Løsningsforslag

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i :	INF2310 — Digital bildebehandling
Eksamensdag :	Onsdag 6. juni 2012
Tid for eksamen :	09:00 – 13:00
Løsningsforslaget er på :	**10 sider**
Vedlegg :	**Ingen**
Tillatte hjelpemidler:	**Ingen**

- Det er 8 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 20 delspørsmål. **Hvert delspørsmål teller like mye.** Det lønner seg derfor 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 resonnementer.

1. Histogramutjevning og gråtonetransform

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

3	7	2	0	6
3	4	2	2	3
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 2 til gråtone 5. Vis også resultatbildet.

Finn det normaliserte histogrammet, og fra dette det normaliserte kumulative histogrammet c[i] (se ovenfor). Sett inn verdier i transform-arrayet T[i] = Round((L-1)*c[i]+k) med L=4 og k=2 for i = 0, 1, …, G-1, der G=8.

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

(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

4	5	3	2	5
4	5	3	3	4
4	5	3	4	3
4	5	3	5	2

(Tabellen er en del av løsningsforslaget)

b) 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 $z = T_g^{-1}(s)$.

2. Konvolusjon

La intensiteten omkring pikselposisjonen (x,y) være modellert ved polynomet

$$f(x,y) = k_1+k_2x +k_3y+k_4x^2+k_5xy+k_6y^2$$

I et lokalt 3x3 område rundt posisjonen (x,y) vil da intensitetene være

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

a) Vis at filtermasken

$$\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

gir et skalert estimat av den korrekte lokale Laplace-verdien av denne modellen.

Den korrekte lokale Laplace-verdien av denne modellen er gitt ved

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

Den gitte filtermasken gir $-6(k_4+k_6)$, som altså er et skalert estimat.

b) Forklar hvordan vi på to måter kan gjøre Laplace-estimatet robust for støy.

- Vi kan konvolvere inn-bildet med et lavpassfilter før vi estimerer Laplace-verdien.
- Eller vi kan utnytte kommutativitets-egenskapen til konvolusjonsope-rasjonen, og konvolvere Laplaceoperatoren med et lavpassfilter, og så anvende resultatet som et filter på bildet.

3. Medianfiltrering

Et ”SWitching Median” (SWM) filter er en to-stegs prosedyre, der man først avgjør om et piksel er påvirket av støy ved å se om absoluttverdien av differansen mellom medianen av pikselverdiene i et naboskap og pikselverdien selv er større enn en gitt terskelverdi T. Hvis det er tilfelle, gis utbildet en pikselverdi lik den klassiske medianverdien innenfor naboskapet. Hvis ikke, bli pikselverdien i utbildet satt lik verdien i innbildet. Altså:

$$g(x,y) = \begin{cases} M_W(x,y) & hvis \ |M_W(x,y) - f(x,y)| > T \\ f(x,y) & ellers \end{cases}$$

der f(x,y) er innbildet, $M_W(x,y)$ er medianverdien innenfor et WxW vindu sentrert om posisjonen (x,y) i innbildet, T er en terskelverdi, og g(x,y) er utbildet.
Anta at innbildet er 5x5-bildet nedenfor, med pikselverdier mellom 0 og 7.

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

a) Finn pikselverdiene i det 3x3 utbildet du får ved å bruke et 3x3 median-vindu.
NB! Her er det ikke nok å bare gi et svar i form av pikselverdiene i utbildet.
Du må også vise hvordan du finner svarene.

4	4	2
4	4	3
3	5	4

(Tabellen er en del av løsningsforslaget)

b) Finn pikselverdiene i det 3x3 utbildet du får ved å bruke et 3x3 SWM-filter med terskelverdi T = 2.
NB! Her er det ikke nok å bare gi et svar i form av pikselverdiene i utbildet.
Du må også vise hvordan du finner svarene.

