1 Sampling

Anta et avbildningssystem som avbilder objekter på lang avstand over på et avbildningsplan. I dette avbildningsplanet vil vi senere utføre sampling av bildet.

Blant det du har mulighet til å avbilde er to punkt-lignende objekter (altså objekter med neglisjerbar utstrekning). Ved nitidig utprøving kommer du frem til at det nærmeste man kan sette disse objektene slik at de blir adskillige i bildet er 0.75 m. Anta at man ved geometriske betraktninger kan regne seg frem til at dette tilsvarer 2 mm i avbildningsplanet.

a) Hva kan du si om bredden på punktspredningsfunksjonen (PSF) til avbildningssystemet?

Bredden $\geq$ 2 mm. [Oppgaven sier intet om formen på punktspredningsfunksjonen.] Hadde bredden vært mindre, ville man kunne sette punktene nærmere hverandre uten at de ble smurt sammen i bildet.

b) Hva kan du si om høyeste mulige romlige frekvens i bildet ($f_{\max}$)?

[I kurset fokuserer vi på frekvenser med bidrag "av betydning", og ser altså bort i fra slikt som dette med uendelig båndbredde grunnet endelig utstrekning på bildet.] Minste periode er 2 mm, altså vil høyeste romlige frekvens være $\frac{1}{2\,\mathrm{mm}} = 0.5\,\mathrm{mm}^{-1} = f_{\max}$.

c) Hvor tett må du sample bildet for å unngå aliasing?

Samplingsteoremet krever $f_s > 2f_{\max}$. Altså må minste samplingsperiode være *mindre* enn $1/f_s = 1/(2f_{\max}) = 1\,\mathrm{mm}$.

d) Det samples nært opptil denne frekvensen. Det utføres så en digital lavpassfiltrering hvor "cut-off"-frekvensen er 1/3 av høyeste mulige frekvens i bildet. Med hvilken faktor kan man nedsample dette bildet uten å miste (nevneverdig) informasjon?

$f_{\max,\text{ ny}} = \frac{1}{3}f_{\max}$, så, i følge samplingsteoremet, kan man sette $f_{\text{s, ny}} = \frac{1}{3}f_s$ som tilsvarer en tre ganger så høy periode, som igjen betyr at vi kan nedsample med en faktor 3.

e) Anta at det *ikke* ble gjort denne lavpassfiltreringen og nedsamplingen. La oss si at man i stedet oppsampler bildet med en faktor 10 per akse, og la oss si at det benyttes en nær perfekt interpolasjon (tenk feks sinc-interpolasjon). Hva er da høyeste mulige romlige frekvens i bildet?

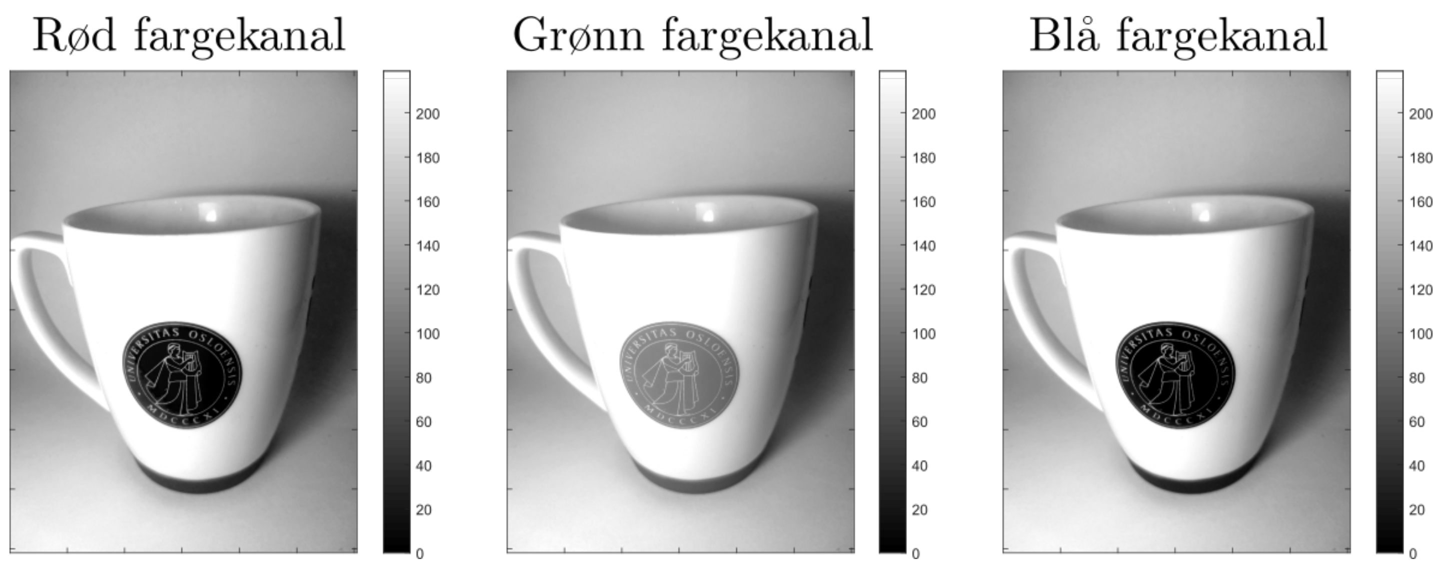
En oppsampling øker ikke den romlige oppløsningen. $f_{\max}$ forblir altså den samme.

