INF2310 oblig 2

haavahu

April 2020

Innhold

1 Oppgave 1

1.1 Oppgave 1.2

Når man gjør filtrering i frekvensdomenet er det nødvendig å nullutvide det transformerte bildet slik at det blir riktig størrelse. Da følger det også at man bør trimme bildet ned til original størrelse når man har transformert tilbake til bildedomenet. Det kan oppstå artefakter ved bilderanden når vi gjør middelverdifiltrering, dette ser vi spesielt på bildet som ble filtrert i bildedomenet i figur (1) med den svarte kanten rundt hele bildet. Det frekvensfiltrerte bildet har også artefakt ved øverste kanten, men ikke på de andre kantene.

(a) Originalt bilde

(b) Middelverdifiltrert

(c) Fourier middelverdifiltrert

Figure 1: Originalt bilde, romlig middelverdifiltrert bilde og frekvensdomene middelverdifiltrert

1.2 Oppgave 1.3

Tidsgrafene tyder på at det lønner seg å utføre filtreringen i frekvensdomenet da dette er raskest. Dette gjelder spesielt for større filtre da det ser ut til at tidskompleksiteten til conv2-funksjonen er eksponentiell, og kompleksiteten til multiplikasjonen i frekvensdomenet er lineær. Så å filtrere i frekvensdomenet er raskere til tross for at man må gjøre ekstra arbeid ved å transformere til fourierform og tilbake igjen, noe som viser at konvolusjon er en svært regnekrevende operasjon. Figur (2) viser kjøretidene i sekunder for filterstørrelser mellom 5 og 100.

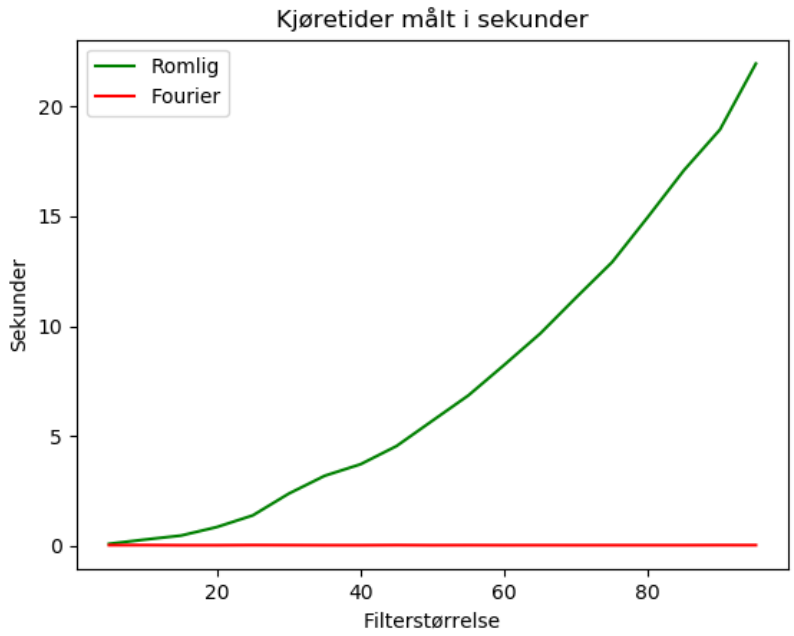


Figure 2: Kjøretider for fourier og romlig konvolusjon

2 Oppgave 2

2.1 Steg 6

Jeg beregnet entropien for både de transformerte og de kvantifiserte blokkene og verdiene er som følger:

- Entropi for kvantifisert bilde med kompresjonsrate q=0.1: 2.076
- Entropi for kvantifisert bilde med kompresjonsrate q=0.5: 2.071
- Entropi for kvantifisert bilde med kompresjonsrate q=2: 1.088
- Entropi for kvantifisert bilde med kompresjonsrate q=8: 0.4897
- Entropi for kvantifisert bilde med kompresjonsrate q=32: 0.1797

