INF2310, Obligatorisk oppgave nr 2, Våren 2012

Oppgavesettet er på fire sider og består av to oppgaver.

Besvarelsen av hele denne og forrige obligatoriske oppgave må være godkjent for at du skal få anledning til å gå opp til endelig skriftlig eksamen i kurset.

Besvarelsene kan utarbeides i smågrupper på opptil to studenter, men det er ikke noe i veien for å arbeide alene. Studenter i samme smågruppe kan levere identisk besvarelse, men samarbeidet må framgå av navnene på forsiden av besvarelsen.

Av side 1 skal det fremgå hvem som har utarbeidet besvarelsen.

Det forventes at arbeidet er et resultat av egen innsats.
Å utgi andres arbeid for sitt eget er uetisk og kan medføre sterke reaksjoner fra Ifis side. Se http://www.mn.uio.no/ifi/studier/admin/obliger/

Besvarelsen skal leveres som én pdf-fil. Denne pdf-filen skal inneholde all programkode og alle figurer og bilder som er nødvendig for å besvare oppgavene og som du ønsker at skal bli vurdert. For oppgaver som krever implementering er det ikke tilstrekkelig å *kun* beskrive hvordan du ville/har implementert en løsning,
ei heller er det tilstrekkelig å sikte til programkode som ikke er inkludert i pdf-filen.

Besvarelsen skal sendes per e-post til gruppelæreren. Følgende er viktig:
- Besvarelsen skal leveres i form av én fil med følgende navn:

inf2310-oblig2-brukernavn.pdf
der ***brukernavn*** byttes ut med ditt eget brukernavn.
- Filen sendes som vedlegg i en e-post til runarfu@ifi.uio.no.
- Husk å sette «INF2310 Oblig2» i subjekt-feltet!
- Ved ønske om annen leveringsform, kontaktes gruppelærer.

Bildene det refereres til vil være å finne under:
http://www.uio.no/studier/emner/matnat/ifi/INF2310/v12/undervisningsmateriale/bilder/

Del 1 av oppgavesettet ble utlevert torsdag 3. mai, oppgave 2 mandag 7. mai 2012.
Innleveringsfrist er fredag 18. mai 2012.

Listene over godkjente oblig'er må være ferdig 1 uke før eksamen 6/6.
Utsatt innlevering er derfor bare aktuelt ved sykdom.
Studieadministrasjonen er rette vedkommende for slike saker.

Oppgave 1: JPEG-kompresjon

I denne oppgaven skal du implementere de viktigste delene av ikke-tapsfri JPEG-kompresjon. Du står fritt til å benytte det programmeringsspråket du selv ønsker – med unntak av lesning og skriving av bildefiler vil du kun trenge grunnleggende funksjoner som er tilgjengelig i alle brukbare programmeringsspråk.

Du skal lage en funksjon/metode som har to parametre:

1. et filnavn som spesifiserer plasseringen til bildefilen som skal benyttes, og
2. ett tall, *q*, som indirekte vil bestemme kompresjonsraten.

Funksjonen/metoden skal beregne omtrentlig hvor stor lagringsplass det angitte bildet vil bruke etter JPEG-komprimering, og finne hvilken kompresjonsrate dette tilsvarer.

Vi vil anta at inputbildet er et gråtonebilde med heltallsintensiteter i intervallet [0, 255] og har både en bredde og en høyde som er multipler av 8.

Bruk gjerne bildet *lena.png* for å teste implementasjonen din underveis.

Du skal bruke følgende algoritme/fremgangsmåte for å lage funksjonen/metoden:

a) Last inn bildet som den første parameteren spesifiserer.

b) Trekk fra 128 (gjør 0 den forventete gjennomsnittlige intensitetsverdien).

c) Begynn i øverste, venstre hjørnet og del opp bildet i 8x8-blokker.
Transformér hver blokk med 2D DCT.

Hint 1: Siden hver blokk har størrelse 8x8, kan vi forenkle den generelle formelen for 2D DCT på side 28 i forelesning 12 til:

$$F(u,v) = \frac{1}{4}c(u)c(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}$$

der:

$$c(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{hvis}\,\xi = 0 \\ 1 & \text{ellers} \end{cases}$$

Hint 2: 2D DCT-en av en 8x8-bildeblokk er en 8x8-transformblokk. Alle de transformerte 8x8-blokkene vil derfor utgjøre et bilde av lik størrelse som det opprinnelige bildet. Dermed kan vi lagre de transformerte 8x8-blokkene som et bilde der hver 8x8-transformblokk er plassert på samme sted som den 8x8-bildeblokken den er beregnet fra.

(oppgave 1 fortsetter på neste side)

d) Rekonstruér det opprinnelige bildet ved å invertere transformen du utførte i forrige deloppgave. Beregn altså 2D IDCT av hver transformert 8x8-blokk og sett blokkene sammen slik de opprinnelig var plassert.

