UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i :	INF2310 — Digital bildebehandling
Eksamensdag :	Fredag 7. juni 2019
Tid for eksamen :	09:00 – 13:00 (4 timer)
Løsningsskissen er på :	**8 sider**
Vedlegg :	**Ingen**
Tillatte hjelpemidler:	**Godkjent kalkulator**

- Det er **6** 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 så fall rede for forutsetningene og antagelsene du gjør.

- Det er tilsammen **24** 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. Romlig oppløsning, Fourier, sampling og geometrisk transform

Punktspredningsfunksjonen til et avbildningssystem beskriver hvordan ett punkt i objektplanet (i "virkeligheten") vil fremstå etter avbildningen (altså i bildet vårt). Om vi antar linearitet ved avbildningen så kan vi si at det "sanne" objektet med uendelig detaljnivå blir konvolvert med denne punktspredningsfunksjonen for å danne bildet som senere blir digitalisert/samplet.

a) Hva sier konvolusjonsteoremet?

SVAR: En konvolusjon i bildedomenet tilsvarer en multiplikasjon i frekvensdomenet, og omvendt.

b) Med utgangspunktet i avbildningssystemet beskrevet i introduksjonen til denne oppgaven, hvordan kan vi ved å studere frekvensspekteret til punktspredningsfunksjonen si noe om den romlige oppløsningen til avbildningssystemet vårt?

SVAR: Ved konvolusjonsteoremet vil frekvensspekteret til bildet være en multiplikasjon av frekvensspekteret til punktspredningsfunksjonen og frekvens-spekteret til det vi avbilder. Altså vil en båndbegrensning av frekvens-spekteret til punktspredningsfunksjonen båndbegrense vårt dannede bilde. En øvre romlig frekvens er en måte å uttrykke et endelig detaljnivå (romlig oppløsning) i bildet.

c) Hvis vi får opplyst at vi er nødt til å sample (plukke pikselpunkter) med en avstand T_s < 5 mm for å unngå aliasing, hva sier det om hvor hovedvekten av våre ikke-ubetydelige frekvenskomponenter er å finne i punktspredningsfunksjonens frekvensspekter?

SVAR: Samplingsteoremet sier at vi er nødt til å ha T_s<1/2 T_{min}, der T_{min} er minste periode i bildet. Frekvenser av betydning høyere enn $1/T_{min}=f_{max}=1/10$ mm^{-1} må derfor ikke forekomme i bildet vi sampler. Ved oppgave b) er det da klart at frekvenser i frekvensspekteret til punktspredningsfunksjonen med betydelig amplitude alle bør være lavere enn f_{max}.

d) Hvis det blir benyttet en samplingsfrekvens som er 3 ganger Nyquistraten (den minste frekvensen vi må sample med for å unngå aliasing), beskriv og/eller skisser hva du vet om hvordan frekvensspekteret til bildene vil se ut. Du kan gjerne hoppe over dette med at spekteret egentlig er repetitivt, da dette ikke er det sentrale vi skal frem til her.

SVAR: [Hopper over dette med at spekteret egentlig er repetativt.] Nyquistraten er 2 x f_{max}, der f_{max} er høyeste romlige frekvens i bildet. (Alle frekvenser høyere enn f_{max} er «ubetydelige».) Vi har at f_s = 3 x Nyquistraten = 3x2xf_{max}. I randen av spekteret vårt har vi bidragene til frekvensene som er blant de høyeste det samplede bildet kan representere; disse frekvensene er ca. $f_s/2$ som er 3 x f_{max}. Altså vil 2/3 av alle de høyeste frekvensene i det samplede bildet være ubetydelige. Om vi fremviser frekvensspekteret som vi typisk har gjort i dette kurset, med nullfrekvensen midt i bildet, så vil det kunne være et rektangel i midten av bildet med ikke-ubetydelige frekvensbidrag, et rektangel som dekker 1/3 av bildet langs hver akse.

e) Anta at det etter sampling vil bli utført følgende geometriske operasjon:

$$x' = cx,$$
$$y' = cy,$$

der (x',y') er hvor koordinat (x,y) blir forflyttet, og hvor *c* er en konstant. La oss videre anta at det som neste steg blir utført en "vanlig" baklengs resampling (dog med «perfekt» interpolasjon). For hvilke *c* vil vi unngå aliasing om det i utgangspunktet hadde blitt benyttet en samplingsfrekvens på 3 ganger Nyquistraten?