4	5	0
4	4	3
5	3	5

(Tabellen er en del av løsningsforslaget)

c) Beskriv fem vanlige måter å håndtere situasjonen når filteret ligger delvis utenfor innbildet, og du ønsker et ut-bilde g(x,y) som er like stort som inn-bildet f(x,y).

Når deler av filtret ligger utenfor inn-bildet,

1. sett g(x,y) = 0
2. sett g(x,y) = f(x,y)
3. trunker filter-masken
4. utvid inn-bildet ved ”reflected indexing”
5. utvid inn-bildet ved ” circular indexing”
6. utvid inn-bildet med 0-er
7. utvid inn-bildet med nærmeste pikselverdi

4. Frekvensdomenet

a) Hva forteller Fourier-spekteret til et gråtonebilde oss om bildet?
Kan vi generelt rekonstruere et bilde ved å bare bruke Fourier-spekteret?

Fourier-spekteret forteller oss hvilke frekvenser gråtonebildet inneholder, mer presist hvor betydningsfull hver frekvens er.
Nei, vi trenger generelt også Fourier-fasen for å rekonstruere et bilde. Faktisk er Fourier-fasen viktigere for rekonstruksjon enn Fourier-spekteret.

b) Beregn verdien til Fourier-spekteret for frekvens (1,1) til følgende 4x4-gråtonebilde:

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

Du kan få bruk for følgende matriser:

1	0	-1	0
0	-1	0	1
-1	0	1	0
0	1	0	-1

Cosinus-bildet av størrelse 4x4 og frekvens (1,1)

0	-1	0	1
-1	0	1	0
0	1	0	-1
1	0	-1	0

Sinus-bildet av størrelse 4x4 og frekvens (1,1)

(alle frekvenskomponenter i denne deloppgaven er nullindeksert)

Summen av punktproduktet av 4x4-bildet og cosinus-bildet er 4. Summen av punktproduktet av 4x4-bildet og sinus-bildet er -3. Dermed er F(1,1) = 4 - 3i der F er 2D DFT-en til 4x4-bildet. Verdien til Fourier-spekteret for frekvensen (1,1) er dermed magnituden til det komplekse tallet F(1,1), altså er:

$$|F(1,1)| = \sqrt{\mathrm{real}(F(1,1))^2 + \mathrm{imag}(F(1,1))^2} = \sqrt{4^2 + (-3)^2} = \sqrt{16+9} = 5$$

(Ligningen er en del av løsningsforslaget)

c) Vi har sett på tre ulike overgangstyper som kan brukes når vi designer filtre i frekvensdomenet, og vi har kalt de resulterende filtrene for enten ideelle filtre, Gaussiske filtre eller Butterworth filtre.
Redegjør kort for fordeler og ulemper ved hver av disse tre overgangstypene.

Ideelle overganger: Maksimalt raske overganger i Fourier-spekteret; går mellom 0 og 1 mellom naboelementer/nabofrekvenser.

+ Kan kontrollere presist hvilke frekvenser som skal bevares og hvilke frekvenser som skal fjernes; de som bevares blir helt bevart (altså ikke dempet), de som fjernes blir helt fjernet (altså ikke bare dempet).
– «Ringing»-effekt; den romlige representasjonen av et ideelt filter har «ringinger» og disse «ringingene» vil ofte være synlig når et slikt filter brukes til filtrering.

Gaussiske overganger: Relativt trege overganger mellom 0 og 1 i Fourier-spekteret (som en normalfordeling).

+ Ingen «ringing»; den romlige representasjonen har også Gaussiske overganger.
– Svært flytende overgang mellom hvilke frekvenser som bevares og hvilke som ikke bevares; de fleste frekvenser blir bare mer eller mindre dempet.

Butterworth overganger: En mellomting mellom ideelle overganger og Gaussiske overganger; for lav orden har Butterworth overganger tilsvarende egenskaper som Gaussiske overganger (ingen «ringing» ved orden 1), for høy orden har Butterworth overganger tilsvarende egenskaper som ideelle overganger.