Programmatisk verifiser at det rekonstruerte bildet er identisk som originalen.

Hint: Den forenklede formelen for 2D IDFT er:

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} c(u)c(v)F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}$$

der funksjonen *c* er definert som over. Pga. upresis flyttallsaritmetikk vil det generelt være behov for å avrunde de resulterende verdiene til nærmeste heltall.

e) La *Q* være følgende kvantifiseringsmatrise:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Ta utgangspunkt i de transformerte 8x8-blokkene fra deloppgave c og punktvis dividér hver blokk med *qQ*, dvs. produktet av tallparameteren *q* og kvantifiseringsmatrisen over. Avrund de resulterende verdiene til nærmeste heltall.

f) Dersom vi skulle fulgt JPEG-algoritmen videre så skulle vi nå separert det øverste, venstre elementet av hver transformert og kvantifisert 8x8-blokk, kalt et kvantifisert DC-element, fra de resterende 63 elementene i hver blokk. DC-elementene skulle blitt differansetransformert (på tvers av blokkene), og de 63 elementene skulle blokk for blokk blitt sikk-sakk-skannet og deretter løpelengdetransformert. Til slutt skulle differansene og løpelengdeparene blitt kodet, f.eks. ved bruk av Huffman-koding.

I stedet for å følge denne prosedyren vil vi påstå at den vil grovt sett resultere i en dataforbruk som tilsvarer entropien til alle elementene i alle de transformerte og kvantifiserte 8x8-blokkene. Beregn denne entropien og bruk den til å estimere hvor stor lagringsplass det angitte bildet vil bruke etter JPEG-komprimering og hvilken kompresjonsrate dette tilsvarer.

(oppgave 1 fortsetter på neste side)

g) Bruk det transformerte og kvantifiserte 8x8-blokkene fra deloppgave e til å rekonstruere en tilnærming av det opprinnelige bildet. Skriv dette bildet til fil.

Du skal også teste implementasjonen din. Dette skal du gjøre ved å anvende din funksjon/metode på bildet *lena.png* når du benytter hver av følgende verdier av tallparameteren *q*: 0.5, 1, 2, 4, 8, 16 og 32. Studér rekonstruksjonene, og bemerk når og hvor i bildet *du* først oppdager rekonstruksjonsfeilene «blokk-artefakter», «blurring» og «ringinger». Vurdér også for hvilke verdi(er) av tallparameteren *q* som *du* synes rekonstruksjonen er «god nok» for fremvisning av hele bildet på skjerm.

Hva skal leveres:

I. Komplett og grovt kommentert kildekode av funksjonen/metoden.
II. De rekonstruerte bildene av *lena.png* med de sju forskjellige verdiene av tallparameteren *q*.
III. En drøfting av komprimeringen av *lena.png*. Inkludér 1) dine oppdagelser om rekonstruksjonsfeilene, 2) din vurdering om av når rekonstruksjonen er «god nok» og 3) forklar hvorfor den estimerte kompresjonsraten øker med verdien av tallparameteren *q*.

NB!
Du kan benytte ferdige MATLAB-funksjoner til å lese/skrive fra/til fil.
Transformasjonene og entropiberegningen MÅ du implementere selv.

Oppgave 2: Terskling med eksponensiell entropi

I oppgave 4 i ukeoppgave nr 11 har du implementert en terskling ved hjelp av første ordens Shannon-entropi. Anta nå at vi representerer statistisk uvitenhet med [1-p(i)] istedenfor [1/p(i)], og at vi erstatter negeringen av 2-logaritmen med eksponensial-funksjonen, slik at vi får funksjonen:

$$B(t) = E_1(t) + E_2(t), \quad der$$

$$E_1(t) = \sum_{i=0}^{t}\left(\frac{p(i)}{P_1(t)}\right)\exp\left(1 - \frac{p(i)}{P_1(t)}\right), \quad P_1(t) = \sum_{i=0}^{t} p(i)$$

$$E_2(t) = \sum_{i=t+1}^{G-1}\left(\frac{p(i)}{P_2(t)}\right)\exp\left(1 - \frac{p(i)}{P_2(t)}\right), \quad P_2(t) = \sum_{i=t+1}^{G-1} p(i)$$

- Implementér en beregning av B(t) for alle mulige verdier av terskelverdien, t, i et gitt bilde.
- Anvend dette på bildet mona.png, og terskle bildet med den terskelverdien T_B der B(t) har sitt maksimum.
- Sammenlign – i en graf – funksjonen B(t) med funksjonen A(t) fra oppgave 4 i ukeoppgave 11, og diskuter de to tesklingsresultatene.

Lykke til!