SVAR: Den nye, effektive samplingsraten vil etter denne skaleringen være fsny=c*fs, der fs er den originale samplingsraten. Siden samplingsteoremet krever fsny=c*fs=c*3*Nyquistraten>Nyquistraten for å unngå aliasing, må derfor c>1/3.

2. En konvolusjons-operator

a) Gitt et 8-bits gråtonebilde f(x,y). Hva inneholder ut-bildet G(x,y) når

$$G(x,y) = \max_{k=0,\cdots,3}\left\{\left|g_k(x,y)\right|\right\} \quad , \text{ der } \quad g_k(x,y) = f(x,y)*h_k \ , k = 0, \ldots, 3$$

og $h_0, \ldots, h_3$ er konvolusjonsfiltrene

$$h_0 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, h_1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, h_3 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

Svar: Dette er en 4-retnings "kompass-operator" basert på Sobel-operatoren, produsert ved sirkulær rotasjon mot urviseren. For hvert piksel i bildet finner operatoren absolutt-verdien av den lokale gradient-magnituden i den av de 4 laveste 8-retningene der absolutt-verdien av gradientmagnituden er størst.
(Merk at vi trenger ikke å estimere gradientmagnituden i alle 8 retningene, siden fire av resultatene finnes ved å invertere resultatet fra de fire andre (motsatte) retningene.)

b) Hvilket trinn i Canny-algoritmen kan denne operatoren erstatte?

Svar: Den kan erstatte beregning av gradient-magnitude (som finnes med filtrene h_0 og h_2), og man slipper kvadrering-sum-kvadratrot.

c) Anta at vi også tar vare på den verdien av k som gir maksimal absoluttverdi av $g_k(x,y) = f(x,y)*h_k$, (k = 0,1,2,3), for hvert piksel i f(x,y), og lagrer denne i et ut-bilde θ(x,y).
Hvilket trinn i Canny-algoritmen kan dette «k-bildet» erstatte, og hvilke matematiske operasjoner unngår vi med dette?

Svar: «k-bildet» kan erstatte beregning av gradient-retning ved $\text{arctg}(g_2/g_0)$ med en simpel max-operator, der vi også slipper avrunding av resultatet av arctg til nærmeste 45 grader. Vi kommer altså enklere til tynningen av gradientmagnitude-bildet i eller mot den 8-retningen der gradienten er størst.

3. Diskret Fourier-transform

a) Anta at vi har et NxN bilde, f, og at vi har gjort en diskret Fourier transform (i praksis en FFT) av dette bildet og lagret resultatet i NxN-matrisen F, der F(0,0) gir nullfrekvensen («DC komponenten»). Hvis nå alle elementene i F er null bortsett fra F(5,4) og F(N-5, N-4), beskriv hvordan bildet ser ut.

SVAR: Et rent «bølgebilde» hvor vi i den ene retningen har 5 repetisjoner av en ren sinusbølge og i den andre retningen 4 repetisjoner. Fasen er ukjent.

b) Anta samme type matrise F som beskrevet i deloppgave a), men for et generelt bilde f. Gi pseudokode for hvordan man kan hente ut amplituden og fasen til frekvens (u, v), der u er antall repetisjoner vertikalt og v antall repetisjoner horisontalt.

