\frac{5}{25} + 100\frac{5}{25} + 150\frac{5}{25} + 200\frac{5}{25} + 250\frac{5}{25} \\ &= \frac{1}{5}(50 + 100 + 150 + 200 + 250) \\ &= \frac{750}{5} \\ &= 150\end{aligned}$$

$$\sigma_1^2 = 50^2\frac{5}{25} + 100^2\frac{5}{25} + 150^2\frac{5}{25} + 200^2\frac{5}{25} + 250^2\frac{5}{25} - 150^2$$

$$= \frac{1}{5}(50^2 + 100^2 + 150^2 + 200^2 + 250^2) - 150^2$$
$$= \frac{137500}{5} - 22500$$
$$= 5000$$

$$\mu_2 = 50\frac{1}{25} + 100\frac{1}{25} + 150\frac{21}{25} + 200\frac{1}{25} + 250\frac{1}{25}$$
$$= \frac{1}{5}(50 + 100 + 150 \times 21 + 200 + 250)$$
$$= \frac{3750}{25}$$
$$= 150$$

$$\sigma_2^2 = 50^2\frac{1}{25} + 100^2\frac{1}{25} + 150^2\frac{21}{25} + 200^2\frac{1}{25} + 250^2\frac{1}{25} - 150^2$$
$$= \frac{1}{5}(50^2 + 100^2 + 150^2 \times 21 + 200^2 + 250^2) - 150^2$$
$$= \frac{587500}{25} - 22500$$
$$= 1000$$

Bilde 1 har altså middelverdi 150 og varians 5000, mens bilde 2 har middelverdi 150 og varians 1000.

b) Forklar sammenhengen mellom formen på histogrammene og deres middelverdi og varians.

Løsningsforslag: Dette gir middelverdien til bildet. Bilde 1 har like mange ikke-null pikselverdier. Spredningen mellom verdiene blir da større enn hvis majoriteten er på 150, som i bilde 2. Vi har derfor at variansen i bilde 1 er større sammenlignet med bilde 2. Vi ser også av histogrammet til bilde 2 at det er flest piksler som har verdi 150, som gjør at vi kan forvente at middelverdien blir rundt 150. I dette tilfellet ble middelverdien nøyaktig 150 på grunn av symmetri i histogrammet.

4. Histogramutjevning

Gjør histogramutjevning på 3-bit bildet under for hånd og vis alle stegene du gjør.

Bilde som skal histogramutjevnes

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

Løsningsforslag: Histogramutjevning kan f.eks deles inn i fire steg:

1. Finne histogrammet til bildet og deretter normalisere det. Histogrammet h(i) og det normaliserte histogrammet p(i) til bildet er

i	*h(i)*	*p(i)*
0	5	$\frac{5}{30}$
1	4	$\frac{4}{30}$
2	6	$\frac{6}{30}$
3	4	$\frac{4}{30}$
4	0	0
5	3	$\frac{3}{30}$
6	4	$\frac{4}{30}$
7	4	$\frac{4}{30}$

2. Vi finner så det kumulative normaliserte histogrammet c(i) til bildet:

i	*c(i)*
0	$\frac{5}{30}$
1	$\frac{9}{30}$
2	$\frac{15}{30}$
3	$\frac{19}{30}$

4	$\frac{19}{30}$
5	$\frac{22}{30}$
6	$\frac{26}{30}$
7	$\frac{30}{30}$

3. Vi finner LUT-en T[i] = round((G-1)c(i)) = round(7c(i))

i	*T[i]*
0	*1*
1	*2*
2	*4*
3	*4*
4	*4*
5	*5*
6	*6*
7	*7*

5. Og til slutt anvender vi LUT-en på inn-bildet og får

$$\begin{bmatrix} 2 & 4 & 1 & 4 & 4 & 4 \\ 1 & 5 & 5 & 1 & 1 & 2 \\ 4 & 2 & 4 & 6 & 7 & 7 \\ 6 & 7 & 2 & 4 & 4 & 6 \\ 1 & 4 & 4 & 5 & 6 & 7 \end{bmatrix}$$

6. Konvolusjon

Anta at gråtoneflaten i et 3x3 piksels område omkring en pikselposisjon (x,y) i et bilde er gitt ved

$$f(x+i, y+j) = ai + bj + c$$

		i	
	-1,1	0,1	1,1
j	-1, 0	0, 0	1, 0
	-1, -1	0, -1	1, -1

a) Vis hvordan vi kan bruke Sobel-operatorene nedenfor til å estimere de lokale stigningstallene a og b i uttrykket ovenfor.

