Løsningsforslag

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i : INF2310 — Digital bildebehandling
Eksamensdag : Onsdag 1. juni 2011
Tid for eksamen : 14:30 – 18:30
Løsningsforslaget er på : **9 sider**
Vedlegg : **Ingen**
Tillatte hjelpemidler: **Ingen**

- Det er 7 oppgaver i dette oppgavesettet.

- Poengsummen for hver oppgave er angitt, og du kan maksimalt få 200 poeng.

- Les gjennom hele oppgaven før du begynner å løse oppgaven. Kontroller at oppgavesettet er komplett før du begynner å besvare det. Dersom du savner opplysninger i oppgaven, 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 23 delspørsmål, og det lønner seg å disponere tiden slik at man får besvart alle oppgavene. Hvis dere står fast på enkeltoppgaver, gå videre slik at dere 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.

- Dette oppgavesettet foreligger på bokmål, nynorsk og engelsk. Dersom du er usikker på fagtermene som er brukt i den norske oppgaveteksten – siden læreboka er på engelsk - kan du be eksamensvaktene om å få også den engelske oppgaveteksten.

1. Sampling og geometrisk transform

Anta at vi har et samplet bilde med samplingsfrekvens f_s. (Samplingsfrekvensen er lik langs både x- og y-aksen i bildet.) Ved å studere bildets spekter ser vi at det er betydelig bidrag også i de høyeste frekvensområdene.

a) Hva er den høyeste romlige frekvensen det kan være i bildet?

I følge samplingsteoremet må fs>2fmax, der fmax er høyeste frekvens i bildet. Altså må fmax<½ fs.

Det blir så benyttet følgende geometriske transform på det samplede bildet:

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

der c er en positiv konstant, x og y er koordinatene i «innbildet» og x' og y' er de transformerte koordinatene, og det benyttes en «vanlig» resampling ved baklengs transform.

b) For hvilke verdier av c vil vi unngå å introdusere aliasingfeil som følge av den geometriske transformen?

Siden spekteret indikerer at vi har bidrag også for de høyeste frekvensene, kan vi ikke redusere samplingsraten, altså må c>=1.

c) Anta at c = ¼ i transformen over. Uttrykk den nye, effektive, samplingsraten etter transformen ved f_s.

Ved c = ¼ vil vi effektivt ende opp med å velge ut hver fjerde piksel langs hver akse, altså får vi en samplingsrate som er ¼ av den originale; $f_{sny} = c\, f_s = ¼\, f_s$.

d) Hvilken operasjon ville du benyttet på bildet før den geometriske transformen for å redusere aliasingproblemene?

Vi er nødt til å demp/fjerne de høyeste frekvensen, altså de den nye samlingsraten ikke kan håndtere; lavpassfiltrere.

2. Fourier-transform

a) Hva menes med at et bilde er representert i frekvensdomenet?

Bildet er representert som en vektet sum av enkle sinus- og cosinus-bilder, og vi har disse vektkoeffisienten tilgjengelig. Bildet er representert med en alternativ lineær basis.

b) b) Man sier ofte at et alternativ til å utføre en konvolusjonsfiltrering direkte i det romlige domenet, er å gjøre den i frekvensdomenet. Forklar hva som menes med dette utsagnet, og forklar hvilket teorem som er sentralt.

[..] **Konvolusjonsteoremet** sier oss at en konvolusjon i bildedomenet er kun en enkel elementvis multiplikasjon i frekvensdomenet. [..]