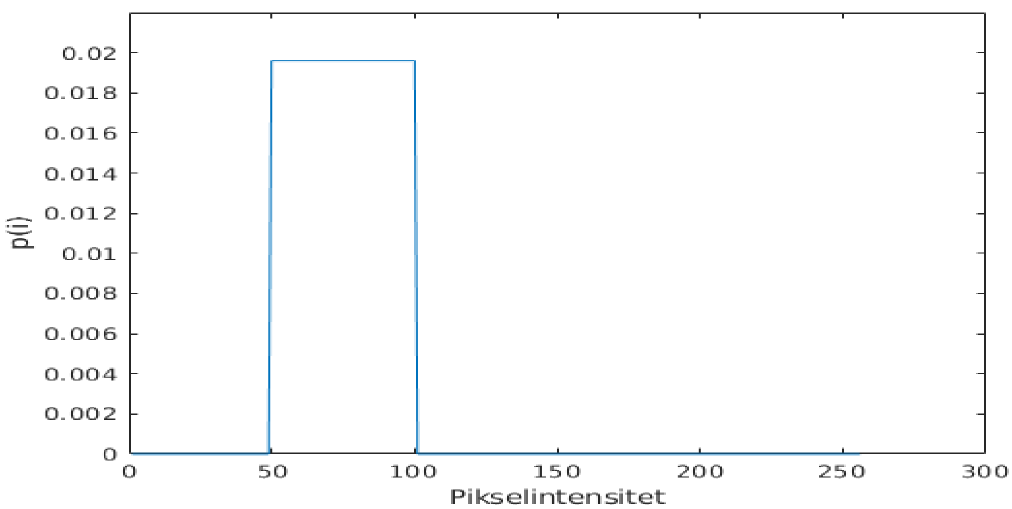
SVAR: amplitude = abs(F(u,v)); fase = atan2(imag(F(u,v)), real(F(u,v)));

c) Anta så at vi translaterer (forflytter) bildet, f, 10 piksler til høyre og 5 piksler nedover, men på en slik måte at piksler som kommer utenfor bildematrisen kommer inn igjen på motsatt side (det samme som om bildet var repeterende langs begge akser).
Hvordan ville en slik forflytning påvirke amplitudespekteret?

SVAR: Ved skiftteoremet så vil en slik forflytning kun endre frekvenskomponentenes fase. Altså vil amplitudespekteret forbli uendret.

4. Histogramutjevning og histogramtilpasning

Ved histogramutjevning av gråtonebilder benyttes en global gråtonetransform i et forsøk på å gi et resultatbilde med mest mulig flatt gråtonehistogram.
Slike globale gråtonetransformer er gitt ved en funksjon T[i], for pikselintensiteter i.
Ved histogramtilpasning mener vi her at vi åpner opp for å benytte slike gråtone-transformer i et forsøk på gi resultatbildet et histogram som ligner på et hvilket som helst ønsket resultathistogram.



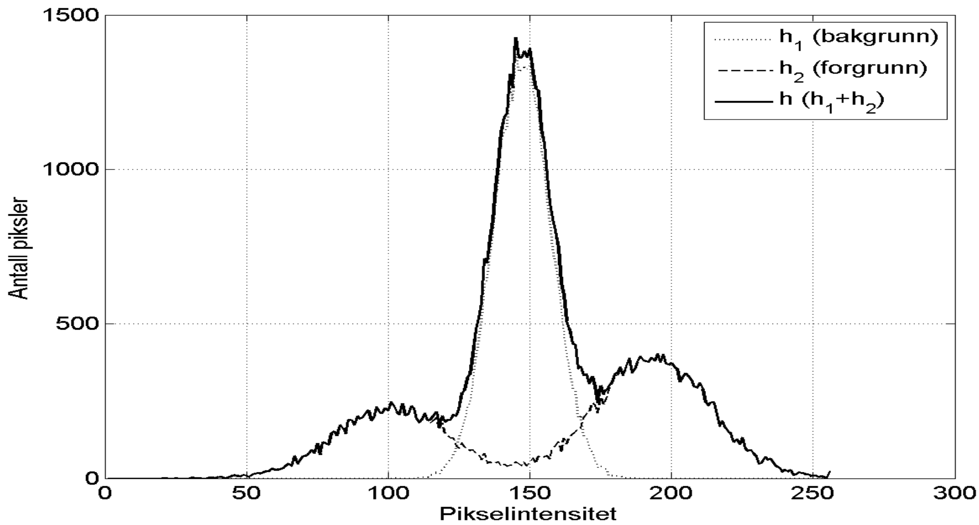
a) La oss anta at vi har et bilde med et normalisert gråtonehistogram som vist i figuren over. Bildet har heltallige gråtoneverdier fra 0 til 255.
Beskriv og skisser gråtonetransformen, T, du ville benyttet for å utføre en histogramutjevning på dette bildet.