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

Løsningsforslag: Hvis intensiteten omkring punktet (x,y) i et bilde er gitt ved ligningen ovenfor, så er pikselverdiene i et 3x3 område omkring punktet gitt ved

-a+b+c	b+c	a+b+c
-a+c	c	a+c
-a-b+c	-b+c	a-b+c

Sobel-operatorene konvolvert med intensitetene ovenfor gir gradient-estimatene i punktet (x,y). Husk at operatorene roteres 180 grader før multiplikasjon, summasjon:

$$
\begin{aligned}
g_x(x,y) &= \begin{bmatrix} -a+b+c & b+c & a+b+c \\ -a+c & c & a+c \\ -a-b+c & -b+c & a-b+c \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \\
&= (-a+b+c)(-1)+(b+c)(0)+(a+b+c)(1) \\
&\quad +(-a+c)(-2)+(c)(0)+(a+c)(2) \\
&\quad +(-a-b+c)(-1)+(-b+c)(0)+(a-b+c)(1) \\
&= a-b-c+a+b+c+2a-2c+2a+2c+a+b-c+a-b+c \\
&= \underline{\underline{8a}}.
\end{aligned}
$$

Og tilsvarende i y-retningen:

$$
\begin{aligned}
g_y(x,y) &= \begin{bmatrix} -a+b+c & b+c & a+b+c \\ -a+c & c & a+c \\ -a-b+c & -b+c & a-b+c \end{bmatrix} * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\
&= (-a+b+c)(1)+(b+c)(2)+(a+b+c)(1) \\
&\quad +(-a+c)(0)+(c)(0)+(a+c)(0) \\
&\quad +(-a-b+c)(-1)+(-b+c)(-2)+(a-b+c)(-1) \\
&= -a+b+c+2b+2c+a+b+c+a+b-c+2b-2c-a+b-c \\
&= \underline{\underline{8b}}.
\end{aligned}
$$

b) Forklar hvordan og hvorfor vi må skalere resultatene fra Sobel-operatorene for å finne riktige estimater av stigningstallene a og b.

Løsningsforslag: Et estimat av stigningstallet ved hjelp av to piksler er gråtonedifferansen dividert med avstanden mellom pikslene. Operatorene estimerer her stigningstallene a og b i x- og y-retning ved å ta differansen mellom 2 striper a 3 piksler som ligger 2 piksler fra hverandre. På grunn av avstanden 2 må vi dividere estimatet med 2. Deretter må vi dividere med summen av tallverdien av vektene innenfor hver stripe, som er 4.
Vi må altså dividere resultatene fra Sobel-operatorene med 8 for å finne de riktige estimatene av stigningstallene a og b.

c) Bruk 3x3 Frei-Chen operatorene nedenfor til å finne en 5x5 approksimasjon til Laplace-operatoren.

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

Løsningsforslag: Laplace-operatoren er gitt ved

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

Og en approksimasjon finnes ved å konvolvere hver av de to gradient-operatorene med seg selv, summere resultatet, og skifte fortegn (husk å rotere 180 grader!):

$$-\nabla^2\left(f(x,y)\right) = -\left(\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} + \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} * \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \right)$$

$$= -\left(\begin{bmatrix} 1 & 0 & -2 & 0 & 1 \\ 2\sqrt{2} & 0 & -4\sqrt{2} & 0 & 2\sqrt{2} \\ 4 & 0 & -8 & 0 & 4 \\ 2\sqrt{2} & 0 & -4\sqrt{2} & 0 & 2\sqrt{2} \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2\sqrt{2} & 4 & 2\sqrt{2} & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -4\sqrt{2} & -8 & -4\sqrt{2} & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2\sqrt{2} & 4 & 2\sqrt{2} & 1 \end{bmatrix} \right) = \begin{bmatrix} -2 & -2\sqrt{2} & -2 & -2\sqrt{2} & -2 \\ -2\sqrt{2} & 0 & 4\sqrt{2} & 0 & -2\sqrt{2} \\ -2 & 4\sqrt{2} & 16 & 4\sqrt{2} & -2 \\ -2\sqrt{2} & 0 & 4\sqrt{2} & 0 & -2\sqrt{2} \\ -2 & -2\sqrt{2} & -2 & -2\sqrt{2} & -2 \end{bmatrix}$$

d) En tilnærming til en 5x1 og en 1x5 Gauss-profil kan produseres ved fire konvolusjoner av henholdsvis [1 1] og $[1\ 1]^T$. Benytt at Sobel-operatorene som er gitt i delspørsmål a) er separable og at konvolusjon er kommutativ og distributiv, og vis at en approksimasjon til en 9x9 LoG-operator kan skrives som

$$LoG\left(f(x,y)\right) = L * H * L^T * L^T + L * L * L^T * H^T$$

der L er lavpassfilteret [1 4 6 4 1] og H er høypassfiltret [-1 0 2 0 -1].

Løsningsforslag: Fire konvolusjoner av [1 1] gir [1 4 6 4 1] =L.
Fire konvolusjoner av $[1\ 1]^T$ gir $[1\ 4\ 6\ 4\ 1]^T = L^T$.