+ Kan regulere (ved hjelp av ordenen) viktigheten av lite «ringing» mot god kontroll over hvilke frekvenser som bevares / ikke bevares.
– Kan gi både «ringing» og flytende overganger mellom hvilke frekvenser som bevares og hvilke som ikke bevares.

5. Kompresjon

a) Beskriv kort de tre ulike typene redundans som er aktuelle for kompresjon av *bilder*. Hvilken redundans fører generelt til ikke-tapsfri kompresjon?

Psykovisuell redundans: Informasjon som vi ikke ser.
Intersampel redundans: Nærliggende piksler ligner ofte på hverandre.
Kodingsredundans: Differansen mellom gjennomsnittlig kodelengde og entropien (den førsteordens Shannon entropi).
Det er kun den psykovisuelle redundansen som generelt sett fører til ikke-tapsfri kompresjon, intersampel redundans og kodingsredundans fører isolert sett alltid til tapsfri kompresjon.

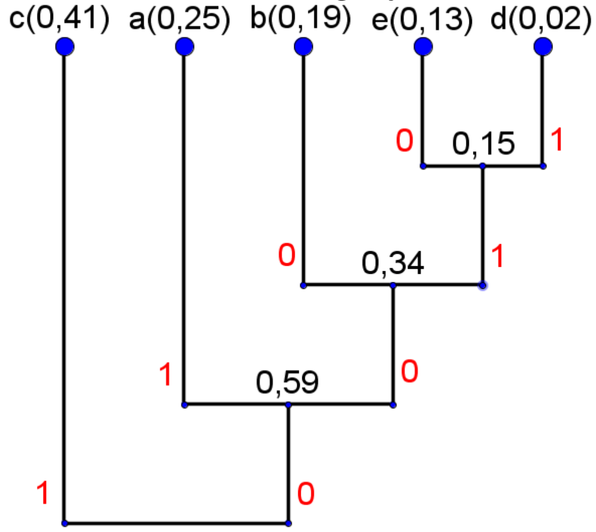
b) Finn en Huffman-kodebok for følgende sannsynlighetsmodell:

Symbol	a	b	c	d	e
Sannsynlighet	0.25	0.19	0.41	0.02	0.13

Vi sorterer symbolsannsynlighetene:

Symbol	c	a	b	e	D
Sannsynlighet	0.41	0.25	0.19	0.13	0.02

Huffman-sammenslåingene blir for denne modellen entydige. Hvis vi for hver forgrening tilordner 0 til den mest sannsynlige gruppen og 1 til den minst sannsynlige gruppen, får vi følgende Huffman-kodetre:



(Figuren er en del av løsningsforslaget)

Dette gir følgende Huffman-kodebok:

Symbol	a	b	c	d	e
Huffman-kodeord	01	000	1	0010	0011

c) Beskriv kort gangen i JPEG-komprimering, fra gråtonebilde til bitkode.

