Løsningsforslag

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i :	INF2310 — Digital bildebehandling
Eksamensdag :	Tirsdag 25. mars 2014
Tid for eksamen :	15:00 – 19:00
Løsningsforslaget er på :	**10 sider**
Vedlegg :	**Ingen**
Tillatte hjelpemidler:	**Ingen**

- Det er 4 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 18 deloppgaver. **Hver deloppgave teller like mye.** Det lønner seg derfor å disponere tiden slik at man får besvart alle deloppgavene. Hvis du står fast på en deloppgave, gå videre slik at du får gitt et kort svar på alle deloppgavene.

- **Alle svar skal begrunnes.** Gjør rede for bruken av eventuelle teoremer, prinsipper eller forutsetninger slik at en tredjeperson kan følge dine resonnementer.

1. Interpolasjon og sampling

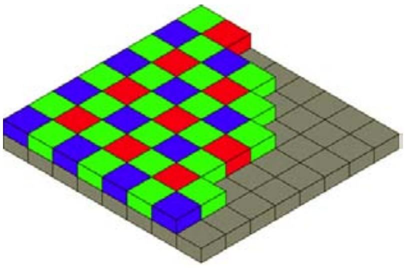
a) Gitt at pikselverdiene i fire nabopiksler i et gråtonebilde er f(0,0)=1, f(0,1)=3, f(1,0)=3, f(1,1)=9. Vi gjør bilineær interpolasjon for å finne en interpolert pikselverdi i punktet (x,y) = (0.3,0.75). Hvilken pikselverdi får vi? Vis hvordan du går fram.

Svar: Ved interpolasjon finner man f(0.3,0.75) =4.
Dette kan finnes på flere måter, alle hentet fra en forelesningsfoil:
Interpoler først i x-retning. Interpoler deretter i y-retning. Altså:

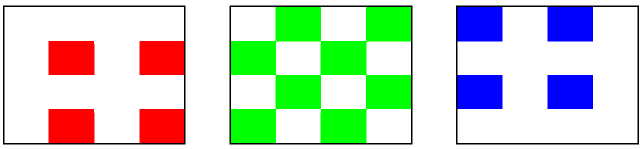
$$f(x,y) \approx (1-y)\,f(x,0) + y\,f(x,1)$$
$$der \quad f(x,0) \approx (1-x)\,f(0,0) + x\,f(1,0)$$
$$og \quad f(x,1) \approx (1-x)\,f(0,1) + x\,f(1,1)$$
$$\Rightarrow f(x,y) \approx (1-x)(1-y)\,f(0,0) + x(1-y)\,f(1,0) + (1-x)y\,f(0,1) + x\,y\,f(1,1)$$

Eller man kan interpolere i y-retning først, og deretter i x-retning.

b) I de fleste digitale kameraer sitter det en sensor der 50% av pikslene detekterer grønt lys, 25% blått og 25% rødt, i et mønster som vist på figuren til høyre (illustrasjon: en.wikipedia.org/wiki/Bayer_filter).



I et 4x4 piksel utsnitt vil altså lysintensiteten bli målt på litt forskjellige steder for hver av de tre fargene, som vist nedenfor, og slik at vi i utgangspunktet mangler informasjon om intensiteten i rødt, grønt eller blått på mange steder i bildet.



Anta at en gitt baklengs geometrisk transform forskyver pikslene et helt antall piksler i diagonal retning, og at bilineær interpolasjon mellom de kjente pikselverdiene benyttes til å finne nye pikselverdier.

Beskriv resultatet, gjerne som et 2D konvolusjons-filter anvendt på det opprinnelige bildet der dette er mulig, for hver av fargene rød, grønn og blå hvis den diagonale forskyvningen er 1 piksel.
(Du kan se bort fra problemer nær kanten av bildet).

Svar: For rød og blå:
I opprinnelige målte piksler: 1/4 [1 0 1 ; 0 0 0 ; 1 0 1]
Mellom to opprinnelig målte nabo-punkter horisontalt: 1/2 [0 1 0 ; 0 0 0 ; 0 1 0]
Mellom to opprinnelig målte nabo-punkter vertikalt: 1/2 [0 0 0 ; 1 0 1 ; 0 0 0]
Mellom fire opprinnelig målte nabo-punkter: [0 0 0 ; 0 1 0 ; 0 0 0]
Hvis vi antar at verdien i ikke-målte piksler er 0 så kan de fire konvolusjonsfiltrene over sammenfattes i filteret: 1/16 [1 2 1 ; 2 4 2 ; 1 2 1]

For grønn:
I opprinnelige målte piksler: [0 0 0 ; 0 1 0 ; 0 0 0]
Mellom fire opprinnelig målte nabo-punkter: 1/4 [0 1 0 ; 1 0 1 ; 0 1 0]
Hvis vi antar at verdien i ikke-målte piksler er 0 så kan de to konvolusjonsfiltrene over sammenfattes i filteret: 1/8 [0 1 0 ; 1 4 1 ; 0 1 0]
I alle konvolusjonsfiltrene over er origo i øverste venstre hjørne.