Konvolveres disse to får vi en 5x5 Gauss-profil: $L*L^T$.
De to 3x3 Sobel-operatorene gir en 5x5 approksimasjon til en Laplace-operator, på samme måte som ovenfor. Konvolusjonen av en 5x5 Laplace med en 5x5 Gauss gir en 9x9 Laplace of Gauss (LoG).

Sobel-operatorene er separable:

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

Vi benytter oss av at konvolusjon er kommutativ: A*B = B*A,
og distributiv: A*(B*C)=(A*B)*C.
Og LoG-operatoren kan skrives som

$$LoG\left(f(x,y)\right) = -\left(\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} + \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \right) * L * L^T$$

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

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

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

$$= \left(L^T * H + L * H^T \right) * L * L^T = \underline{\underline{L * H * L^T * L^T + L * L * L^T * H^T}}.$$

6. Konvolusjon

a) Du får oppgitt en kombinert konvolusjonsoperator:

$$O = -\left(A * A * B^T * B^T \ + \ A^T * A^T * B * B\right)$$
$$A = \begin{bmatrix}1 & 0 & -1\end{bmatrix}, \qquad B = \begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

Hvilken operator er O, og hva bruker vi den til?
Bruk egenskaper ved konvolusjon til å begrunne svaret uten å utføre konvolusjonen!

Løsningsforslag: Operatoren O kan omformes til

$$O = -\left((A * B^T) * (A * B^T) \ + \ (A^T * B) * (A^T * B)\right)$$

der ($A*B^T$) og (A^T*B) er de to konvolusjons-filtrene i Prewitt-operatoren som vi bruker til å estimerer gradienten (førstederiverte) i hhv x- og y-retning.
*($A*B^T$) * ($A*B^T$) og (A^T*B)* (A^T*B) gir oss da estimater av den andre-deriverte i hhv x- og y-retning. Så operatoren O svarer til en Laplace-operator realisert ved hjelp av Prewitt-operatoren.*

b) Hvor stor er filtermatrisen O ovenfor?
Begrunn svaret uten å utføre konvolusjonen!

Løsningsforslag: Operatoren O = -($A*A*B^T*B^T$ + A^T*A^T*B*B) kan skrives som

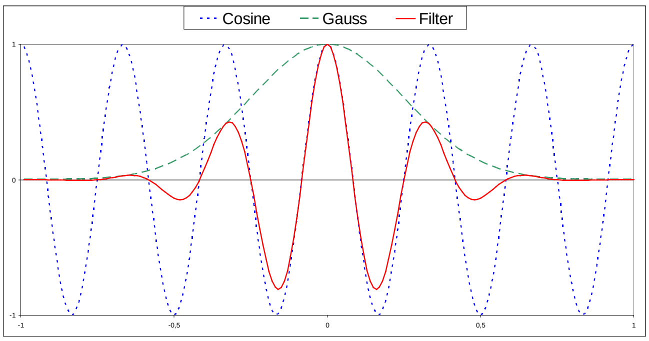
O = -(($A*A$)(B^T*B^T) + (A^T*A^T)*($B*B$)).*

Der ($A*A$) og ($B*B$) er radvektorer som er [5x1].

Og (A^T*A^T) og (B^T*B^T) er kolonnevektorer som er [1x5].

O er da en matrise som er [5x5].

c) Hvis vi lager oss et filter ved å multiplisere et 2-D lavpass Gauss-filter G(x,y) med en 1-D cosinus-funksjon Cos(x), så vil et snitt gjennom filtret langs x-aksen kunne se slik ut:



Filtret vil altså ha flere minima og maksima med avtagende amplitude på hver side av den sentrale toppen.

Hvilket teorem fra pensum kan du bruke til å forklare at Fourier-transformen av dette filtret bare har to maksima?
Forklar hvordan du bruker teoremet i dette tilfellet.

Løsningsforslag: Konvolusjonsteoremet sier at en multiplikasjon i bildedomenet svarer til en konvolusjon i Fourier-domenet. Fourier-transformen av filtret er da en konvolusjon av Fourier-transformen av Gaussfiltret - som er en 2-D Gauss-funksjon – og Fourier-transformen av en Cosinus-funksjon, som er to singulariteter på u-aksen ved ± frekvensen til sinusoiden.
Når det <u>bare</u> er disse to singularitetene som skal konvolveres med en 2-D Gauss, får vi bare to maksima.