SVAR: Vi vet at T er en skalert versjon av bildets kumulative histogram. Her vil altså T være 0 frem til i=50, deretter stige lineært til i=100 hvor T videre vil ha en verdi på 255.

b) Anta at image er en bildematrise med h som gråtonehistogram, og at vi har følgende (selvforklarende) funksjoner tilgjengelig:
T = getHistogramEqTransform(h),
T_{inv} = getInverseTransform(T),
im_{res} = transformImage(image, T).
Skisser (pseudokode) hvordan du kan benytte disse funksjonene til å utføre en histogramtilpasning til et ønsket histogram $h_{ønsket}$.

SVAR: (Rett etter forelesningsnotatene s. 13, 20190213)
T = getHistogramEqTransform(h)
imEq = transformImage(image,T)
Tg = getHistogramEqTransform(h_ønsket)
Tg_inv = getInverseTransform(Tg)
im = transformImage(imEq, Tg_inv)

5. Segmentering ved terskling



I figuren over vises histogrammet, h(i), der i er pikselintensitet, til et bilde som består av to klasser: forgrunn og bakgrunn. Videre har vi tegnet inn histogrammene til disse to klassene, henholdsvis h_1 og h_2. Vi har altså $h=h_1 +h_2$. Vårt mål her er å benytte pikslenes gråtoneverdier til å bestemme om en piksel skal settes til forgrunn eller bakgrunn (klassifisere).

a) Basert på h, h_1 og h_2 i figuren over, hvilke pikselintensitetsintervaller ville du klassifisert til bakgrunn og hvilke intervaller ville du klassifisert til forgrunn? (Som alltid, forklar og begrunn svaret ditt.)

SVAR: Minimerer antall feilklassifiserte piksler ved å velge forgrunn for i der h2(i)>h1(i) (eventuelt bakgrunn for i der h1(i)>h2(i)).

b) Ved en slik intervallinndeling som du kom frem til i deloppgave a), benytt h_1 og h_2 til å gi et matematisk uttrykk for antall *riktig* klassifiserte piksler.

SVAR: \sum_i = \max(h_1(i),h_2(i))

c) For bildet som omtales i denne oppgavens introduksjonstekst, anta at vi ikke har h_1 og h_2 tilgjengelig, kun bildet og dets histogram, h.
Ville Ridler og Calvards algoritme (jfr. k-means) for automatisk terskling trolig gi et tilfredstillende resultat på dette bildet?

SVAR: Nei. Histogrammene for forgrunn og bakgrunn tilfredsstiller ikke på nær antagelsene bak den nevnte algoritmen; som strengt tatt er at hver klasse er normal-fordelt, og med tilnærmet lik varians og a-priori sannsynlighet.

d) Om det samlede objekt-areal vi vil segmentere ut enten er veldig stort eller veldig lite i forhold til den totale bildeflaten vil vi fort få utfordringer ved automatisk terskling a la Ridler og Calvards metode. Om vi får markert et sett med piksler som befinner seg nær randen av objektene vi er interessert i, hvordan kan vi benytte denne informasjonen til å finne bedre terskelverdier?

SVAR: Om vi kun tar utgangspunkt i piksler liggende i nærheten av våre oppgitte kantpiksler, vil vi trolig få et mer balansert forhold mellom bakgrunns- og forgrunnspiksler, og på den måten komme nærmere antagelsene bak metoder *a la* Ridler og Calvards.

e) Ved bl.a. Otsus metode for automatisk terskling kan vi få ut et kvantitativt mål, for eksempel et tall mellom 0 og 1, på hvor godt vår tilpassede modell skiller våre klasser, for eksempel mellom forgrunn og bakgrunn. Beskriv hvorfor og hvordan dette kan komme til nytte ved lokal, adaptiv terskling.