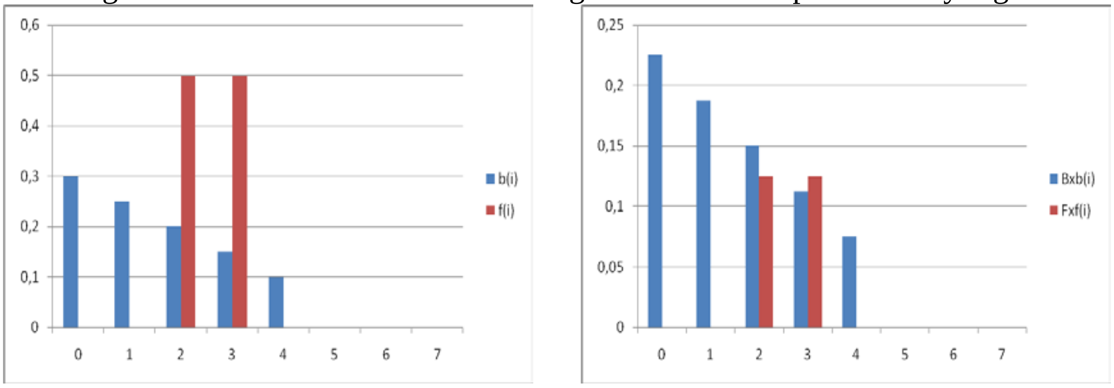
1) Gråtonebildet deles opp i 8x8-blokker, og for hver blokk gjøres følgende separat:
   i. Trekk 2^{b-1} fra hver pikselintensitet dersom intensitetene er gitt uten fortegn.
   ii. Transformer med 2D diskret cosinus-transform (2D DCT).
   iii. 2D DCT-koeffisientene skaleres med en vektmatrise og kvantifiseres til et heltall.
   iv. AC-komponentene, dvs. alle de skalerte og kvantifiserte 2D DCT-koeffisientene utenom DC-koeffisienten, sikk-sakk-skannes til en 1D-følge.
   v. 1D-følgen løpelengdetransformeres og kodes, enten med Huffman-koding eller med aritmetisk koding.
2) Den ene DC-koeffisienten fra hver blokk samles, differanse-transformeres og kodes, enten med Huffman-koding eller med aritmetisk koding.

6. Segmentering ved terskling

Anta at de normaliserte bakgrunns- og forgrunnsfordelingene av intensitetene i et 3 biters gråtonebilde er gitt ved følgende tabell:

i	b(i)	f(i)
0	0.3	0
1	0.25	0
2	0.2	0.5
3	0.15	0.5
4	0.1	0
5	0	0
6	0	0
7	0	0

a) Med a priori sannsynlighetene for hhv bakgrunn og forgrunn B=0.75 og F=0.25, tegn en skisse som viser de to fordelingene veiet med a priori sannsynlighet.



(Figurene er en del av løsningsforslaget)

b) Anta at du terskler bildet slik at du får minst mulig total klassifikasjonsfeil,

- Hvilke pikselverdier vil da bli klassifisert som forgrunn og bakgrunn?
- Hvor stor andel av bildet vil bli feilklassifisert?
- Hvor stor andel av forgrunnspikslene vil bli feilklassifisert?

  - Det er bare gråtone 3 som blir klassifisert som forgrunn, mens gråtonene 0, 1, 2, og 4 blir klassifisert som bakgrunn.
  - 0.15*0.75 = 0.1125 (11.25%) av bildet blir feilklassifisert som F. 0.5*0.25 = 0.125 (12.5%) av bildet blir feilklassifisert som B. Til sammen 11.25+12.5=23.75% av bildet blir feilklassifisert.
  - 0.125/0.25 = 50% av forgrunnspikslene blir feilklassifisert.

c) Anta at du terskler et vilkårlig bilde med gråtoneskala fra 0 til L-1 med en terskel T, men at du ønsker et utbilde med to *gråtoneverdier*, slik at den gjennomsnittlige gråtonen i utbildet og innbildet er den samme.
Finn og begrunn et uttrykk for gråtoneverdiene i utbildet, gitt innbildets normaliserte histogram p(i).
Middelverdien i innbildet og middelverdien av pikslene under og over terskelen T er gitt ved

$$\mu \equiv \sum_{i=0}^{L-1} i\, p(i) \quad , \quad \mu_B = \frac{\sum_{i=0}^{T} i\, p(i)}{\sum_{i=0}^{T} p(i)} \quad , \quad \mu_F = \frac{\sum_{i=T+1}^{L-1} i\, p(i)}{\sum_{i=T+1}^{L-1} p(i)}$$