7. Randutvidelse

Vi har et bilde med positive verdier langs randen og ønsker å anvende en gradient estimator på det. Vi kan bl.a velge mellom to ulike randutvidelser:

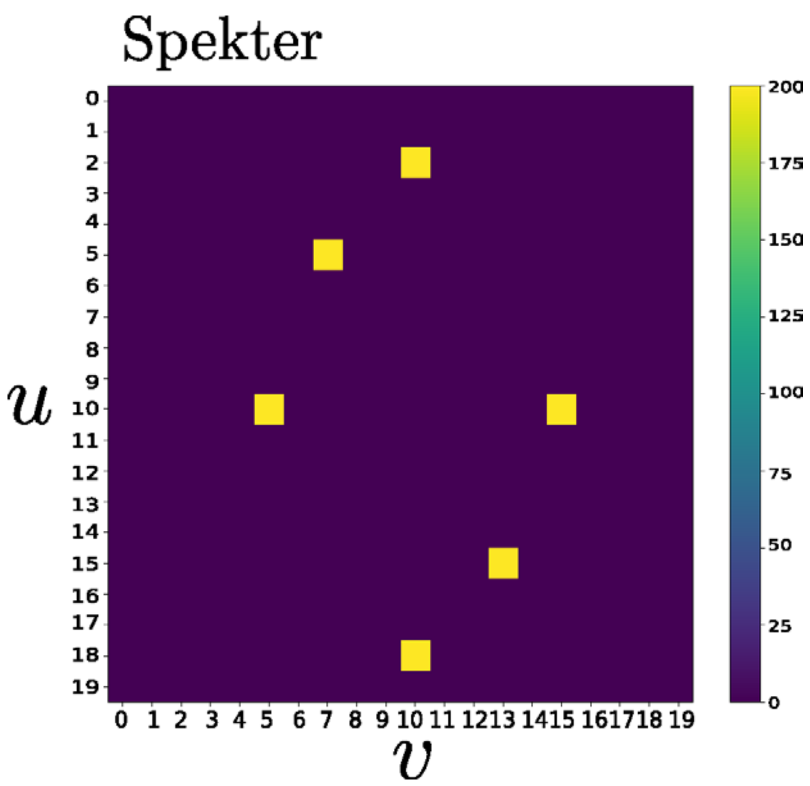
- Nærmeste nabo utviding
- Nullutviding

Hvilken av de to nevnte randutvidelser kan gi opphav til en ikke-null respons langs bilderanden etter anvendelsen av gradient estimatoren? Forklar hvorfor.

Løsningsforslag: Siden verdiene langs randen er ikke null, vil en nullutviding skape en diskontinuitet langs randen til bildet.
Dette vil gi en ikke-null respons langs bildekanten da gradienten, eller forandringen, langs bilderand og nullutvidet område vil bli større enn null.

8. Spekter

Et reelt bilde har følgende spekter:



Ved å se på dette spekteret, hva kan du si om hvordan bildet ser ut og hva det består av?

Løsningsforslag: I dette tilfellet, så kan vi til en viss grad si noe om hvordan bildet ser ut og hva det består av. Vi ser seks topper i spekteret. Siden bildet er reelt, er det kompleks konjugert symmetri i bildet. Bildet inneholder derfor tre frekvenskomponenter. Bildet består av en sum av sinuser eller cosinuser med frekvens (10, 5), (5, 7) og (2, 10). Vi kan ikke se fra spekteret alene om det er cosinuser eller sinuser i bildet.

9. Endring av spekter

Anta vi har Fourier-transformasjonen $F(u, v)$ til et bilde og at vi lager en modifikasjon $F_{ny}(u, v)$ av den. Vi modifiserer den ved å sette dens DC-komponent lik en annen verdi a og lar de andre verdiene være uforandret.
Vi får altså at

$$F_{ny}(u,v) = \begin{cases} a, & u = 0 \; og \; v = 0 \\ F(u,v), & ellers \end{cases}$$

Vi tar en inverstransformasjon av $F_{ny}(u, v)$ og får et bilde $f_{ny}(x, y)$,

$$f_{ny}(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F_{ny}(u,v) e^{2\pi j \left(\frac{ux}{M} + \frac{vy}{N}\right)}, \quad x = 0,1, \dots, M-1, y = 0,1, \dots, N-1$$

Hva er middelverdien til $f_{ny}(x, y)$? Forklar!

Løsningsforslag: Siden

$$F(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

til et vilkårlig bilde $f(x, y)$ og vi vet at $F_{ny}(u, v)$ er Fourier transformasjonen til $f_{ny}(x, y)$, så må

$$F_{ny}(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f_{ny}(x,y)$$

$$a = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f_{ny}(x,y)$$

Skalerer vi resultatet med $\frac{1}{MN}$ får vi middelverdien til $f_{ny}(x, y)$.
Vi får altså at middelverdien til $f_{ny}(x, y)$ er lik $\frac{a}{MN}$.

Merk at det er mulig å regne ut dette direkte, men utregningen kan være lang og er ikke forventet at studenten skal kunne gjøre.

10. Cosinus- og sinusbilder

Anta vi har fått et cosinusbilde og sinusbilde med frekvens $u = 2$ og $v = 1$. Bildene er 10×12 store. Hvordan kan vi bruke disse bildene til å finne frekvensresponsen ved $u' = 8$ og $v' = 11$ til et like stort bilde? Forklar.

