

## DN2220, Tillämpade numeriska metoder I

### Lab3: SVD, DFT, DCT

I laborationen ska du pröva att komprimera bildmatriser med tre metoder: Singulärvärdesfaktorisering (SVD), diskret fouriertransform (DFT) och diskret cosinustransform (DCT). I kurskatalogen /info/tilnum1-09 finns filerna som behövs för labben.

#### 1. Singulärvärdesfaktorisering (SVD)

- a) I filen `svdmini.m` finns en  $6 \times 6$ -matris  $A$  med intensitetsbild och singulärvärdesfaktorisering av  $A$ . Utför komprimering med ett, två och tre singulärvärden medtagna.
- b) I `kallesvd.m` visas komprimering av  $31 \times 30$ -matrisen `kalle.mat`, och i `calvinsvd.m` komprimeras matrisen `calvin.mat` som har 200 rader och 200 kolumner. Hur många singulärvärden ungefär krävs för att få god skärpa?
- c) Testa Cleve Moler's matlabfunktion `imagesvd`, t ex med anropet `imagesvd('Flodhastar.jpg')` (fungerar snabbare på PC än på unixdatorer). Ett enda singulärvärde utnyttjas i första suddiga bilden (rank 1). Med 10 medtagna singulärvärden (rank 10) är bilden skönjbar. Pröva gärna funktionen på egen jpg-bild.

#### 2. Diskret fouriertransform (DFT)

- a) Beräkna med papper och penna DFT-vektorn till vektorn  $x = 8 * [1 \ 1 \ 0 \ 0 \ 0 \ 2]'$ .
- b) Utför DFT på  $x = 100 * [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0]'$  med  $Y = \text{fft}(x)$  och bilda  $Y_{\text{app}}$  med frekvenskomponenterna 4, 5 och 6 nollställda (DFT-lågpassfilter). Hur väl approximeras  $x$ -vektorn, alltså vad blir  $x_{\text{app}} = \text{ifft}(Y_{\text{app}})$ ? (Avrunda gärna  $x_{\text{app}}$  till heltal.)
- c) Diskret fouriertransform på en matris  $A$  erhålls genom att man först utför DFT på matrisens radvektorer och därefter DFT på kolumnvektorerna i den just erhållna matrisen. (Matlab har funktionen  $Y = \text{fft2}(A)$  för hela förfarandet.)  
I filen `dft2mini.m` behandlas samma  $6 \times 6$ -matris som i `svdmini` men nu med DFT. Komprimering erhålls genom att enbart låga

frekvenser tas med (lågpassfilter). Modifiera koden och visa resultatet av komprimering då DFT-matrisens element (3:5, 3:5) sätts till noll.

### 3. Diskret cosinustransform (DCT)

a) Utför DCT på  $\mathbf{x} = 100*[0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0]'$ . Bilda  $\mathbf{yapp}$  genom att sätta de fyra sista DCT-komponenterna till noll (DCT-lågpassfilter). Hur väl approximeras  $\mathbf{x}$ -vektorn nu, alltså vad blir  $\mathbf{xapp}=\mathbf{C}'*\mathbf{yapp}$ ? (Avrunda till heltal.)

b) Studera `dctNbild.m`: 2D-komprimering med DCT på en  $8 \times 8$ -matris dels med lågpassfilter dels med linjär kvantisering. Jämför med `dftNbild.m`: 2D-komprimering med diskret fouriertransform och lågpassfilter. Hur utfaller jämförelsen kod- och resultatmässigt?

c) I `hundhussedct.m` utförs DCT-komprimering på en  $16 \times 16$ -matris, (bildmatrisen är (`Hundhusse.mat`)). Studera koden så att du förstår och kan förklara algoritmen!

Det här är principen för JPEG-komprimering; bilden delas upp i  $8 \times 8$ -block och DCT med kvantisering utförs.

d) Slutligen ska du pröva programmet `flodhkoll3.m` som komprimerar flodhästfotot ovan (nu i svart-vitt) med alla metoderna: SVD och DFT med olika grad av trunkering samt DCT med kvantisering. Testa andra trunkeringsval på detta foto, men pröva gärna algoritmerna på ett eget foto!

Namn: .....

Lab3 godkänd!

Datum, lärare:.....