3. Konvolusjons-operatorer

Et par av konvolusjonsfiltre H = (H_1, H_2) er definert ved

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ x & 0 & -x \\ 1 & 0 & -1 \end{bmatrix}, \quad H_2 \text{ er } H_1 \text{ rotert } \frac{\pi}{2} \text{ mot urviserenn}$$

a) Forklar hva slags konvolusjons-operator dette filterparet er for x = 1, hva den brukes til, og de viktigste ligningene som beskriver hvordan den brukes.
For x = 1 er dette en Prewitt-operator, som brukes til å estimere gradient-magnitude og gradient-retning i gråtonebilder.
Konvolusjon med H_1 gir gradientmagnituden i x-retning, $g_x(x,y)=f(x,y)*H_1$.
Konvolusjon med H_2 gir gradientmagnituden i y-retning, $g_y(x,y) = f(x,y)*H_2$.
Den totale gradient-magnituden er $G(x,y) =(g_x(x,y)^2 + g_y(x,y)^2)^{1/2}$
Gradient-retningen er gitt ved $\Theta(x,y) = tg^{-1}(g_y(x,y)/g_x(x,y))$

b) Hvilken kjent konvolusjons-operator får vi ved å sette x = 2 og deretter konvolvere et bilde med operatoren $L_1 = H_1*H_1 + H_2*H_2$, der * betegner konvolusjon?
Med x=2 gir H_1 og H_2 de to filtrene i Sobel-operatoren (navnet er ikke viktig).
Det første leddet i L_1 gir et estimat av den to ganger deriverte i x-retning, det andre leddet gir et estimat av den to ganger deriverte i y-retning.
Altså er L_1 en diskret tilnærming til Laplace-operatoren, med negativt fortegn:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

(Ligningen er en del av løsningsforslaget)

c) Vi gjør en syklisk ombytting av filterkoeffisientene i (H_1, H_2), og danner filterparet R = (R_1, R_2), gitt ved

$$R_1 = \begin{bmatrix} x & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -x \end{bmatrix}, \quad R_2 = R_1^T$$

Deretter lager vi operatoren $L_2 = R_1*R_1 + R_2*R_2$, der * betegner konvolusjon.
Vis at $\mathbf{x} = \sqrt{2}$ vil gi $L_2 = L_1$.
Hvis vi setter inn i L_1 og utfører konvolusjonene (husk å rotere det ene filtret 180 grader!), ser vi at (vi trenger bare å utføre den første konvolusjonen, siden $H_2*H_2 = H_1^T*H_1^T$)

$$L_1 = \begin{bmatrix} 1 & 0 & -1 \\ x & 0 & -x \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ x & 0 & -x \\ 1 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 1 & x & 1 \\ 0 & 0 & 0 \\ -1 & -x & -1 \end{bmatrix} * \begin{bmatrix} 1 & x & 1 \\ 0 & 0 & 0 \\ -1 & -x & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -2 & 0 & 1 \\ 2x & 0 & -4x & 0 & 2x \\ x^2+2 & 0 & -2x^2-4 & 0 & x^2+2 \\ 2x & 0 & -4x & 0 & 2x \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2x & x+2 & 2x & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -4x & -2x-4 & -4x & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2x & x^2+2 & 2x & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 2x & x^2 & 2x & 2 \\ 2x & 0 & -4x & 0 & 2x \\ x^2 & -4x & -4x^2-8 & -4x & x^2 \\ 2x & 0 & -4x & 0 & 2x \\ 2 & 2x & x^2 & 2x & 2 \end{bmatrix}$$

Mens vi tilsvarende for L_2 får

$$L_2 = \begin{bmatrix} x & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -x \end{bmatrix} * \begin{bmatrix} x & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -x \end{bmatrix} + \begin{bmatrix} 0 & 1 & x \\ -1 & 0 & 1 \\ -x & -1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & x \\ -1 & 0 & 1 \\ -x & -1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} x^2 & 2x & 1 & 0 & 0 \\ 2x & 2 & -2x & 2 & 0 \\ 1 & -2x & -2x^2-4 & -2x & 1 \\ 0 & -2 & -2x & 2 & 2x \\ 0 & 0 & -1 & 2x & x^2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 2x & x^2 \\ 0 & -2 & -2x & 2 & 2x \\ 1 & -2x & -2x^2-4 & -2x & 1 \\ 2x & 2 & -2x & -2 & 0 \\ x^2 & 2x & 1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} x^2 & 2x & 2 & 2x & x^2 \\ 2x & 0 & -4x & 0 & 2x \\ 2 & -4x & -4x^2-8 & -4x & 2 \\ 2x & 0 & -4x & 0 & 2x \\ x^2 & 2x & 2 & 2x & x^2 \end{bmatrix}$$

De to 5x5 filtermatrisene er like bortsett fra de 8 stedene der det i den ene står 2 og i den andre står x^2.