f) Anta uniform fordeling av lysintensiteter (tenk like mange av alle intensiteter). Lag figurer og argumenter for hvorfor kvantiseringsfeilen vil kunne halveres om man får ett bit mer tilgjengelig per piksel.

Ett ekstra bit gir dobling av antall kvantiseringsnivåer. Videre er en nærliggende løsning å tenkte arealer av trekanter. [Ikke vist her.]

2 Farger

Vi har et fargebilde av en kopp med UiO-logoen (det er logoen der det står "UNIVERSITAS OSLOENSIS" og illustrerer en kvinne som holder en lyre). Vi får se kun bildets RGB-kanaler:



Forklar og begrunn hvilken farge UiO-logoen på koppen har og hvilken farge det er på bakgrunnen i det innleste fargebildet basert på informasjonen fra dets RGB-kanaler.

Fargen på logoen er grønn fordi vi ser høy respons i den grønne fargekanalen, men lite respons i de to andre kanalene. Fargebildet ser slik ut:

Siden bakgrunnen ser omtrent lik ut i form av respons i både rød-, grønn- og blå-kanal, er bakgrunnen grå med variende intensitet, som vi ser i det presenterte fargebildet over.

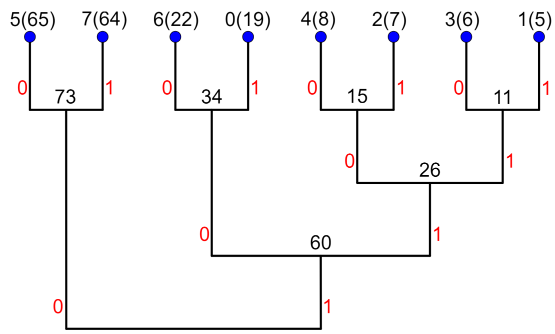
3 Huffman-koding

Vi får vite at forekomsten til pikselintensiteter til et 3-bit bilde er:

Intensitet	0	1	2	3	4	5	6	7
Antall piksler	19	5	7	6	8	65	22	64

a) Lag en kodebok med Huffman-kodeord til hver intensitet. Vis hvordan du kom frem til Huffman-kodeordene.

Vi kan bruke hyppighetene (det oppgitte antall piksler per intensitet) til å lage følgende Huffman-kodetre (der 0 tilordnes den mest sannsynlige gruppen og 1 den minst sannsynlige gruppen etter hver sammenslåing):



Dette gir følgende kodebok:

Intensitet	0	1	2	3	4	5	6	7
Huffman-kodeord	101	1111	1101	1110	1100	00	100	01

b) Hvor mange bits trengs for å lagre bildet etter Huffman-koding? Vi ser bort fra lagringsplassen kodeboken tar.