Løsningsforslag: Vi kan bruke anti-symmetri til sinus og symmetri til cosinus. Fra anti-symmetri til en sinus, vil et $M \times N$ sinusbilde med frekvens u, v, for u, v ulik null, være lik minus seg selv ved frekvens $M - u$ og $N - v$.
Fra symmetri til cosinus, vil et $M \times N$ cosinusbilde med frekvens u, v, for u, v ulik null, være lik seg selv ved frekvens $M - u$ og $N - v$.
Siden $u' = 8 = 10 - 2 = 10 - u$ og $v' = 11 = 12 - 1 = 12 - v$, og vi har cosinus- og sinusbilde ved frekvens $u = 2$, $v = 1$, kan vi bruke symmetriegenskapene til å finne frekvensresponsen ved u', v'. Dette gjør vi ved å:

1. *Ta en elementvis multiplikasjon med bildet og cosinusbildet.*
   La resultatmatrisen bli kalt for C.
2. *Ta en elementvis multiplikasjon med bildet og sinusbildet.*
   La resultatmatrisen bli kalt for S.
3. *Summer elementene i C, og la den resulterende verdien være c.*
   Summer elementene i S, og la den resulterende verdien være s.
3. *Finn til slutt $c - js$, som er frekvensresponsen til innbildet ved frekvens u',v'.*

10. Filtrering i frekvensdomenet

Forklar hvordan man kan undersøke hva et filter i bildedomenet gjør med bildet i frekvensdomenet. Beskriv også hva vi kan analysere av hvordan filteret påvirker resultatbildet.

Løsningsforslag: Siden vi har fra konvolusjonsteoremet at sirkelkonvolusjon i bildedomenet tilsvarer elementvis multiplikasjon i frekvensdomenet, kan vi nullutvide filteret og ta en fouriertransformasjon av det. Fouriertransformasjonen av filteret gir oss informasjon om bl.a hvilke frekvenser bildet nuller ut. Det er også mulig å se om filteret kan gi opphav til ringing.

12. Normalfordeling

Vi har et bilde som består av en forgrunn og en bakgrunn. Vi får vite følgende om det normaliserte histogrammet til forgrunnspikslene og histogrammet til bakgrunnspikslene:

- For- og bakgrunnspikslene er normalfordelte.
- For- og bakgrunnspikslene har lik a priori sannsynlighet.
- Forgrunnspikslene har en middelverdi på 125 og 95% av pikslene ligger i intervallet [95, 155].
- 68% av bakgrunnspikslene ligger i intervallet [30, 50].

a) Forklar hva:

- – Standardavviket er til forgrunnspikslene
- – Standardavviket og middelverdien er til bakgrunnspikslene

Løsningsforslag: Siden histogrammene er normalfordelte, kan vi bruke "68-95-99"-regelen til å hente ut manglende informasjon om histogrammene.

– 95% av forgrunnspikslene ligger i intervallet $\mu \pm 2\sigma$, der μ er middelverdien til forgrunnspikslene og σ standardavviket til pikslene. Fra oppgaveteksten, er $\mu = 125$. Vi kan finne σ ved å enten løse $\mu - 2\sigma = 95$ eller $\mu + 2\sigma = 155$. Vi velger å løse sistnevnte ligning, men svaret skal også bli det samme om en løser den førstnevnte ligningen:

$$\mu + 2\sigma = 155$$
$$125 + 2\sigma = 155$$
$$2\sigma = 155 - 125$$
$$2\sigma = 30$$
$$\sigma = 15$$

Så standardavviket til bildet er 15.

– 68% av bakgrunnspikslene ligger i intervallet $\mu \pm \sigma$, der μ er middelverdien til forgrunnspikslene og σ standardavviket til pikslene. Vi får vite at $\mu - \sigma = 30$ og $\mu + \sigma = 50$. Vi har to ukjente, nemlig μ og σ, og to ligninger. Det er derfor mulig å løse ligningene for å finne μ og σ:

$$\mu - \sigma = 30$$
$$\mu = 30 + \sigma$$

$$\mu + \sigma = 50$$
$$30 + \sigma + \sigma = 50$$
$$2\sigma = 20$$
$$\sigma = 10$$

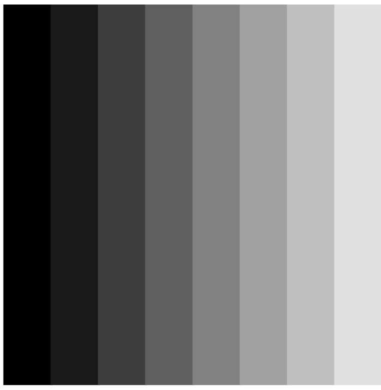
Med $\sigma = 10$ får vi at $\mu = 30 + \sigma = 30 + 10 = 40$.
Forgrunnspikslene har derfor middelverdi 40 og standardavvik 10.

b) Skissér histogrammene og forklar hvordan du har tenkt.