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

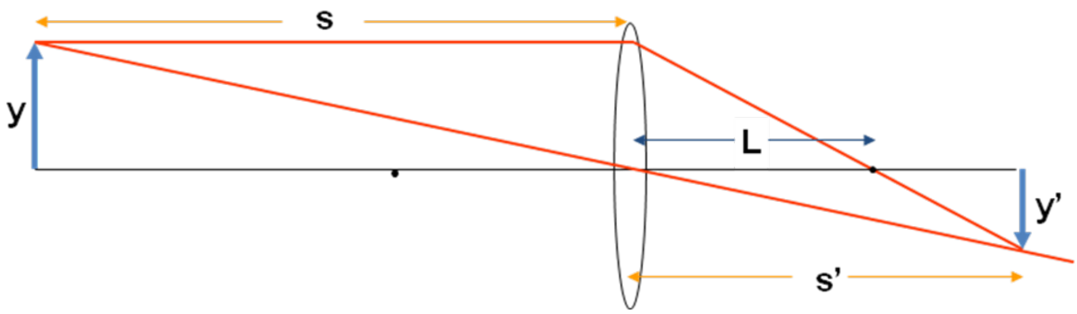
Svar: Romlig aliasing er en frekvensforvrengning som oppstår når man sampler med en lavere samplingsrate enn Nyquist-raten, dvs 2 ganger den høyeste romlige frekvensen som finnes i et bånd-begrenset bilde.

En aliasing-frekvens er en frekvens fa som oppstår eller styrkes i det samplede bildet p.g.a. aliasing og er gitt ved fa = fs – f, når f < fs < 2f, der fs 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.

d) Vi ser at fargebilder samples forskjellig i de tre fargene rødt, grønt og blått. Anta at vi har en periodisk struktur i horisontal eller vertikal retning i bildet, med en periode på P piksler. Hvor stor er den høyeste frekvensen, *f*, vi kan ha i bildeplanet, og fortsatt få en sampling av strukturen i henhold til samplingsteoremet i hhv rødt, grønt og blått?

Svar:Vi må ha minst to samplinger per periode. P må altså være på minst 4 piksler, dvs en frekvens som er lavere enn ¼ i rødt og blått, mens P må være på minst 2 piksler i grønt, altså en frekvens som er lavere enn 1/2.



e) Hvis vi bruker detektor-matrisen som er vist i deloppgave b) til å ta et bilde av et stakitt-gjerde med like brede hvite sprosser og svarte mellomrom, avbildet med en linse med fokallengde L = 20 mm fra en avstand s = 10 meter, og detektor-elementene (pikslene) er 50x50 mikrometer, hvor smale kan da stakitt-sprossene være (i cm) hvis vi skal få en avbildning med riktige farger? Bruk gjerne skissen ovenfor til å finne den sanne bredden av sprossene i stakittet, gitt s, L og bredden på stakitt-sprossene i bildet.

Svar: Vi trenger minst 2 piksler per periode i grønt for å dekke en sprosse og et mellomrom, mens vi trenger minst 4 piksler i rødt og blått, altså må en periode være minst 4*50=200 mikrometer i bildeplanet. Sprossene er da y’ = 100 mikrometer i bildeplanet. Sammenhengen mellom y og y’, gitt s og L, er gitt ved:

$$\frac{y'}{y} = \frac{s'}{s} \quad \Rightarrow \quad y = \frac{y's}{s'}$$

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{L} \Rightarrow s' = \frac{sL}{(s-L)}$$

$$\Rightarrow \quad y = \frac{y'(s-L)}{L} = 100 * 10^{-6} \frac{(10-0.02)}{0.02} * 100 \approx \underline{\underline{5cm}}$$

3

2. Histogramutjevning og gråtonetransformer

La oss anta at vi har følgende 4x5 gråtonebilde med en 3 biters gråtoneskala.