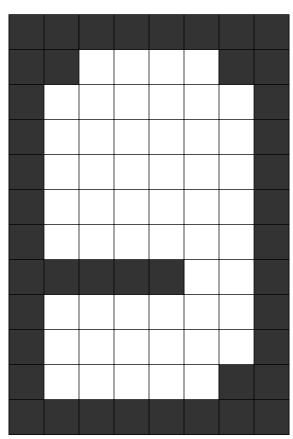
19*3 + 5*4 + 7*4 + 6*4 + 8*4 + 65*2 + 22*3 + 64*2 = 485 biter trengs for å lagre hele bildet.

c) Hva blir kompresjonsraten for dette bildet ved Huffman-koding? Vi ser igjen bort fra lagringsplassen kodeboken tar.

19 + 5 + 7 + 6 + 8 + 65 + 22 + 64 = 196 pikslerverdier finnes i bildet og det tar 3 biter å lagre hver av dem med naturlig binærkoding, noe som gir et totalt forbruk på 588 biter. Kompresjonsraten blir dermed $\frac{588}{487}$, som er omtrent 1,2.

4 Kantdeteksjon av binære bilder ved morfologiske operasjoner

Gitt følgende binære bilde:

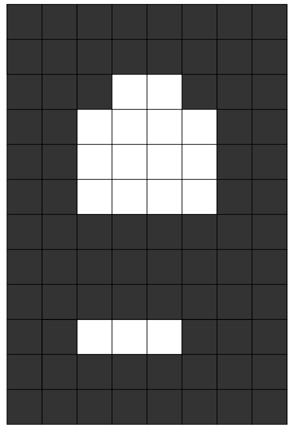


Her lar vi sort representere 0 og hvit 1.

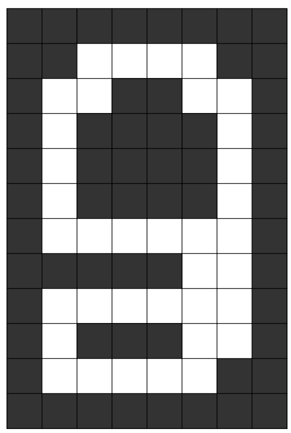
a) Utfør en kantdeteksjon på det binære bildet for hånd. Bruk et 3×3 kvadratisk strukturelement i ett av stegene av kantdeteksjonen. Vis hvordan strukturelementet blir brukt og forklar hvordan du går frem.

I kantdeteksjon er det to steg; erosjon av bildet og deretter en subtraksjon mellom innbildet og det eroderte bildet.

Det eroderte bildet med et 3×3 kvadratisk strukturelement er



Differansen mellom innbildet og det eroderte bildet er



som er det kantdetekterte bildet der et 3×3 kvadratisk strukturelement har blitt brukt.

b) Forklar hvilken tilkobling vi generelt ville ha fått på kantene etter kantdeteksjon ved å bruke et 3×3 sirkulært strukturelement (dette er da et 3×3 pluss-formet strukturelement) og et 3×3 kvadratisk strukturelement.

Kanten blir 4-tilkoblet etter en kantdeteksjon der vi har brukt et kvadratisk strukturelement. Derimot vil kanten generelt bli 8-tilkoblet dersom vi bruker et 3×3 sirkulært strukturelement.

5 Filtrering

I denne oppgaven skal vi se på følgende bilde:

Bildet

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

Dersom verdien til piksler utenfor bilderanden settes til verdien til nærmeste piksel innenfor bilderanden, hva blir da ut-bildet, som er av samme størrelse som inn-bildet, etter:

a) 3×3-middelverdifiltrering?

$$\frac{1}{9}\begin{pmatrix} 57 & 56 & 41 & 29 \\ 43 & 41 & 31 & 23 \\ 24 & 24 & 19 & 14 \\ 5 & 5 & 5 & 6 \end{pmatrix} \approx \begin{pmatrix} 6,3 & 6,2 & 4,6 & 3,2 \\ 4,8 & 4,6 & 3,4 & 2,6 \\ 2,7 & 2,7 & 2,1 & 1,6 \\ 0,6 & 0,6 & 0,6 & 0,7 \end{pmatrix}$$

b) 3×3-medianfiltrering?

