Løsningsforslag

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i :	INF2310 — Digital bildebehandling
Eksamensdag :	Tirsdag 4. juni 2013
Tid for eksamen :	09:00 – 13:00
Løsningsforslaget er på :	**13 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 19 deloppgaver. **Hvert deloppgaver teller like mye.** Det lønner seg derfor å disponere tiden slik at man får besvart alle oppgavene. Hvis du står fast på en deloppgave, gå videre slik at du får gitt et kort svar på alle oppgavene.

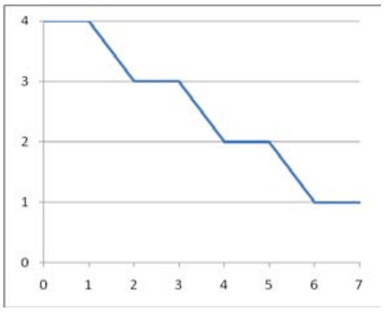
- **Alle svar skal begrunnes.** Gjør rede for bruken av eventuelle teoremer, prinsipper eller forutsetninger slik at en tredjeperson kan følge dine resonnementer.

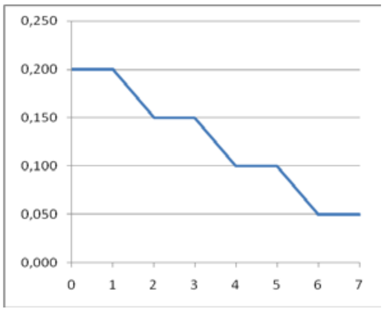
1. Histogramtransform

Anta at vi har følgende 4 x 5 piksels gråtonebilde med en 3 biters gråtoneskala:

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

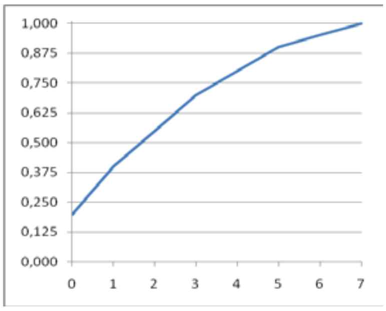
a) Skisser histogrammet, h(i), til bildet h, og det normaliserte histogrammet, p(i).





b) Forklar hvordan vi finner gråtonetransformen som histogramutjevner et bilde som har normalisert histogram p(i), og hvordan histogramutjevningen utføres. Vis resultatet av histogramutjevningen av bildet ovenfor.

Finn det normaliserte kumulative histogrammet c(i), se figur til høyre og tabell nedenfor.
Sett inn i transform-arrayet
T[i] = Round[7*c(i)],
se tabell nedenfor.
Dette gir et nytt bilde, vist nedenfor.



i	h(i)	p(i)	c(i)	T[i]
0	4	0.2	0.2	1
1	4	0.2	0.4	3
2	3	0,15	0.55	4
3	3	0.15	0.7	5
4	2	0.1	0.8	6
5	2	0.1	0.9	6
6	1	0.05	0.95	7
7	1	0.05	1.0	7

6	6	5	4	3
5	4	6	5	4
1	3	3	7	6
1	1	1	3	7

c) Vil en histogramutjevning påvirke entropien til et gråtonebilde ? Forklar!

Intuitivt skulle man kanskje tro at entropien øker, siden målet med histogramutjevning er å få et (tilnærmet) uniformt histogram, som jo har maksimal entropi.
Men entropien vil bli lik eller mindre enn entropien før histogramutjevningen.

Histogramutjevning vil flytte eller slår sammen søyler i histogrammet. Flytting endrer ikke på bidraget til entropien fra hver histogramsøyle. Så hvis histogramutjevningen bare medfører flytting av histogramsøyler, så blir det ingen endring i entropien.

Men sammenslåing av søyler vil minske entropien, siden $-(p_1+p_2)\log_2(p_1+p_2) < -p_1\log_2(p_1)-p_2\log_2(p_2)$, fordi $\log_2(p)$ er ikke-lineær.