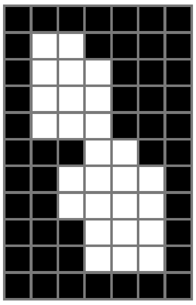
$L_1 = L_2$ krever da at $x = \sqrt{2}$.

Det vil si at H er Frei-Chen operatoren (navnet er ikke vesentlig her).

(Ligningene ovenfor er en del av løsningsforslaget)

4. Morfologiske operasjoner på binære bilder

La hvitt være 1 og svart være 0 i bildet under. Anta at vi benytter et 3x3 kvadratisk strukturelement med origo i midten.

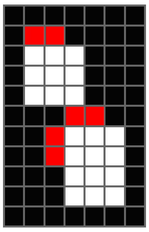


a) Hva vil slutt-resultatet være ved gjentatte morfologiske erosjoner?

Svar: All struktur i bildet forsvinner. Kun bestående av 0-ere.

b) Utfør og vis resultatet av en morfologisk åpning. (Morfologisk åpning er morfologisk erosjon etterfulgt av morfologisk dilasjon.)

Svar: Piksler merket med rødt vil forsvinne.



(Figuren over er en del av løsningsforslaget)

c) Hvis man etter en segmentering satt igjen med et binært bilde hvor objektene hadde litt rufsete kanter og små, uønskede hull spredt rundt omkring, hvilke morfologiske operasjoner ville du da benyttet for å "rengjøre" bildet?

Svar: Morfologisk lukking, siden det vil fylle hullene, evt. Fulgt av åpning slik at både viker og nes fjernes fra den rufsete kanten

5. Kompresjon, koding og filtrering

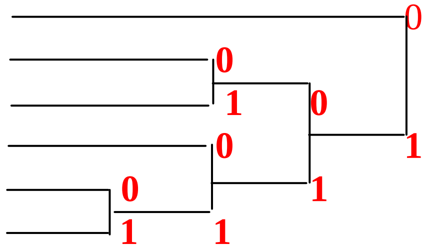
Gitt et 5 x 5 utsnitt av et 3 bits gråtonebilde med pikselverdier

4	2	1	2	2
2	1	3	2	2
2	3	5	5	6
2	1	5	5	6
2	2	6	6	6

a) Vis hvordan du vil gå fram for å Huffman-kode dette bildet, og finn det gjennomsnittlige antall biter per piksel, R, i det kodede bildet.

Gråtonene (første kolonne) og histogrammet (andre kolonne) gir kodeboken

2:	10	**0**
6:	5	**100**
5:	4	**101**
1:	3	**110**
3:	2	**1110**
4:	1	**1111**



Og det gjennomsnittlige antall bits pr piksel etter koding blir

R = (10x1 + (5+4+3)x3 + (2+1)*4)/25 = 58/25 = **<u>2.32 biter/piksel.</u>**

b) ”I dette bildeutsnittet er kodingsredundansen 0.0595”
Hva mener vi når vi sier dette?
Hva skal til for at kodingsredundansen for Huffman-koding skal bli 0?
Svar: Kodingsredundansen er definert som R – H,
der H er første ordens gråtone-entropi, $H = -\text{sum}(p(i)\log_2(p(i)))$.
For at kodingsredundansen skal bli null må alle sannsynlighetene i bildets normaliserte histogram kunne skrives som en invers toer-potens.

c) Anvend et 3x3 kvadratisk medianfilter på bildeutsnittet. Ut-bildet skal her være like stort som inn-bildet. Du må altså finne medianen også når filtermasken bare dekker deler av bildeutsnittet. Hvis antall bildepiksler under filtermasken er et partall, så er medianen snittet av de to pikselverdiene midt i den sorterte sekvensen. Hva blir ut-bildet?
Svaret er:

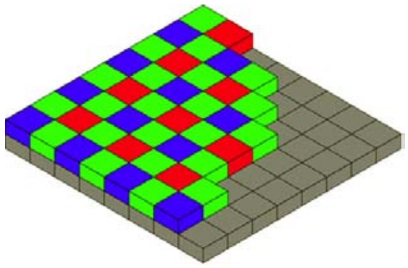
2	2	2	2	2
2	2	2	2	2
2	2	3	5	5
2	2	5	6	6
2	2	5	6	6

d) Kan en slik median-filtrering påvirke kompresjonsraten i Huffman-koding av et bilde? Forklar kort!
Svar: Ja, det kan den gjøre, fordi medianfiltreringen kan endre bildets histogram. Her blir R = 1.56, fordi histogrammet samles om færre pikselverdier.

e) Hvis vi har et perfekt histogramutjevnet bilde med 2^b gråtoner, er det da noe poeng i å foreta en Huffman-koding av bildet? Forklar kort!
Svar: Hvis vi har en perfekt histogramutjevning, dvs at alle gråtonene har sannsynlighet $p = 1/2^b$, så er det ikke noe poeng i en Huffman-koding, siden det optimale er like lange kodeord.

6. Farger og fargetabeller

I de fleste digital-kameraer sitter det en sensor som er basert på Bryce Bayers patent fra 1976. Her er det tatt hensyn til at synssystemet vårt er mer følsomt for grønt lys enn for rødt og blått.
I et 8 x 8 piksel utsnitt av en slik detektor registrerer 50% av pikslene grønt lys, mens 25% registrerer blått og 25% rødt, i et "Bayer-mønster" som vist på figuren til høyre.



a) Hvis nå R og B lagres med 4 biter per piksel og G med 8 biter per piksel i dette såkalte "rå-formatet", hvor mange biter må vi da lagre for hvert 8x8 "Bayer-mønster"?
(64/4)*4 + (64/2)*8 + (64/4)*4 = (64/2)*(8+4) =64*6 = 384 Byte.

b) Detektoren måler intensiteten i blått og rødt bare i ¼ av pikslene.
Beskriv minst to metoder vi kan bruke for å rekonstruere den blå og røde komponenten i resten av pikslene.
Enklest (og dårligst): Nærmeste nabo interpolasjon.
Bi-ineær interpolasjon mellom fire og fire piksler der intensiteten i blått er registrert. Dermed kan vi finne intensiteten i de øvrige fem pikslene. Gjentas for alle 3x3 kvadrater med blå detektorer i hjørnene. Tilsvarende for 3x3 kvadrater med røde detektorer i hjørnene.
Bikubisk interpolasjon; som involverer de 16 nærmeste naboene til et piksel.

c) Anta at det rekonstruerte RGB-bildet er lagret med 6 bit rød + 6 bit grønn + 4 bit blå, altså til sammen 16 biter per piksel. Hvor mange forskjellige rene gråtoner kan det maksimalt være i et slikt bilde?
For at et piksel skal være en ren gråtone, må de tre RGB-komponentene være like. Det er 2^6 = 64 forskjellige verdier av R og G, men bare 2^4 = 16 forskjellige verdier av B. Altså kan det bare være 16 forskjellige rene gråtoner i dette bildet.

d) En produsent sier at "Vi har en meget god rekonstruksjon, og bruker en LUT med 2^{16} = 65 536 mulige farger i hvert piksel, og LUT'en gir 24 bits RGB i hvert eneste piksel".
Hvor mange Byte trengs da for å lagre et 8x8 piksels utsnitt av bildet, og hvor stor plass tar LUT'en, uttrykt i MiB (1024*1024 Byte = 2^{20} Byte)?
Pikslene i bildet må ha en ordlengde på 16 biter for å kunne kunne henvise til de 65 536 (= 2^{16}) linjene i LUT'en. 16 biter er 2 Byte.
Et 8x8 piksels utsnitt av bildet tar da ganske enkelt 8*8*2 Byte = 128 Byte.
Det ligger et kraftig hint i oppgaven om at bildeutsnittet tar mye mindre plass enn LUT'en. For LUT'en tar selvsagt mye større plass enn dette lille utsnittet:
2^{16} linjer a 3 Byte = 3* 2^{16} Byte.
1 MiB er 2^{20} Byte.
Altså tar LUT'en $3/2^4$ MiB =3/16 MiB.