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

a) 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.

Finn det normaliserte histogrammet, og fra dette det normaliserte kumulative histogrammet c[i] (se tabellen under). Sett inn verdier i transform-arrayet
T[i] = Round((L-1)*c[i]+k) med L=4 og k=0 for i = 0, 1, …, G-1, der G=8.

i	h(i)	c(i)	T(i)
0	2	0.10	0
1	2	0.20	1
2	4	0.40	1
3	6	0.70	2
4	3	0.85	3
5	1	0.90	3
6	1	0.95	3
7	1	1.00	3

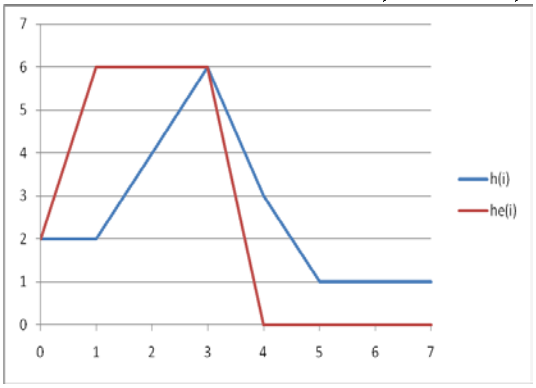
(Tabellen er en del av løsningsforslaget)

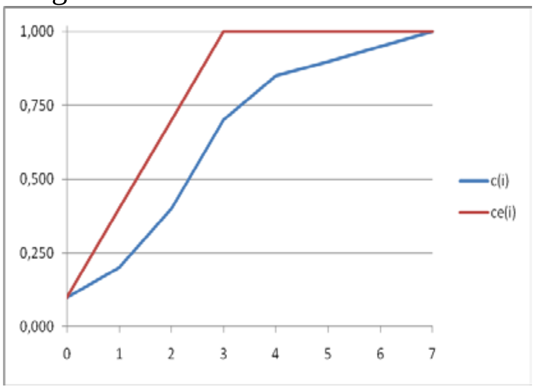
Gå deretter gjennom bildet piksel for piksel, og sett g(x,y) = T[i(x,y)]. Resultatet blir da

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

(Tabellen er en del av løsningsforslaget)

b) Vis histogrammet og det normaliserte kumulative histogrammet før og etter histogramutjevningen.
Hvorfor lyktes ikke histogramutjevningen i å produsere et ut-histogram som er helt flatt for gråtonenen 0 – 3? Er det mulig å oppnå dette med en annen histogramtransform? Begrunn svaret.



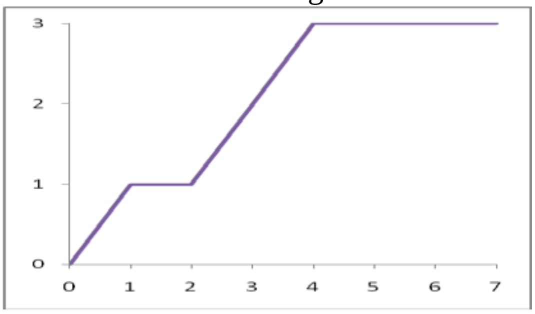


Det er 20 piksler i dette bildet, og vi skal ha 4 forskjellige gråtoner i ut-bildet. Ideelt sett skulle da hver gråtone i utbildet representeres med 5 piksler. Men siden vi ikke kan splitte histogramsøyler eller bytte om på dem, er ikke det mulig her, og med vår histogramutjevning ender vi opp med histogrammet [2 6 6 6 0 0 0 0] istedenfor [5 5 5 5 0 0 0 0].

c) Beskriv hvordan du kan modifisere histogrammet slik at det blir tilnærmet Gaussisk.

1. Finn s = T[i] som ovenfor og gjør histogramutjevning på innbildet.
2. Spesifiser det ønskede histogrammet g(z), som her er Gaussisk.
3. Finn den transformen T_g som histogramutjevner g(z) og den tilhørende inverstransformen T_g^{-1}.
4. Inverstransformer det histogramutjevnede bildet fra punkt 1 med $z = T_g^{-1}(s)$.