Det er ganske enkelt å vise:
Entropibidraget til en sammenslått søyle er:
$-(p_1+p_2)\log_2(p_1+p_2) = -p_1\log_2(p_1+p_2) - p_2\log_2(p_1+p_2)$
mens entropibidraget til de to opprinnelige søylene er:
$-p_1\log_2(p_1) - p_2\log_2(p_2)$
Siden p_1 og p_2 er positive vil $\log_2(p_1+p_2) > \log(p_1)$ og $\log_2(p_1+p_2) > \log(p_2)$,
og dermed får vi:
$\underline{-p_1\log_2(p_1+p_2) - p_2\log_2(p_1+p_2) < -p_1\log_2(p_1) - p_2\log_2(p_2)}$

2. Lavpassfiltrering

I denne oppgaven skal du lavpassfiltrere følgende 5x5-bilde:

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

Når vi ber deg utføre en filtrering, så trenger du bare å beregne responsen for pikslene der hele filteret overlapper med bildet.

a) Median-filtrer bildet ved bruk av et sentrert 3x3-naboskap.
Vis hvordan du går frem for å finne responsene!

Filtreringsresultatet er:

1	2	2
2	3	5
3	5	6

(Tabellen er en del av løsningsforslaget.)

b) I *K Nearest Neighbour*-filtrering er responsen i piksel (x,y) gitt som middelverdien av de *K* pikslene i naboskapet rundt (x,y) som «ligner mest» på (x,y) i pikselverdi, der vi med «ligner mest» mener at absoluttverdien av differansen er minst mulig. Merk at (x,y) kan være sin egen nabo.

Gjør en K Nearest Neighbour-filtrering av bildet ved bruk av et sentrert 3x3-naboskap og K=4. Vis hvordan du går frem for å finne responsene!

Filtreringsresultatet er:

2	2	1,5
1	6	5,5
2,5	5,5	7

(Tabellen er en del av løsningsforslaget.)

c) Generelt sett, hvordan påvirker parameteren K resultatet av en K Nearest Neighbour-filtrering?

Lavere K vil fjerne mindre støy. Med høyere K risikerer vi å glatte kanter. Mer presist, hvis vi bruker et n×n-naboskap der n=2a+1, vil:

- K=1: gi ingen effekt
- K≤n: bevare tynne linjer (for tynne linjestykker vil endene kunne bli påvirket selv om K≤n)
- K≤$(a+1)^2$: bevare hjørner
- K≤(a+1)n: bevare rette kanter

3. Fourier-transform

a) Hvilken frekvens har følgende 4x8-cosinus-bilde? Forklar!

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

Angi frekvensen ved bruk av nullindeksering,
d.v.s. slik at frekvensen til DC-komponenten er (0,0).

Bildet inneholder to hele perioder horisontalt, dermed er horisontal frekvens 2. Bildet er konstant vertikalt, dermed er vertikal frekvens 0.
Cosinus-bildet har dermed frekvens (0,2).

b) Beregn koeffisienten til den todimensjonale diskrete Fourier-transformen (DFT) av følgende bilde for en frekvens (u,v) der $0 \leq u \leq 3$ og $0 \leq v \leq 7$:

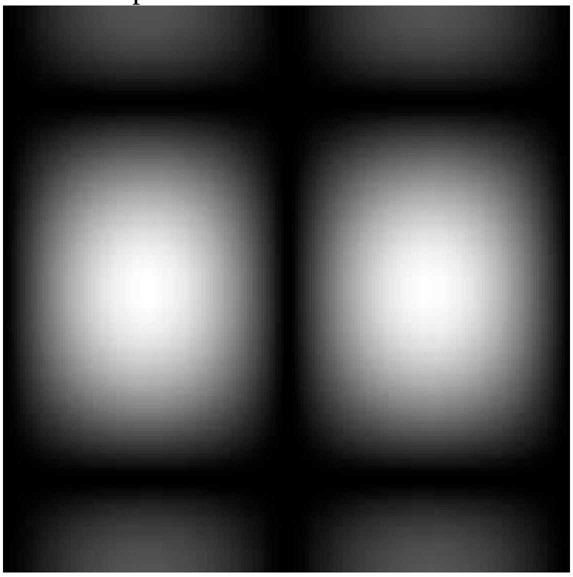
3	2	3	3	1	2	2	2
3	0	3	1	3	0	3	0
0	1	0	3	3	3	2	2
3	2	3	0	3	3	1	0