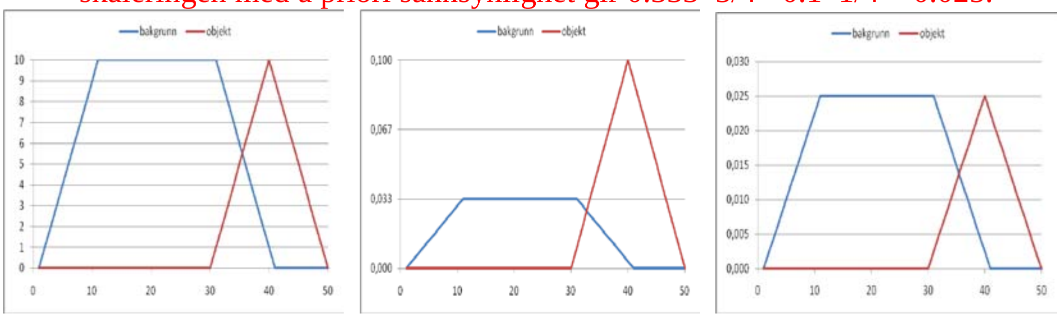
7. Segmentering ved terskling

Et 20x20 piksler gråtonebilde har en bakgrunnsintensitet som stiger fra øvre venstre hjørne mot nedre høyre hjørne. Midt i bildet ligger et 10x10 piksler objekt der intensiteten stiger i motsatt retning. Pikselverdiene er gitt nedenfor, og objektet er markert.

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
7	8	9	10	11	49	48	47	46	45	44	43	42	41	40	22	23	24	25	26
8	9	10	11	12	48	47	46	45	44	43	42	41	40	39	23	24	25	26	27
9	10	11	12	13	47	46	45	44	43	42	41	40	39	38	24	25	26	27	28
10	11	12	13	14	46	45	44	43	42	41	40	39	38	37	25	26	27	28	29
11	12	13	14	15	45	44	43	42	41	40	39	38	37	36	26	27	28	29	30
12	13	14	15	16	44	43	42	41	40	39	38	37	36	35	27	28	29	30	31
13	14	15	16	17	43	42	41	40	39	38	37	36	35	34	28	29	30	31	32
14	15	16	17	18	42	41	40	39	38	37	36	35	34	33	29	30	31	32	33
15	16	17	18	19	41	40	39	38	37	36	35	34	33	32	30	31	32	33	34
16	17	18	19	20	40	39	38	37	36	35	34	33	32	31	31	32	33	34	35
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

a) Tegn tre skisser: en skisse av bakgrunns- og objekthistogrammet, en skisse av de normaliserte histogrammene for bakgrunn og objekt, og en skisse av de normaliserte histogrammene skalert med a priori sannsynlighetene.
Pass på å sette riktige verdier på aksene!

Her er det vesentlig at maksverdiene i histogrammet er 10, at maksverdiene i de normaliserte histogrammene er hhv 10/300 = 0.033.. og 10/100 = 0.1, og at skaleringen med a priori sannsynlighet gir 0.333*3/4= 0.1*1/4 = 0.025.