Entropien til et bilde er en nedre grense for hvor kompakt bildet kan lagres, dersom hver pikselintensitet er like sannsynlig med sannsynlighet lik $1/2^b$ hvil for eksempel entropien være lik b. Det betyr at i det spesielle tilfellet vil den nedre grensen for antall bits som er nødvendig for å lagre bildet være lik b. Det er altså det verste mulige tilfellet når vi skal komprimere bildet. Dersom alle pikslene er like derimot så er entropien lik null, da vi ikke trenger å lagre noe informasjon om pikslene siden vi allerede vet at alle pikslene har samme sannsynlighet, i dette tilfellet er entropien lik null. Entropien representerer også den gjennomsnittlige mengden informasjon i hver piksel. Antall bits i det originale bildet er $b = 8$. En entropi lik 2.886, som er det jeg fikk med $q = 0.1$ for det transformerte bildet, vil si at det minste antallet bits per piksel er $round(2.886) = 2.0$ bits. Vi ser også at entropien for de kvantifiserte bildene med $q = 8$ og $q = 32$ har entropi mindre enn 1. Dette betyr at minste antall bits er 0 og at hver piksel inneholder nesten ingen informasjon, som igjen vil si at sannsynligheten for de forskjellige verdiene er veldig like. Siden de fleste tallene jeg fikk ligger rundt 2 og 1 vil jeg nå bruke resultatet for $q = 2$ som et eksempel da denne er medianresultatet, størrelsen på bildet er $248x496$. Da blir kompresjonsraten lik

$$C = \frac{8 \cdot 248 \cdot 496}{1 \cdot 248 \cdot 496} = \frac{8}{1} = 8.$$

Som vil si at den relative redundansen til bildeinformasjonen er

$$1 - \frac{1}{C} = 1 - \frac{1}{8} = \frac{7}{8} = 0.875,$$

dette indikerer at 87.5% av informasjonen er redundant.
Plassen som trengs for å komprimere bildet er minst $1 \cdot 248 \cdot 496 = 123008$ bits. For større entropiverdier vil kompresjonsraten bli mindre, men jeg observerte også at tapet av informasjon ble større når q vokste. Dette gir mening ettersom flere verdier i matrisen vil bli null når q er større og hver piksel vil dermed inneholde mindre informasjon.

2.2 Steg 7

Figur (3) viser at det beste resultatet med minst informasjonstap var bildet hvor $q = 0.1$ ble brukt, altså den minste verdien av q. Det er tydelig at resultatene ble dårligere når q økte, hvor både $q = 8$ og $q = 32$ ga veldig tydelig dårlig resultat. Dette tyder på at DCT-metoden jeg har brukt muligens gir for små koeffisienter og at det dermed lønner seg å dividere med små tall. Som vi ser i figur 4 så er det tydelige artefakter i det rekonstruerte bildet, noe som får meg til å tro at det eneste bildet som egner seg for visning i fullskjerm er det rekonstruerte bildet med $q = 0.1$ i og med at vi også ser artefakter allerede med $q = 0.5$. Det man taper på å velge en lav q-verdi er kompresjonsraten, da en lavere q-verdi gir høyere entropi og dermed en lavere kompresjonsrate.

(a) Rekonstruert bilde med q=0.1

(b) Rekonstruert bilde med q=0.5

(c) Rekonstruert bilde med q=2

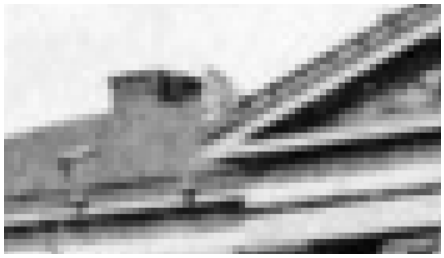
(d) Rekonstruert bilde med q=8

(e) Rekonstruert bilde med q=32

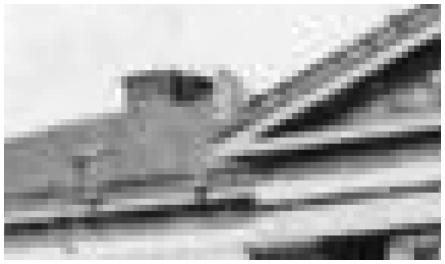
Figure 3: Rekonstruerte bilder med forskjellige verdier av q

(a) Originalt bilde

(b) Rekonstruert med $q = 2$



(c) Rekonstruert med q=0.1



(d) Rekonstruert med q=0.5

Figure 4: Zoomede bilder av samme seksjon for originalt og rekonstruerte bilder for de tre laveste q-verdiene