Alternativ 1: DC-komponenten til 2D DFT-en er summen av alle pikselverdiene i bildet, som her er 60.
Alternativ 2: Vi kan danne 4x8-sinus-bildet av samme frekvens som det oppgitte cosinus-bildet ved å huske at sinus-funksjonen starter i 0 (sin(0)=0) og sinus-bildet blir først negativt og deretter positivt. Altså er det tilhørende sinus-bildet:

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

(Tabellen er en del av løsningsforslaget.)
Summen av punktproduktet av bildet og cosinus-bildet er 2.
Summen av punktproduktet av bildet og sinus-bildet er -2.
Altså er 2D DFT-koeffisienten til frekvens (0,2): 2-2i
Alle andre alternativer vil også gi full uttelling dersom resonnementet og utregningen er korrekt.

c) Bildet under viser Fourier-spekteret til et 3x3-konvolusjonsfilter etter nullutvidelse til 200x200 for fremvisning. Fourier-spekteret er vist slik at DC-, eller (0,0)-frekvensen, ligger midt i bildet. 0 er visualisert som svart og maksimalverdien i Fourier-spekteret er visualisert som hvit.



Hvilket kjent 3x3-konvolusjonsfilter har dette Fourier-spekteret? Forklar!
Hva vil dette filteret gjøre med et bilde dersom bildet konvolveres med filteret?

DC-komponenten og verdien til Fourier-spekteret i den største horisontale frekvensen er 0, og mellom disse to punktene stiger verdien opp mot et maksimum midt mellom dem. Filteret vil altså høypassfiltrere i horisontal retning, og fra lokaliseringen av min og max i Fourier-spekteret skjønner vi at filteret baserer seg på den horisontale gradientkomponent-estimatoren i den symmetriske 1D-gradientoperatoren; [1 0 -1].

I vertikal retning er maksimumet på den horisontale aksen, og verdien til Fourier-spekteret avtar derfra opp til 2/3 av største vertikal frekvens. I 2/3 av største vertikal frekvens er Fourier-spekterets verdi 0, og videre opp mot største vertikale frekvens øker Fourier-spekterets verdi. Filteret vil altså lavpassfiltrere i vertikal retning, og fra ringingen skjønner vi at filteret baserer seg på 3x1-middelverdifiltreret; $[1\ 1\ 1]^T$.

For å oppnå det avbildede Fourier-spekteret kan vi punktmultiplisere Fourier-spekteret til [1 0 -1] og $[1\ 1\ 1]^T$ etter nullutvidelse. Fra konvolusjonsteoremet vet vi at punktmultiplikasjon i Fourier-domenet tilsvarer konvolusjon i bildedomenet (egentlig sirkelkonvolusjon, men p.g.a. nullutvidelsen av konvolusjonsfiltrene før 2D DFT-beregning vil det effektivt bli som konvolusjon der vi bruker nullutvidelse), hvilket betyr at 3x3-konvollusjons-filteret er den horisontale gradientkomponent-estimatoren i Prewitt-operatoren:

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

(Tabellen er en del av løsningsforslaget.)

Filteret vil altså tilnærme den deriverte i horisontal retning, men er, relativt til det aktuelle konvolusjonsfilteret i den asymmetriske 1D-gradientoperatoren, gjort noe støyrobust ved å 3x1-middelverdifiltrere i vertikal retning og 1x2-middelverdifiltrere i horisontal retning (det er denne 1x2-middelverdi-filtreringen som gjør at Fourier-spekterets verdi er 0 i største horisontale frekvens). Responsen i homogene områder blir 0, mens høy (i absoluttverdi) ved kraftige intensitetsendringer i horisontal retning, f.eks. forårsaket av kanter eller støy. Filteret kan sees på som en detektor av vertikale kanter.