$$\begin{pmatrix} 7 & 6 & 5 & 2 \\ 6 & 6 & 2 & 2 \\ 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

c) symmetrisk nærmeste nabo-filtrering med 3×3-naboskap? Hvis to piksler har ulik verdi og ligner like mye på verdien de skal sammenlignes med (så absoluttverdien av differansene er like), velg da den minste pikselverdien blant de to aktuelle pikselverdiene.

$$\frac{1}{5}\begin{pmatrix} 33 & 29 & 29 & 9 \\ 31 & 32 & 27 & 6 \\ 5 & 2 & 2 & 6 \\ 0 & 1 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 6,6 & 5,8 & 5,8 & 1,8 \\ 6,2 & 6,4 & 5,4 & 1,2 \\ 1,0 & 0,4 & 0,4 & 1,2 \\ 0,0 & 0,2 & 0,6 & 0,2 \end{pmatrix}$$

d) Hvilken filtering fra deloppgavene a) - c) er best til å bevare kanter og redusere støy innad i objektene? Merk at vi har to objekter i bildet. De tre første pikslene fra venstre i første rad og de tre første pikslene fra venstre i andre rad tilhører et objekt, mens de resterende pikslene tilhører et annet objekt.

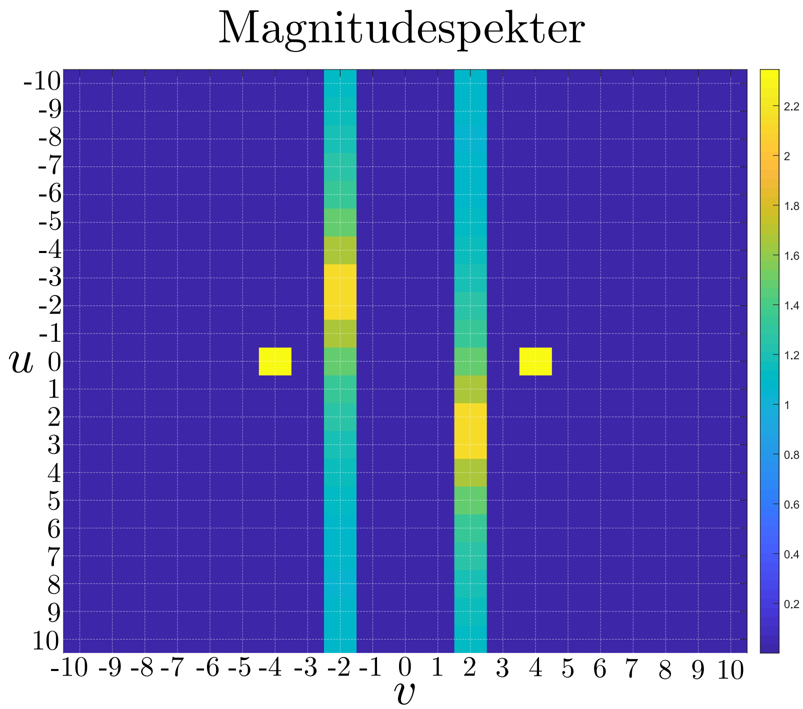
3×3-middelverdifiltrering glatter ut kanter, men er god på å redusere støy i form av små (innenfor 3×3-vinduet), lokale variasjoner (altså små endringer av mange pikselverdier).

3×3-medianfiltrering bevarer kanter og er god på en del type støy slik som salt-pepper-støy, men kan avrunde hjørner og være litt mindre passende for annen type støy slik som lokale variasjoner.

Symmetrisk nærmeste nabo-filtrering med 3×3-naboskap forsøker å utnytte at det kan være to piksel-populasjoner i naboskapet rundt et vilkårlig punkt (x, y) og at pikselverdien i (x, y) definerer hvilken av de to piksel-populasjonene vi bør velge (der sammenligningen begrenses til hvert symmetrisk piksel-par i naboskapet rundt (x, y)). På den måten kan kanter og hjørner bevares ganske godt, samtidig som en del typer støy (f.eks. lokale variasjoner) reduseres ganske godt (men ikke like godt som middelverdifiltrering). Filtrerer ikke salt-pepper-støy like bra som medianfiltrering.