Løsningsforslag: Histogrammene skal bli to Gauss som er sentrert ved 125 og 40. Studenten skal kunne bruke den gitte informasjonen til å skissere omtrent klokke-formen til fordelingene og skal gjerne bruke verdiene som ble funnet i deloppgave a. Er ikke deloppgave a. besvart, skal de i allefall vise hvordan en normalfordeling ser ut og hvordan de tolker den gitte informasjonen om histogrammene.

13. Terskling av gråtonebilde

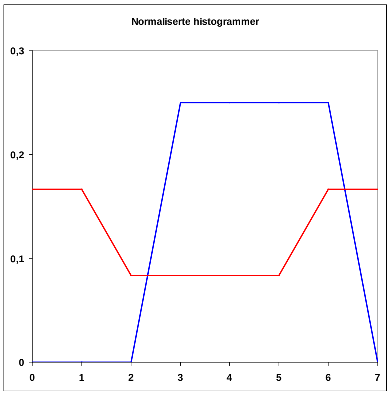
I et 3 biters gråtonebilde med størrelse 64 x 64 piksler består bakgrunnen av vertikale striper som er 8 piksler brede. Intensiteten i stripene øker fra venstre mot høyre, fra 0 til 7, som illustrert i figuren til høyre, som bare viser bakgrunnen i bildet.



På hver av de fire midterste stripene legges det objekter som utgjør 1/2 av stripens areal, og som er ett gråtonetrinn lysere enn bakgrunnen.

a) Skisser de normaliserte histogrammene til bakgrunnspikslene og objektpikslene i bildet.

Løsningsforslag: De to normaliserte histogrammene ser ut som i figuren til høyre.



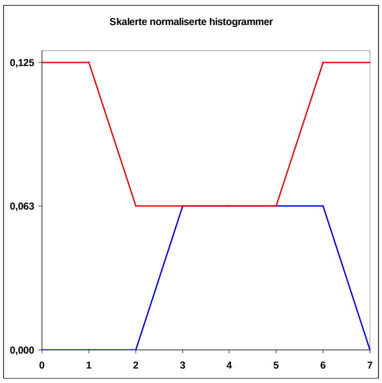
Det er 3 biter i hvert piksel, så skalaen går fra 0 til 7.

Bakgrunnspikslene i de to første og de to siste bakgrunnsstripene har samme sannsynlighet, de fire stripene i midten bare halvparten, og tilsammen skal disse sannsynlighetene være 1. pikselverdier.
Altså er det normaliserte bakgrunns-histogrammet
1/6, 1/6, 1/12, 1/12, 1/12, 1/12, 1/6, 1/6.

Og det normaliserte forgrunnshistogrammet er
0, 0, 0, ¼, ¼, ¼, ¼, 0.

b) Skisser de normaliserte histogrammene skalert med a priori sannsynlighet.

Løsningsforslag:Forgrunnspikslene utgjør ½ av hver av de fire midterste stripene, altså ¼ av bildet.



Bakgrunnspikslene utgjør da ¾ av bildet.

Skalert normalisert bakgrunnshistogram blir da
1/8, 1/8, 1/16, 1/16, 1/16, 1/16, 1/8, 1/8.

Skalert normalisert forgrunnshistogram blir:
0, 0, 0, 1/16, 1/16, 1/16, 1/16, 0.

c) Forklar hvorfor vi her ikke har noen entydig løsning på om vi må ha en eller to terskler for å få minst mulig total feil-segmentering ved terskling, og heller ikke en entydig løsning på hvor terskelverdien(e) ligger.

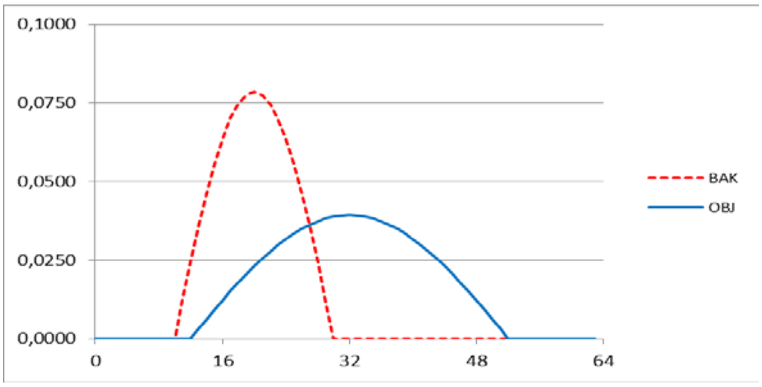
Løsningsforslag: De to skalerte fordelingene skjærer ikke hverandre, de faller bare sammen for tre gråtoneverdier. I beste fall vil ¼ av bildet feilklassifiseres for t =[3,4,5].