4. Kompresjon og koding

I denne oppgaven skal vi komprimere følgende sekvens av 42 symboler:

bcbcaaabcacbbcacaabcbbaabcbcacabcacabaabcb

Symbolsekvensen består av 15 «a»-er, 14 «b»-er og 13 «c»-er.

a) Finn en Huffman-kodebok for symbolsekvensen.
Hvor mange biter vil vi bruke på å kode symbolsekvensen med Huffman-koding dersom vi ser bort fra den plassen som trengs for å lagre kodeboken?

Fra de oppgitte symbolforekomstantallene ser vi at første sammenslåing i Huffman-kodingsalgoritmen vil slå sammen gruppen som består av «b» og gruppen som består av «c» (ettersom disse to symbolene har færrest forekomster i symbolsekvensen). Siden vi da står igjen med to grupper, gruppen som består av «a» (totalt antall forekomster: 15) og gruppen som består av «b» og «c» (totalt antall forekomster: 27), vil ikke Huffman-kodingsalgoritmen utføre flere sammenslåinger.

Hvis vi for hver sammenslåing, inklusivt sammenslåingen av de to gruppene sammenslåingsdelen av Huffman-kodingsalgoritmen terminerer med, tilordner 0 til den mest sannsynlige gruppen og 1 til den minst sannsynlige gruppen, får vi følgende Huffman-kodebok:

Symbol	a	b	c
Huffman-kodeord	1	00	01

(Tabellen er en del av løsningsforslaget.)

Med denne Huffman-kodeboken (og enhver annen Huffman-kodebok som er dannet ved bruk av forekomstene/sannsynlighetene av symbolene i den oppgitte symbolsekvensen) vil vi bruke:

$$15*1 + 14*2 + 13*2 = 15 + 27*2 = 15 + 54 = 69$$

biter på å kode symbolsekvensen dersom vi ser bort fra plassen som trengs for å lagre kodeboken.

b) Finn Lempel-Ziv-Welch-koden (LZW-koden) av symbolsekvensen når sender og mottaker er enig om å bruke alfabetet {a, b, c}.
Den initielle listen til både LZW-senderen og LZW-mottakeren er altså:

LZW-kode	Symbolsekvens
0	a
1	b
2	c

Hvor mange biter vil vi bruke på å lagre LZW-koden av symbolsekvensen dersom vi lagrer LZW-kodene ved bruk av naturlig binærkoding?

Tips: LZW-koden av symbolsekvensen vil inneholde flere LZW-koder som hver representerer en del av symbolsekvensen.

Tips: Etter LZW-koding av hele symbolsekvensen vil LZW-senderens liste bestå av 16 symbolsekvenser (med tilhørende LZW-koder).

LZW-koden av symbolsekvensen er: 1 2 3 0 6 5 4 8 7 9 11 10 14 13
Etter å ha gjennomgått hele symbolsekvensen er LZW-senderens listen slik:

LZW-kode	**Symbolsekvens**
0	a
1	b
2	c
3	bc
4	cb
5	bca
6	aa
7	aab
8	bcac
9	cbb
10	bcaca
11	aabc
12	cbba
13	aabcb
14	bcacab
15	bcacaba

(Tabellen er en del av løsningsforslaget.)

Med naturlig binærkoding av LZW-kodene vil vi bruke 4 biter per LZW-kode. Siden LZW-koden av symbolsekvensen består av 14 LZW-koder vil vi bruke 14*4 = 56 biter på å lagre LZW-koden av symbolsekvensen.

c) Gjennomsnittlig antall biter per symbol etter Huffman-koding av symbolsekvensen er omtrent 1,64 og gjennomsnittlig antall biter per symbol etter LZW-koding av symbolsekvensen er omtrent 1,33.
Entropien til symbolsekvensen er 1,58.

Hva angir entropien til symbolsekvensen?
Vil alltid det gjennomsnittlige bitforbruket per symbol etter Huffman-koding være større eller lik entropien? Forklar!
Vil alltid det gjennomsnittlig bitforbruket per symbol etter LZW-koding være mindre eller lik entropien? Forklar!