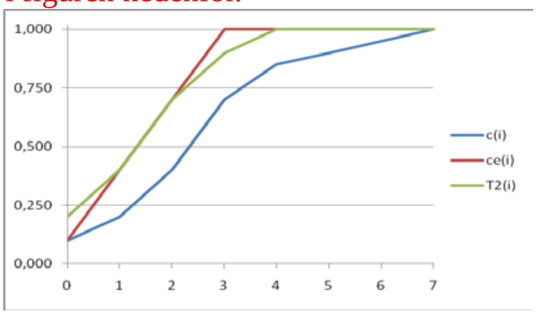
d) Kan histogramutjevningen ovenfor gjøres som en gråtonemapping fra input-intensitet til output-intensitet. Vis med figur.



e) Anta at du bruker en gråtonetransform gitt ved T(i) = floor(0.75*i) for i<6, og T(i)=4 ellers, på det opprinnelige bildet. Hvordan blir utbildet og det kumulative histogrammet da?

2	4	1	0	2
2	3	1	1	4
2	3	0	2	0
2	3	1	3	0

Oppgaven spør etter det kumulative histogrammet: [4 8 14 18 20], men det er også OK å gi det normaliserte kumulative histogrammet [0.2 0.4 0.7 0.9 1.0]}, som også er vist i figuren nedenfor.



3. Kantbevarende lavpassfiltrering

Vi har gitt følgende 3-biters gråtonebilde:

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

Når du i denne oppgaven skal beregne en filtrering, så trenger du bare å beregne responsen for pikslene der bildet og filteret overlapper i alle posisjoner, dvs. de pikslene der hele naboskapet til filteret ligger innenfor bildet når filterets origo er plassert i pikselen man ønsker å beregne responsen for.

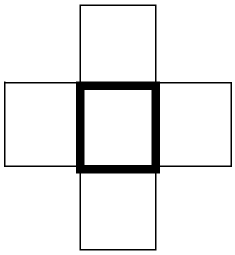
a) Beregn konvolusjonen av bildet og middelverdifilteret av størrelse 3x3.

Konvolusjonsresultatet blir:

6	6	4	3
4	4	3	2
3	3	2	2

når vi begrenser størrelsen av ut-bildet til posisjonene med full overlapp.

b) Beregn medianfiltreringen av bildet ved bruk av følgende naboskap:



Et pluss-formet naboskap.
Cellen med tykk kant angir filterets origo.

Filtreringsresultatet blir:

6	5	5	1
5	5	5	1
1	1	1	0

når vi begrenser størrelsen av ut-bildet til posisjonene med full overlapp.

c) Beskriv et lavpassfilter som kan sees på som et kompromiss mellom et middelverdifilter og et medianfilter.

Alfa-trimmet middelverdifilter tilbyr et direkte kompromiss der vi tar middelverdien av de mn-d midterste pikselverdiene (etter sortering) i mxn-naboskapet. d er en parameter til filteret og i grensetilfellene for d, d=0 og d=mn-1, så er det alfa-trimmede middelverdifilteret identisk som henholdsvis mxn-middelverdifilteret og mxn-medianfilter.
Man kunne selvsagt også brukt ikke-rektangulære naboskap i alle tre filtrene.

Ingen konvolusjonsfiltre vil konsekvent kutte ut ekstremeverdier og vil dermed ikke ha en oppførsel som kan minne om eller nærme seg et medianfilter.
K Nearest Neighbour-filteret, *K* Nearest Connected Neighbour-filteret, Sigma-filteret og symmetrisk nærmeste nabo-filteret baserer seg alle på å lete etter pikselverdier som er nær pikselverdien i pikselen vi skal finne responsen til (og så ta gjennomsnittet av disse «nære» pikslene). Slik særbehandling av pikselen vi skal finne responsen til har verken middelverdifilteret eller medianfilteret.
Andre lavpassfiltre som vi har sett på, som andre rang-filtre (enn medianfiltret), MinimalMeanSquareError-filteret, Max-homogenitet-filteret og Mode-filteret, er også vanskelig å argumentere for at kan sees på som et kompromiss mellom et middelverdifilter og et medianfilter.

4. Konvolusjon og Fourier-transformen

Merk: Selv om du ikke får til en deloppgave så vil det være mulig å løse neste.

Cosinus-bildet for frekvens (0,2) og frekvens (2,2) er for 4x4-bilder:

c_1 =

$\boxed{1}$	-1	1	-1
1	-1	1	-1
1	-1	1	-1
1	-1	1	-1

Cosinus-bildet
for frekvens (0,2)

c_2 =

$\boxed{1}$	-1	1	-1
-1	1	-1	1
1	-1	1	-1
-1	1	-1	1

Cosinus-bildet
for frekvens (2,2)

Cellen med tykk kant angir origo i bildet, dvs. der $x = y = 0$.