14. Segmentering ved terskling

Anta at de normaliserte gråtone-histogrammene til objekt- og bakgrunn i et analogt bilde er gitt ved

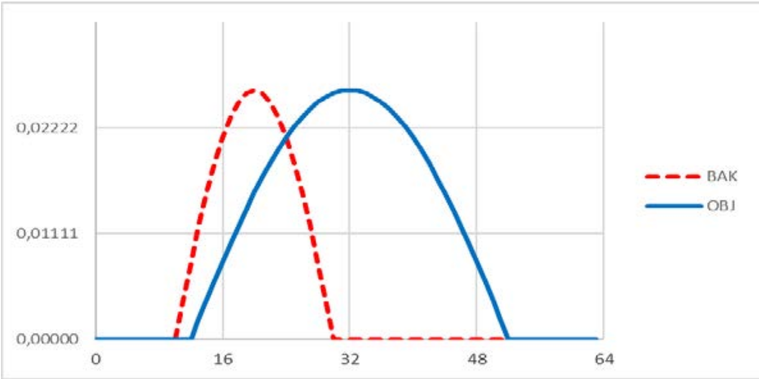
$$p(z) = \begin{cases} \dfrac{\pi}{4a}\cos\left[\dfrac{(z-\mu)\pi}{2a}\right] & \textit{for} \quad (\mu - a) \le z \le (\mu + a) \\ 0 & \textit{ellers} \end{cases}$$

Med parametrene a = 10, μ = 20 for bakgrunn og a = 20, μ = 32 for objekt ser histogrammene slik ut:



a) Med *a priori* sannsynlighet for bakgrunn B = 1/3, tegn en skisse som viser de to fordelingene skalert med a priori sannsynlighet. Forklar resonnementet!

Løsningsforslag: Her bør man fra skissen og de parametrene som er gitt se at bakgrunnen har lavere gråtoneverdier enn bakgrunnen. Bakgrunnsfordelingen skaleres altså med 1/3. Objektfordelingen skaleres med B = (1-O) = 2/3. De to fordelingene skalert med a priori sannsynlighet blir da like høye (figuren nedenfor er en del av løsningsforslaget).



b) Med den samme *a priori* sannsynlighet som er gitt i forrige del-oppgave, finn den terskelverdien T som vil gi minst total feil. Vis hvordan du gjør dette.

Løsningsforslag (ligningene nedenfor er en del av løsningsforslaget):

$$B \cdot P_B(T) = (1-B)\cdot P_O(T) \Rightarrow \frac{1}{3}\cdot\frac{\pi}{40}\cos\left[\frac{(T-20)\pi}{20}\right] = \left(1-\frac{1}{3}\right)\cdot\frac{\pi}{80}\cos\left[\frac{(T-32)\pi}{40}\right]$$

$$\Rightarrow \cos\left[\frac{(T-20)\pi}{20}\right] = \cos\left[\frac{(T-32)\pi}{40}\right] \Rightarrow \frac{(T-20)}{20} = \pm\frac{(T-32)}{40} \Rightarrow \begin{cases} 2(T-20) = (T-32) \\ 2(T-20) = -(T-32) \end{cases}$$

$$\Rightarrow T = \begin{cases} 8\ (umulig) \\ \underline{\underline{72/3 = 24.}} \end{cases}$$

Man kan også resonnere slik at løsningen ligger der de to a priori-skalerte fordelingene skjærer hverandre, som gitt i ligningen ovenfor. Siden de to fordelingene har samme parametrisering og er skalert til samme maksimumsverdi, men med bredde 20 og 40, må skjæringen ligge 1/3 av avstanden mellom de to middelverdiene 20 og 32,
*altså ved T = 20 + (32-20)*1/3 = 24.*

c) Anta at du terskler med den terskelverdien du nettopp har funnet.
Vis hvordan du kan finne <u>andelen feilklassifiserte bakgrunnspiksler.</u>
Hint: Den deriverte av sin(x) er cos(x), og sin(π/5) = 0.58778525…

Løsningsforslag: Her er hovedsaken at man beskriver hvordan dette skal gjøres, ikke at man beregner eksakt riktig verdi.
Integralet av bakgrunnsfordelingen fra terskelverdien til øvre ende av bakgrunnsfordelingen er (substituerer y = z – 20 for å forenkle integralet):

$$E = \int_{z=24}^{z=30} \frac{\pi}{40}\cos\left[\frac{(z-20)\pi}{20}\right]dz = \frac{\pi}{40}\int_{y=4}^{y=10}\cos\left[\frac{y\pi}{20}\right]dy$$

$$= \frac{\pi}{40}\left.\frac{\sin\left(\frac{y\pi}{20}\right)}{\frac{\pi}{20}}\right|_4^{10} = \frac{1}{2}\left(\sin\frac{\pi}{2} - \sin\frac{\pi}{5}\right) = \frac{1}{2}(1-0.587785...) \approx 0.20610$$