Vi ser fortsatt bort fra den plassen som trengs for å lagre Huffman-kodeboken og antar fortsatt at vi lagrer LZW-kodene ved bruk av naturlig binærkoding.

Entropien til symbolsekvensen angir en nedre grense for gjennomsnittlig bitforbruk per symbol dersom vi koder symbol for symbol.

Siden Huffman-koding koder symbol for symbol, vil det gjennomsnittlige bitforbruket per symbol etter Huffman-koding være nedre begrenset av entropien til symbolsekvensen. Derfor vil alltid det gjennomsnittlige bitforbruket per symbol etter Huffman-koding være større eller lik entropien til symbolsekvensen.

Siden LZW-koding ikke koder symbol for symbol, vil ikke kompresjonsraten være begrenset av entropien til symbolsekvensen. Likevel er det ingen garanti for at LZW-

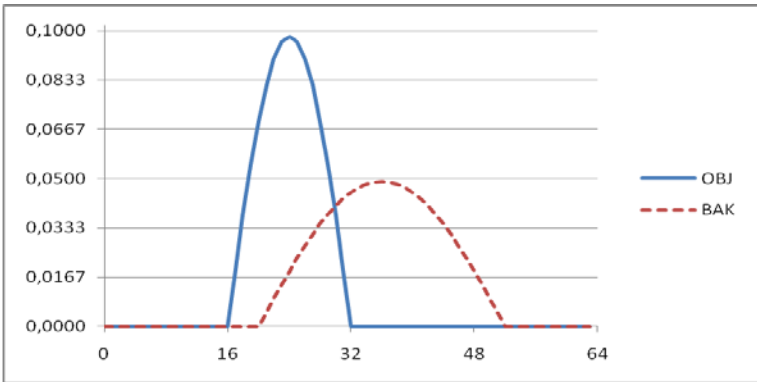
koding resulterer i et lavere (eller likt) gjennomsnittlig bitforbruk per symbol enn det entropien av symbolsekvensen angir. Faktisk, for en tilfeldig symbolsekvens vil LZW-koding normalt sett resultere i et større gjennomsnittlig bitforbruk per symbol enn det entropien av symbolsekvensen angir.
Dette skyldes at LZW-koding premierer mønstre i symbolsekvensen, og i fravær av mange mønstre, som det normalt sett vil være i en tilfeldig symbolsekvens, vil LZW-kodingen resultere i dårlig kompresjonsrate (faktisk kan det gjennomsnittlige bitforbruket per symbol etter LZW-koding t.o.m. være større enn v.b.a. naturlig binærkoding av symbolene).

5. Segmentering ved terskling

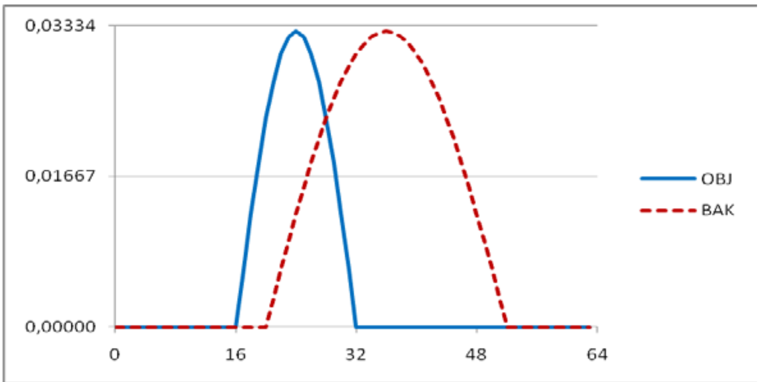
Anta at de normaliserte gråtone-histogrammene til objekt- og bakgrunn i et analogt bilde er gitt ved