Ofte lar vi (0,0)-frekvensen, DC, være den midterste frekvensen. For 4x4-bilder vil det si at hver frekvenskomponent, u og v, er en verdi i mengden {-2, -1, 0, 1}. Med en slik definisjon så vil vi kalle de to cosinus-bildene over for cosinus-bildet for henholdsvis frekvens (0,-2) og frekvens (-2,-2).

De tilhørende sinus-bildene er lik 0-matrisen (matrisen med 0 i alle elementer) av størrelse 4x4.

a) 2D diskret Fourier-transformen (2D DFT-en) til hvert av cosinus-bildene over er hver sin reelle 4x4-matrise som er 0 i alle elementer bortsett fra ett. Hva er frekvensen til dette elementet for hvert av de to cosinus-bildene over og hva er verdien i dette elementet?

Vi har i forelesningen sett at 2D DFT-en av en MxN-samplet cosinus i 2D med heltallige frekvenskomponenter u og v i intervallet [-M/2, M/2) og [-N/2, N/2), henholdsvis, som er det et cosinus-bilde er, er en matrise som er 0 i alle elementer bortsett fra
- i (u,v) hvis u=-M/2 eller v=-N/2, der er verdien MN, eller
- i (u,v) og (-u,-v) hvis ikke, der er verdien MN/2 i begge elementene.
2D DFT-en til 4x4-cosinus-bildet med frekvens (0,-2) er altså bare ikke-0 i frekvens (0,-2) og her er verdien $MN = 4\cdot 4 = 16$.
Tilsvarende er 2D DFT-en til 4x4-cosinus-bildet med frekvens (-2,-2) bare ikke-0 i frekvens (-2,-2) og verdien er også her $MN = 4\cdot 4 = 16$.

Det er i forelesningene ikke blitt lagt vekt på hva verdien er i disse ikke-0-elementene, men det er enkelt å finne ved å beregne 2D DFT-koeffisienten for den aktuelle frekvensen, som i dette tilfellet blir å ta ett av cosinus-bildene over og punktmultiplisere med seg selv og så summere produktene. Man får da at man skal summere MN ett-tall, som er MN.

b) Anta at vi sirkelkonvolverer et 4x4-bilde med et av cosinus-bildene over. I filtreringen velger vi å beholde inn-bildet størrelse, altså har ut-bildet samme størrelse som inn-bildet.

Bruk konvolusjonsteoremet til å forklare at 2D DFT-en av ut-bildet vil:

1. Være en 4x4-matrise som er 0 i alle elementer bortsett fra i det elementet som 2D DFT-en av cosinus-bildet er ikke-0.
2. I sitt ikke-0-element være lik *k* ganger verdien i det samme elementet i 2D DFT-en av inn-bildet, der *k* er verdien i det samme elementet i 2D DFT-en av cosinus-bildet.

Du kan bruke det som står i deloppgave a.

Konvolusjonsteoremet sier at 2D DFT-en av hver av disse sirkelkonvolusjonene vil være punktproduktet mellom 2D DFT-en av inn-bildet og 2D DFT-en av det cosinus-bildet vi bruker i filtreringen. I deloppgave a står det at 2D DFT-en av hvert av de to cosinus-bildene er 0 i alle elementer bortsett fra ett. Dermed vil punktproduktet også være 0 i alle de samme elementene (0 ganger et vilkårlig tall er 0). Videre vil verdien i det ikke-0-elementet være produktet av verdien i det samme elementet i 2D DFT-en av innbildet og k=MN, som er verdien i det samme elementet i 2D DFT-en av cosinus-bildet.

c) Hvis vi tar følgende to Laplace-operatorer:

$$l_1 = \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & \boxed{4} & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} \qquad l_2 = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & \boxed{8} & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Cellen med tykk kant angir origo i filteret, dvs. der $x = y = 0$.

og nullutvider hver av dem til størrelse 4x4 og skifter («flytter») filteret slik at origo blir i øverste venstre hjørne, der vi i skiftingen antar sirkulær indeksering, så får vi følgende to konvolusjonsfiltre:

$$l_3 = \begin{array}{|c|c|c|c|}\hline \boxed{4} & -1 & 0 & -1 \\ \hline -1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 \\ \hline \end{array} \qquad l_4 = \begin{array}{|c|c|c|c|}\hline \boxed{8} & -1 & 0 & -1 \\ \hline -1 & -1 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 \\ \hline -1 & -1 & 0 & -1 \\ \hline \end{array}$$

Cellen med tykk kant angir origo i filteret, dvs. der $x = y = 0$.