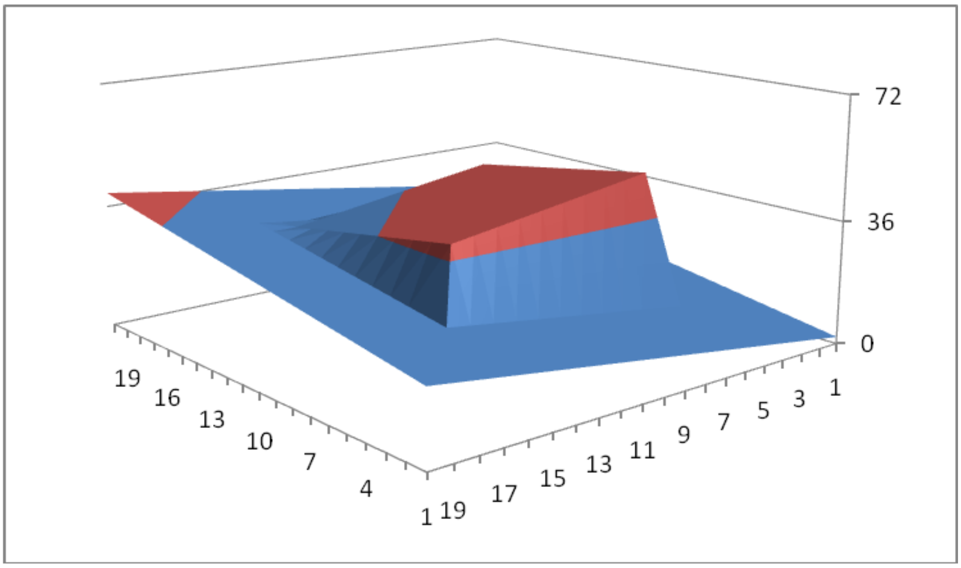
b) Forklar kort hvordan vi kommer fram til en ligning som viser hvordan man generelt finner optimal terskelverdi. Bruk en av skissene ovenfor til å finne den tersklen som gir minst mulig feil ved segmentering av dette bildet, og marker i en skisse av bildet hvilke piksler som etter terskingen er objektpiksler.
Vi får minst mulig feil når vi velger en terskel gitt ved

$$O \cdot o(T) = B \cdot b(T)$$

der O og B er a priori sannsynlighetene for hhv objekt og bakgrunn, mens o(t) og b(t) er de normaliserte objekt- og bakgrunnshistogrammene. Minst feil oppnås altså ved i finne den terskel T der de skalerte normaliserte histogrammene er like. Fra de skalerte normaliserte histogrammene ser vi at bakgrunnshistogrammet ligger over objekthistogrammet for pikselverdier **≤ 35** (hvis vi bruker normalisert histogram her vil vi sette terskelen til 32). Etter terskling med T=35 vil pikslene markert i fet rød font være objektpiksler. Det er 15 piksler i hjørnet av objektet og 15 piksler i hjørnet av bakgrunnen som blir feilklassifisert.

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
7	8	9	10	11	**49**	**48**	**47**	**46**	**45**	**44**	**43**	**42**	**41**	**40**	22	23	24	25	26
8	9	10	11	12	**48**	**47**	**46**	**45**	**44**	**43**	**42**	**41**	**40**	**39**	23	24	25	26	27
9	10	11	12	13	**47**	**46**	**45**	**44**	**43**	**42**	**41**	**40**	**39**	**38**	24	25	26	27	28
10	11	12	13	14	**46**	**45**	**44**	**43**	**42**	**41**	**40**	**39**	**38**	**37**	25	26	27	28	29
11	12	13	14	15	**45**	**44**	**43**	**42**	**41**	**40**	**39**	**38**	**37**	**36**	26	27	28	29	30
12	13	14	15	16	**44**	**43**	**42**	**41**	**40**	**39**	**38**	**37**	**36**	35	27	28	29	30	31
13	14	15	16	17	**43**	**42**	**41**	**40**	**39**	**38**	**37**	**36**	35	34	28	29	30	31	32
14	15	16	17	18	**42**	**41**	**40**	**39**	**38**	**37**	**36**	35	34	33	29	30	31	32	33
15	16	17	18	19	**41**	**40**	**39**	**38**	**37**	**36**	35	34	33	32	30	31	32	33	34
16	17	18	19	20	**40**	**39**	**38**	**37**	**36**	35	34	33	32	31	31	32	33	34	35
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	**36**
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	**36**	**37**
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	**36**	**37**	**38**
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	**36**	**37**	**38**	**39**
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	**36**	**37**	**38**	**39**	**40**

Eller om man vil: Rød som objekt, blå som bakgrunn:



TAKK FOR OPPMERKSOMHETEN !