Andelen av piksler i utbildet som tilhører hhv bakgrunn og forgrunn er

$$P_B = \sum_{i=0}^{T} p(i) \quad , \quad P_F = \sum_{i=T+1}^{L-1} p(i)$$

Hvis andelen P_B piksler får verdien μ_B og andelen P_F piksler får verdien μ_F etter terskling, så blir gjennomsnittsgråtonen i utbildet

$$\mu_T = P_B \frac{\sum_{i=0}^{T} i\, p(i)}{\sum_{i=0}^{T} p(i)} + P_F \frac{\sum_{i=T+1}^{L-1} i\, p(i)}{\sum_{i=T+1}^{L-1} p(i)} = \sum_{i=0}^{T} i\, p(i) + \sum_{i=T+1}^{L-1} i\, p(i) = \sum_{i=0}^{L-1} i\, p(i) = \mu$$

Hvis utbildet har de to gråtoneverdiene μ_B og μ_F, så vil inn- og utbildet ha samme middelverdi.

7. Farger og fargerom

a) Hvor mange forskjellige gråtoner kan det finnes i et 12 biters NxM fargebilde der hver av fargene R, G og B er representert med 4 biter?

Størrelsen på bildet har ingen betydning, bortsett fra hvis bildet er mindre enn antall mulige gråtoner.

For at vi skal ha en gråtone i RGB må vi ha (R,G,B) = (g, g, g). Så selv om det er 2^{12} = 4096 mulige verdier i et 12 biters fargebilde, så er det bare 2^4 = 16 mulige gråtoner i dette bildet.

b) Beskriv hvordan I-komponenten i IHS-systemet varierer over de tre ytterflatene av RGB-kuben som sees i figuren nedenfor, når du vet at de tre hjørnene som svarer til primærfargene R, G og B har RGB-verdier (1,0,0), (0,1,0) og (0,0,1).

Intensiteten varierer på samme måte over de tre sideflatene i RGB-kuben:
I hjørnet som svarer til en RGB primærfarge er intensiteten I=1/3: (1+0+0)/3, 0+1+0)/3, og (0+0+1)/3. I motstående hjørne finner vi hvit; I=(1+1+1)/3 =1. Mellom disse ytterpunktene varierer I lineært, slik at diagonalene mellom RGB sekundærfargene har I = 2/3.

8. Morfologi

La oss anta at vi har følgende 8x8 binære bilde:

0	1	0	0	0	1	0	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	0	0	1	1	1

der 1 angir forgrunnspiksel og 0 angir bakgrunnspiksel.

a) Finn erosjonen av bildet med strukturelementet:

0	1	0
1	$\boxed{1}$	1
0	1	0

Den den uthevde rammen angir origo.
Erosjonen er pikslene der strukturelementet passer i bildet (relativt til origo).
Vi får at erosjonen blir:

0	**0**	0	0	0	**0**	0	**0**
0	**0**	0	**0**	**0**	1	**0**	**0**
0	1	**0**	1	1	1	**0**	0
0	1	1	1	1	1	**0**	0
0	1	1	1	1	1	1	**0**
0	**0**	1	1	1	1	1	**0**
0	0	**0**	**0**	**0**	1	1	**0**
0	0	**0**	0	0	**0**	**0**	**0**

Der uthevningene markerer pikslene som endres ved erosjonen.

b) Hvordan kan man generelt finne kantene i et binært bilde basert på morfologi? Hvilket strukturelement vil man bruke for at kantene skal være sammenhengende med henholdsvis 4-naboskap og 8-naboskap?
Differansen mellom det binære bildet og erosjonen av det binære bildet og ett av to strukturelement vil resultere i et binært bilde av kantene til det opprinnelige binære bildet; $g = f - (f \ominus S)$ når f er det opprinnelige binære bildet, S er strukturelementet og g er erosjonsresultatet.
De to aktuelle strukturelementene (for todimensjonale bilder) er:

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

Gir sammenhengende kanter ved bruk av 4-naboskap

0	1	0
1	$\boxed{1}$	1
0	1	0

Gir sammenhengende kanter dersom man brukes 8-naboskap

TAKK FOR OPPMERKSOMHETEN!