I innledningen står det at ligningen gir de normaliserte gråtone-histogrammene. Dette kan sjekkes for objekt-fordelingen ved å se på integralet av hele objektfordelingen, fra z = 10 til z = 30, dvs fra y = – 10 til y = 10, er A = ½[sin(π/2) – sin(-π/2)] = ½[1 + 1] = 1.
Men vi krever ikke at man gjør dette.
Andelen feilklassifiserte objektpiksler er <u>E/A ≈ 0,20610…</u>

15. Kompresjon og koding

Vi har et 4 bits, 9 x 9 piksels bilde med pikselverdier fra 1 til 9, som igjen er delt opp i 9 blokker á 3 x 3 piksler. I alle linjer, kolonner og blokker skal alle pikselverdiene være forskjellige fra hverandre, slik som i eksemplet i figuren.
(Hvis du synes dette minner om SuDoKu, har du helt rett!)

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

a) Beskriv trinnene i en Huffman-koding av dette bildet, finn kodeboken, og finn det gjennomsnittlige antall bits per piksel etter koding, og kompresjonsraten.

Løsningsforslag: Resultatet blir det samme om vi arbeider med histogrammet eller det normaliserte histogrammet. Her er alle pikselverdiene like sannsynlige.

1. *Sorter symbolene etter sannsynlighet, slik at de minst sannsynlige kommer sist. Her er alle pikselverdiene like sannsynlige.*
2. *Slå sammen de to minst sannsynlige symbolene til en gruppe, og sorter igjen.*
3. *Gjenta 2 til det bare er to grupper igjen.*
4. *Gi koden 0 til den ene gruppen og koden 1 til den andre.*
5. *Traverser innover i begge gruppene og legg til 0 og 1 bakerst i kodeordet til hver av de to undergruppene.*

Kodeboken blir da

(symbol	*histogram*	*kode):*
1	*9*	*001*
2	*9*	*0000*
3	*9*	*0001*
4	*9*	*110*
5	*9*	*111*
6	*9*	*100*
7	*9*	*101*
8	*9*	*010*
9	*9*	*011*

La b være antall biter per symbol i den ukomprimert datamengden (b=4), og c er gjennomsnittlig antall biter per symbol i den komprimerte datamengden:
$c = 9 \times (7 \times 3 + 2 \times 4) / 81 = (21 + 8) / 9 = \underline{3.222.}$ Da er $CR = b/c = 4 \times 9 / 29 \approx \underline{1.24.}$

b) Anta at vi bytter ut 9 x 9 gråtonebildet med et 16 x 16 «sudoku-bilde» med 16 gråtoner. Kan du da angi gjennomsnittlig antall bits per piksel etter Huffman-koding, uten å lage kodeboken eller utføre noen beregning?

Løsningsforslag: Dette trenger vi ikke kalkulator til. Det er 4 bits per piksel i bildet (like lange kodeord). Histogrammet er igjen helt flatt, men nå er sannsynlighetene inverse potenser av 2: $p=1/(2^4)$, så gjennomsnittlig antall bits per piksel etter Huffman-koding blir lik entropien, som er $H = -16*p*\log_2(p) = -16*(1/16)*\log_2(1/2^4) = -[\log_2(1) - \log_2(2^4)] = -[0 - 4] = 4$.
Så selv om Huffman-kodingen her er optimal, så oppnår vi ingen kompresjon.

c) Hvis vi har et 8 x 8 gråtonebilde der laveste verdi er 0 og høyeste verdi er 63, hver pikselverdi forekommer bare en gang, og alle rader, kolonner og diagonaler summerer til samme tall, (som i det magiske kvadratet nedenfor), vil vi da spare noe lagringsplass ved Huffman-koding?
Forklar! Hva blir kodingsredundansen her?

60	2	1	63	56	6	5	59
11	53	54	8	15	49	50	12
19	45	46	16	23	41	42	20
36	26	25	39	32	30	29	35
28	34	33	31	24	38	37	27
43	21	22	40	47	17	18	44
51	13	14	48	55	9	10	52
4	58	57	7	0	62	61	3

Løsningsforslag: Pikselverdiene i dette magiske kvadratet er 0, …, 63. De representeres ved like mange bit (6), siden $2^6 = 64$. Bruker vi Huffman-algoritmen på dette, får vi en kodebok som er lik naturlig binærkoding, eller en ekvivalent representasjon. Vi oppnår derfor ingen plassbesparelse, selv om vi bruker en entropi-basert metode, fordi histogrammet er helt flatt. Kodingsredundansen blir null, siden vi har et flatt histogram der alle kodeordene har 6 bit, og entropien = 6.

Takk for oppmerksomheten!