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

Med parametrene a = 8, μ = 24 for objekt og a = 16, μ = 36 for bakgrunn ser histogrammene slik ut:



a) Med *a priori* sannsynlighet for objekt O = 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 objektet har lavere gråtoneverdier enn bakgrunnen, og ikke gå i den fellen at objekt-pikslene er lyse og bakgrunnen mørk. Objektfordelingen skaleres altså med 1/3.
Bakgrunnsfordelingen 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):

$$O \cdot p_o(T) = (1-O) \cdot p_b(T) \Rightarrow \frac{1}{3} \cdot \frac{\pi}{32} \cos\left[\frac{(T-24)\pi}{16}\right] = \left(1 - \frac{1}{3}\right) \cdot \frac{\pi}{64} \cos\left[\frac{(T-36)\pi}{32}\right]$$

$$\Rightarrow \cos\left[\frac{(T-24)\pi}{16}\right] = \cos\left[\frac{(T-36)\pi}{32}\right] \Rightarrow \frac{(T-24)}{16} = \pm\frac{(T-36)}{32} \Rightarrow \begin{cases} 2(T-24) = (T-36) \\ 2(T-24) = -(T-36) \end{cases}$$

$$\Rightarrow T = \begin{cases} 12 \ (umulig) \\ \underline{\underline{84/3 = 28.}} \end{cases}$$

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 16 og 32, må skjæringen ligge 1/3 av avstanden mellom de to middelverdiene 24 og 36, altså ved T = 24 + (36-24)*1/3 = 28.

c) Anta at du terskler med den terskelverdien du nettopp har funnet. Vis hvordan du kan finne <u>andelen feilklassifiserte objektpiksler.</u> Hint: Den deriverte av sin(x) er cos(x), og sin(π/4) = 0.5√2.

Her er hovedsaken at man viser hvordan dette skal gjøres, ikke at man beregner eksakt riktig verdi. Integralet av objektfordelingen fra terskelverdien til øvre ende av fordelingen er (substituerer y = z – 24 for å forenkle integralet):

$$E = \int_{28}^{32} \frac{\pi}{32} \cos\left[\frac{(z-24)\pi}{16}\right] dz = \frac{\pi}{32} \int_{4}^{8} \cos\left[\frac{y\pi}{16}\right] dy$$

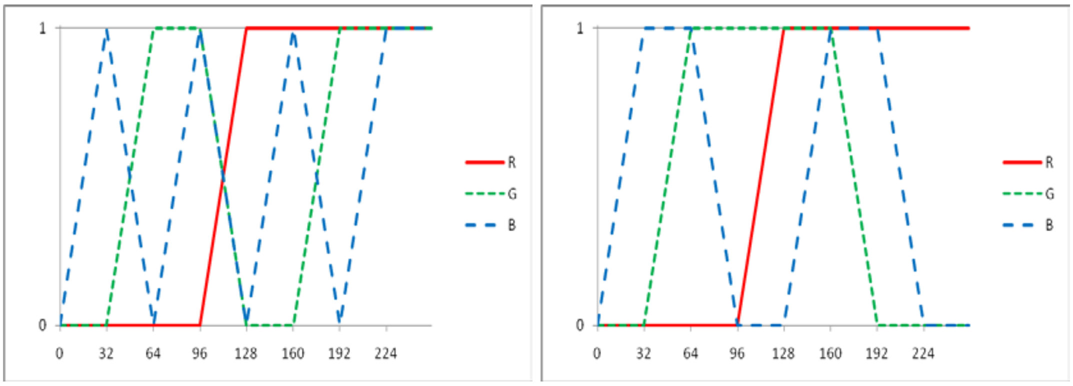
$$= \frac{\pi}{32} \left. \frac{\sin\left(\frac{y\pi}{16}\right)}{\frac{\pi}{16}} \right|_{4}^{8} = \frac{1}{2}\left(\sin\frac{\pi}{2} - \sin\frac{\pi}{4}\right) = \frac{1}{2}\left(1 - \frac{1}{2}\sqrt{2}\right) \approx 0.15$$

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 = 16 til z = 32, dvs y = -8 til y = 8, 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 ≈0,15.</u>

6. Fargetabeller

Et 8 biters gråtonebilde vises fram med to RGB pseudofargetabeller der R, G og B-komponentene er som vist i de to figurene nedenfor, der mengden av hver farge-komponent er normalisert slik at den ligger mellom 0 og 1.



a) Hvis du tegner en RGB-kube, hvilke deler av RGB-kuben vil fargetabellene dekke når gråtonen z går fra 0 til 255? Illustrer med to figurer!