Vi får akkurat samme ut-bildet dersom vi *sirkel*-konvolverer et bilde med l_1 eller med l_3, og på samme måte er også l_2 og l_4 ekvivalente.

Vis at 2D DFT-koeffisienten til l_3 er 4 i frekvens (0,-2) og 8 i frekvens (-2,-2). Vis også at 2D DFT-koeffisienten til l_4 er 12 i frekvens (0,-2) og 8 i frekvens (-2,-2).

Summen av punktproduktet av l_3 og cosinus-bildet for frekvens (0,-2) er 4. Summen av punktproduktet av l_3 og cosinus-bildet for frekvens (-2,-2) er 8. Summen av punktproduktet av l_4 og cosinus-bildet for frekvens (0,-2) er 12. Summen av punktproduktet av l_4 og cosinus-bildet for frekvens (-2,-2) er 8.

Siden sinus-bildet for frekvens (0,-2) og frekvens (-2,-2) er begge lik 0-matrisen, så er de fire punktproduktene mellom enten l_3 eller l_4 og et av disse to sinus-bildene lik 0-matrisen. Dermed er 2D DFT-koeffisientene for l_3 og l_4 for frekvens (0,-2) og (-2,-2) lik de respektive realdelene vi beregnet over.

d) Vis hva sirkelkonvolusjonen av l_3 og hvert av de to cosinus-bildene over er. Vis også hva sirkelkonvolusjonen av l_4 og hvert av de to cosinus-bildene over er, og forklar!

Du kan bruke det du skulle forklare i deloppgave b og vise i deloppgave c.

Du kan også bruke at 2D invers DFT er en lineær transformasjon, noe som blant annet innebærer at 2D invers DFT av et tall ganger 2D DFT-en av et bilde er lik det samme tallet ganget med det samme bildet.
Fra det vi skulle forklare i deloppgave b så vet vi at 2D DFT-en til hver av disse sirkelkonvolusjonene er 0 i alle elementer utenom elementet med samme frekvens som cosinus-bildet og i dette elementet er verdien lik verdien til 2D DFT-en av cosinus-bildet i samme element ganget med verdien til 2D DFT-en av bildet i samme element. Hvis vi kaller verdien til 2D DFT-en av bildet i samme element for *a* så betyr dette at 2D DFT-en til sirkelkonvolusjonen er *a* ganger 2D DFT-en av bildet, som ved bruk av at 2D invers DFT er en lineær transformasjon vil si at sirkelkonvolusjonen er *a* ganger bildet. Ved bruk av 2D DFT-koeffisientene vi skulle vise i deloppgave c får vi dermed at:

$$l_3 * c_1 = 4c_1$$
$$l_3 * c_2 = 8c_2$$
$$l_4 * c_1 = 12c_1$$
$$l_4 * c_2 = 8c_2$$

der * betegner sirkelkonvolusjon.

e) 2D DFT er en lineær transformasjon, noe som blant annet innebærer at 2D DFT-en av summen av to filtre er lik summen av 2D DFT-ene til hvert av filtrene.

Bruk dette og det du skulle vise i deloppgave c til å oppgi et 4x4-konvolusjons-filter som har en 2D DFT med lik verdi for frekvens (0,-2) og frekvens (-2,-2). Forklar! Oppgi hvor origo er i konvolusjonsfilteret.
Fra 2D DFT-koeffisientene vi skulle vise i deloppgave c så vet vi at 2D DFT-en til l_3 er 4 og 8 for henholdsvis frekvens (0,-2) og frekvens (-2,-2), og at 2D DFT-en til l_4 er 12 og 8 for de samme frekvensene (i samme rekkefølge). Filtreret som har en 2D DFT lik summen av disse to 2D DFT-ene vil derfor ha lik verdi, 16, for frekvens (0,-2) og frekvens (-2,-2). Siden 2D DFT er en lineær transformasjon vet vi at summen av l_3 og l_4 har denne 2D DFT-en, som er konvolusjonsfilteret:

$l_3 + l_4 =$

12	-2	0	-2
-2	-1	0	-1
0	0	0	0
-2	-1	0	-1

Cellen med tykk kant angir origo i filteret, dvs. der $x = y = 0$.

TAKK FOR OPPMERKSOMHETEN!