6 Fouriertransformasjon

a) Under ser vi magnituden av Fourier-spekteret til et innbilde. Beskriv hvordan bildet ser ut i bildedomenet og hva det består av basert på informasjonen du får ved å bare se på spekteret.



Bildet vil se ut som to sinuser (det er ikke mulig å se fra kun magnitude spekteret om det sinus eller cosinus, men en av de to skal studenten ha svart.) som er addert sammen. Hver av dem har en vertikal frekvens u og horisontal frekvens v. Den ene sinusen har frekvens $(0, 4)$ og den andre $(2.5, 2)$.

b) Hvilket artefakt kan vi få i et bilde ved å filtrere det med et filter som nuller ut frekvenser innad en sirkel sentrert om $(0, 0)$-frekvensen og er lik 1 ellers?

Ringing. Denne effekten kan sees ved at bl.a. kanter brer seg utover i bildet i et gjentakende mønster.

c) Hvis vi har to like store bilder f_1 og f_2 og kjenner deres Fouriertransformasjon F_1 og F_2, hvordan kan vi finne Fouriertransformasjonen av $f_1 + f_2$ *uten* å gjøre noen flere Fouriertransformasjoner?

Vi kan finne Fouriertransformasjonen av $f_1 + f_2$ ved å beregne summen $F_1 + F_2$. Dette kan man beskrefte ved å beregne Fouriertransformasjonen av $f_1 + f_2$ direkte:

$$
\begin{aligned}
F(u,v) &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f_1(x,y) + f_2(x,y)) \mathrm{e}^{-2\pi j \left(\frac{ux}{M} + \frac{vy}{N}\right)} \\
&= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f_1(x,y) \mathrm{e}^{-2\pi j \left(\frac{ux}{M} + \frac{vy}{N}\right)} + f_2(x,y) \mathrm{e}^{-2\pi j \left(\frac{ux}{M} + \frac{vy}{N}\right)} \\
&= F_1(u,v) + F_2(u,v)
\end{aligned}
$$

d) Anta vi har et bilde som består av en disk. Bakgrunnspikslene har intensitet lik 0 og disken intensitet 1. Vi gjør en punktvis multiplikasjon mellom bildet av disken og et bilde som består av en Gauss sentrert midt i bildet. Hvordan vil 2D-DFT'en av bildet etter den punktvise multiplikasjonen bli sammenlignet til 2D-DFT'en til bildet med disken?

Fourier-spekteret vil bli glattet ut siden vi har fra konvolusjonsteoremet at en punktvis multiplikasjon i bildedomenet gir en sirkelkonvolusjon i frekvensdomenet. En Gauss i bildedomenet er også en Gauss i frekvensdomenet. Dette betyr at vi konvolverer 2D-DFT'en til Gauss'en over 2D-DFT'en til disken og fra det "smører" ut 2D-DFT'en til disken. Derfor vil ringingen synlig i spekteret til disken være redusert etter multiplikasjon med Gauss'en i bildedomenet.

7 Segmentering

Anta at vi har et bilde som består av noen objekter av interesse, som fremstår som lysere, på en bakgrunn, som fremstår som mørkere, i bildet. Vi ønsker så å segmentere bildet i forgrunn og bakgrunn kun basert på pikselintensitetene.

Anta også at man har histogrammet for forgrunnen og bakgrunnen hver for seg, henholdsvis h_1 og h_2. Histogrammet for hele bildet er $h = h_1 + h_2$.

a) La oss si det nå blir trukket ut en tilfeldig piksel, og den viser seg å ha intensitet i. Hva er sannsynligheten for at denne pikselen er en forgrunnspiksel?

$\frac{h_1[i]}{h[i]}$.

b) La oss nå si at du mottar et nytt bilde, et bilde som ligner på bildet du har h_1 og h_2 for. Men på dette bildet er objektene (forgrunnen) dobbelt så store, og tilsvarende dekker bakgrunnen halvparten så mange piksler. Gi en regel for hvordan du ville klassifisert (til forgrunn eller bakgrunn) en piksel med intensitet i.