SVAR: Ved lokal, adaptiv terskling beregnes terskelverdien basert på data fra lokale vinduer. Det kan fort være at disse lokale vinduene ikke inneholder piksler fra begge klasser. Terskler fra slike vinduer bør ignoreres, og deteksjonen for dette, eller vektingen av den resulterende terskelen, kan baseres på det nevnte separasjonsmålet.

6. Kompresjon og koding

Gitt et 8 x 8 piksels utsnitt av et 3 bits gråtonebilde med pikselverdier:

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

a) Finn bildets gråtonehistogram, normaliserte histogram og kumulative normaliserte histogram.

Svar: Histogrammet: [2, 2, 4, 8, 16, 16, 8, 8], som summerer til 64,
normalisert histogram: 1/64[2, 2, 4, 8, 16, 16, 8, 8],
Kumulativt normalisert: 1/64[2, 4, 8, 16, 32, 48, 56, 64].

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

Svar: Resultatet blir det samme om vi arbeider med histogrammet eller det normaliserte histogrammet.

1. Sorter symbolene etter sannsynlighet, slik at de minst sannsynlige kommer sist.
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):
4	16	10
5	16	11
3	8	010
6	8	011
7	8	000
2	4	0010
0	2	00110
1	2	00111

La *b* være antall biter per symbol i den ukomprimert datamengden (b=3),
og *c* er gjennomsnittlig antall biter per symbol i den komprimerte datamengden:
c = (32*2 + 24*3 + 4*4 + 4*5)/64 = 172/64 = 2.6875.
Da er CR = b/c ≈ 1.12.

c) Hva er forholdet mellom det gjennomsnittlige antall bits per piksel etter Huffman-koding og entropien til bildeutsnittet, når vi ser bort fra størrelsen på kodeboken? Forklar og begrunn!

Svar: Dette trenger vi ikke regne ut. Forholdet er 1:1, fordi alle sannsynlighetene kan skrives som inverse toer-potenser.

d) Forklar hvordan du vil gjøre en histogram-utjevning av det gitte gråtonebildet til et ut-bilde med færre gråtoneverdier, og hvordan dette generelt vil påvirke Huffman-kodingen av bildet?

Svar: Transformen er gitt som en skalert versjon av bildets kumulative histogram c[i].
I lærebok/forelesning er to alternative ligninger gitt:
T(i) = Round[(L-1)*c(i)], og **T(i) = Ceiling[L*c(i) - 1]**, der L er det nye antall gråtoner.
Hvilken ligning som brukes, er ikke vesentlig her, selv om de kan gi litt forskjellige resultat.

Generelt vil den histogramtransformen som brukes flytte rundt på hele histogramsøyler, uten å splitte dem eller endre rekkefølgen på dem, men den kan slå sammen histogramsøyler.
Ved sammenslåing av histogramsøyler vil vi generelt få en annen kodebok, og en høyere kompresjonsrate enn med det opprinnelige antallet gråtoner.

e) Beskriv stegene i en «lossy» JPEG-kompresjon av bare dette utsnittet.

Svar:
1. Hver blokk transformeres med 2D DCT (diskret cosinus-transform).
   Mye av informasjonen i de 64 pikslene samles dermed i en liten del av de 64 2D DCT-koeffisientene; nemlig de i øverste, venstre hjørne.
2. 2D DCT-koeffisientene punktdivideres med en vektmatrise og avrundes til heltall.
3. DC- og AC-elementene behandles nå separat.
   AC-elementene sikk-sakk-skannes slik at elementene ordnes i en 1D-følge, som løpelengdetransformeres. Løpelengdeparene Huffman- eller aritmetisk kodes.

f) Hvis et større bilde ble delt opp i flere slike 8 x 8 utsnitt, hvilke artefakter ville da kunne oppstå etter kompresjon/dekompresjon, og hvorfor?

Svar: Ved kompresjon/dekompresjon vil det forekomme små avvik fra den originale blokken, siden vi gjør avrundinger til nærmeste heltall, dette kan gjøre at vi ser at bildet har vært blokk-inndelt.

Takk for oppmerksomheten!