SIGBE/15/3

a) Datakomprimering i frekvensplanet.
Talsekvensen i filen `http://www.users.abo.fi/~htoivone/courses/sigbe/signal.dat` representerar en audiosignal (vokalen 'a') som har samplats med samplingsfrekvensen 11025 Hz.

(i) Beräkning av diskreta Fouriertransformen.
Använd MATLAB-programmet `fft` för att bestämma diskreta Fouriertransformen $X(k)$ av sekvensen. Illustrera datasekvensen och dess spektrum grafiskt.
Verifiera även med MATLAB-programmet `ifft` att den inversa diskreta Fouriertransformen rekonstruerar den ursprungliga datasekvensen.

(ii) Representation med lågfrekventa komponenter.
Undersök hur väl datasekvensen kan representeras av lågfrekventa komponenter genom att approximera spektret med ett lågfrekvent spektrum $X_{\text{LF}}(k)$ där frekvenser $> f_1$ försummats (dvs $X_{\text{LF}}(k) = 0$ för de k som motsvarar frekvenskomponenter $> f_1$) *. Bestäm sedan en lågfrekvent approximation $x_{\text{LF}}(n)$ av signalen $x(n)$ som den inversa Fouriertransformen av $X_{\text{LF}}(k)$, och illustrera resultatet grafiskt genom att plotta en delsekvens av signalerna $x(n)$ och $x_{\text{LF}}(n)$ bestående av t.ex. 200 datapunkter. Använd frekvenserna $f_1 = 4000, 3000$ och 2000 Hz.
Beräkna även energin hos approximationsfelet $x - x_{\text{LF}}$ i procent av hela signalens energi, och kontrollera att Parsevals formel gäller. MATLAB-rutinen `norm` kan utnyttjas.
***OBS:** Observera dock att detta gäller endast för $k \le N/2$; för $k > N/2$ bör spektret satisfiera symmetriegenskapen (4.32); $X_{\text{LF}}(N-k) = X^*_{\text{LF}}(k)$.
Observera också att indexeringen av vektorn med Fouriertransformen X i MATLAB går från 1 till N, så att $X(0)$ har indexet 1, $X(1)$ har indexet 2, osv t.o.m. $X(N-1)$ som har indexet N. Frekvenskomponenten k har således indexet $k+1$, och frekvenskomponenten $N-k$ har indexet $N-k+1$.

(iii) Representation med dominerande frekvenskomponenter.
Undersök hur väl datasekvensen kan representeras av sina dominerande frekvenskomponenter genom att approximera $X(k)$ med ett spektrum $X_{\text{appr}}(k)$ där endast de största frekvenskomponenterna hålls kvar, dvs

$$X_{\text{appr}}(k) = \begin{cases} X(k), & \text{om } |X(k)| \ge c \cdot \max(|X(l)|, l = 0, 1, \ldots, N-1) \\ 0, & \text{annars} \end{cases}$$

Använd $c = 0.01, 0.05$ och 0.1.

Illustrera spektret $X_{\text{appr}}(k)$ grafiskt i de olika fallen och bestäm hur många frekvenskomponenter som hålls kvar i det approximerade spektret. Bestäm även motsvarande approximationer $x_{\text{appr}}(n)$ av signalen, och illustrera resultatet grafiskt såsom i (ii).
Beräkna även energin hos approximationsfelet $x - x_{\text{appr}}$ i procent av hela signalens energi.

b) Bildkomprimering.
Skriv ett program som komprimerar en 2-dimensionell signal $\{x(n, m)\}$ genom att dela signalen i 8×8 block och beräkna kvantiserade cosinus-transformer för de olika blocken i enlighet med exempel 5.1 i kompendiet. Bestäm andelen nollor i den komprimerade cosinus-transformerade signalen (detta ger en uppfattning om komprimeringsgraden). Skriv också ett program som rekonstruerar den ursprungliga signalen från den komprimerade transformen i enlighet med exempel 5.1 och jämför resultatet med den ursprungliga signalen. Testa programmen för en lämplig 2-dimensionell signal som representerar en bild.

För komprimering av 8×8 blocken kan Matlab-programmet `dct_compress88` användas, och rekonstruktion av 8×8 blocken kan utföras med programmet `idct_compress88`. En lämplig kvantiseringstabell finns i filen `Q.mat`. Metoden kan testas på bildsignalen `xx` i filen `lenna512.mat`. För bekväm generering av en bild som representeras av en 2-dimensionell signal kan programmet `show_image` användas.