Velg forgrunn om $2 * h_1[i] > 0.5 * h_2[i]$ og ellers bakgrunn. Dette vil minimere totalt antall feilklassifiserte piksler dersom det nye bildet inneholder nøyaktig dobbelt så mange forgrunnspiksler av hver pikselintensitet og nøyaktig halvparten så mange bakgrunnspiksler av hver pikselintensitet.

c) Anta så at vi i stedet ønsker å kun benytte en enkel terskel for å dele et bilde inn i forgrunn og bakgrunn. For det første bildet, forklar hvorfor denne terskelen bør være ved et skjæringspunkt mellom h_1 og h_2 om man ønsker å minimere antall feilklassifiserte piksler. For enkelhet kan vi anta vi har at h_1 er ulik h_2 for alle i bortsett fra i skjæringspunktet.

La oss si at vi har en terskel, t, der $h_1[t] = h_2[t]$ og klasse for $i < t$ og $i > t$ satt optimalt/riktig. Dette siste betyr at $t - 1$ og $t + 1$ isolert sett gir optimal prediksjon av bakgrunn/forgrunn. Endres terskelen vil enten $t - 1$ eller $t + 1$ bytte klasse, og vil følgelig gi opphav til flere feilklassifiserte piksler. [Andre forklaringer også mulig. Godta også om de går veien om kontinuerlige fordelinger.]

8 Gråtonetransfomer

Anta et gråtonebilde hvor piksleverdiene har en uniform fordeling i intervallet [40, 300]. Vi opererer med her 10 bits kvantisering per piksel. Tenk verdien 0 som sort og høyeste heltall som hvitt.

a) Gi den lineære gråtonetransformen som vil maksimere kontrasten.

Ønsker uniform fordeling av intensitetene i støttet intervall. $T[i] = ai - 40a$, der $a = \frac{2^{10}-1}{300-40}$.

b) Anta så at vi har et 10-bits bilde med uniformt histogram. Vi får oppgitt et ønsket histogram $h_ø$. Hvordan kan vi lage en gråtonetransform som endrer histogrammet til bildet vårt slik at det ligner på det ønskede histogrammet?

Histogramutjevning kan gjøres ved bruk av skalert, kumulativt histogram som transform. Vi ønsker det omvendte, og benytter så en invers slik transform. Altså T^{-1}, der $T[i] = G \sum_{k=0}^{i} h_ø[k]/(n \times m)$ og $G = 2^{10}$.

c) Anta at vi har et sett med gråtonebilder. Oppgaven er å gi alle bildene nogenlunde samme lyshet og kontrast. Til dette får du beskjed om å benytte en gråtonetransform på formen $T[i] = ai + b$, der a og b er reelle tall. Forklar, og beskriv hovedtrekkene, for hvordan du ville gått frem.

Vanlig (i kurset) mål på lyshet og kontrast er henholdsvis gjennomsnitt og varians. Disse kan spesifiseres, la oss si m_T og σ_T. Et bilde kan få tilnærmede slike verdier ved beregne dets nåværende gjennomsnitt (m) og standardavvik (σ), og benytte $a = \frac{\sigma_T}{\sigma}$ og $b = m_T - am$. [Jfr. forelesningsnotater s. 26 om gråtonetransformasjoner.]

d) Anta samme mål som i forrige deloppgave, men vi opererer i stedet med fargebilder (RGB-bilder). Hvordan ville du gått frem for å standardisere lyshet og kontrast da?

Utføre transformen på I-komponenten ved en HSI-representasjon av bildene.

e) Anta en gråtonetransform T som opererer på bilder med 10 bits kvantiseringsnivå. Hvilke krav til T må være tilfredstilt for at vi skal kunne finne en annen transform, T^{-1}, som etter at gråtonetransformen, T, er utført gir oss det opprinnelige bildet?

Alle T må være unike, altså $T[i] \neq T[j]$ for alle $i \neq j$. [Antar at T inneholder kun støttede verdier.]