Den første vil starte i svart (0,0,0), gå til blå (0,0,1), gå diagonalt gjennom GB-planet (R=0) til grønn (0,1,0), gå til cyan (0,1,1), gå diagonalt gjennom kuben til rød (1,0,0), til magenta (1,0,1), gå diagonalt gjennom GB-planet (R=1) til gul (1,1,0) og så gå til hvit (1,1,1).

Den andre vil også starte i svart (0,0,0), og gå langs ytterkantene av RGB-kuben til blå (0,0,1), cyan (0,1,1), grønn (0,1,0), gul (1,1,0), hvit (1,1,1), magenta (1,0,1) og rød (1,0,0).

b) Hvor mange og hvilke gråtoner i innbildet vil bli vist fram som gråtoner når vi benytter hver av disse to fargetabellene? Forklar!

Den første fargetabellen går diagonalt gjennom RGB-kuben fra z = 96 (R,G,B = 0,1,1) til z = 128 (R,G,B = 1,0,0), ortogonalt på gråtoneaksen som går fra (R,G,B = 0,0,0) til (R,G,B = 1,1,1). Gråtonen z = 112 vil derfor bli vist fram som (R,G,B = ½, ½, ½), som er den eneste gråtonen denne tabellen kan vise fram, i tillegg til svart (for gråtonen z=0) og hvit (for gråtonene z=224,225,…,255).

Den andre fargetabellen dekker bare 7 av de 12 ytterkantene i RGB-kuben, og vil derfor ikke vise noen av input-gråtonene som gråtoner (R=G=B), bortsett fra svart (for gråtonen z=0) og hvit (for gråtonen z=160).

7. Morfologisk åpning

a) Hva er morfologisk åpning og hva kan det brukes til?

Morfologisk åpning er erosjon etterfulgt av dilatasjon. Samme strukturelement benyttes i erosjonen og dilatasjonen.

Åpning kan brukes til å fjerne støy og lage en åpning (et mellomrom) mellom strukturer som bare henger sammen ved en tynn «bro». Åpning vil også glatte strukturenes omriss (ved å fjerne «utstikkere»).

b) Skisser ut-bildet etter å morfologisk åpne bildet under med strukturelementet under. Vektlegg å få frem hvordan ut-bildet skiller seg fra inn-bildet.
Hvis ønskelig kan du bruke ord og piler til å forklare skissen.

Et binært bilde.
Hvit angir forgrunnspiksel, svart angir bakgrunnspiksel.

Et sirkulært strukturelement.
Hvit angir med i strukturelementet, svart angir ikke med i strukturelementet.
Strukturelementets origo er senterpunktet til sirkelen.

Du kan anta at diameteren til det sirkulære strukturelementet er mindre enn bredden av den tynneste delen i tegnene.

Resultatet etter åpningen er:

(Figuren er en del av løsningsforslaget.)

Viktige momenter:

- De isolerte støyobjektene er borte.
- De tynne koblingene/«broene» mellom «I» og «N» og mellom «2» og «3» er borte.
- Koblingen og støypikselene i nedre del av «3» er borte, sistnevnte p.g.a. bakgrunnspiksler innad i støyområdet.
- Alle konvekse hjørner er avrundet. Konkave hjørner er ikke avrundet.

- Støyområdene som er tilkoblet «I» (til venstre) og «F» (rett over midten) er ikke borte (de er derimot glattet ut, men det er ikke et viktig poeng).
- Støyområdet som neste fyller igjen utstikkingen i øvre del av «1» er ikke borte (også her skjer det noe glatting, uten at det er noe stort poeng i denne oppgaven).

Takk for oppmerksomheten!