Gerd Eriksson
feb-09 TILNUM1

DN2220, Tillämpade numeriska metoder I

Lab3: SVD, DFT, DCT

This laboration deals with compression of image matrices with three different basic methods: Singular value decomposition (SVD), Discrete Fourier transform (DFT) and Discrete cosine transform (DCT).
The m-files and data files needed for the lab are found in the course catalogue /info/tilnum1-09.

1. **Singular value decomposition (SVD)**

   a) The file `svdmini.m` contains a 6×6-matrix A with an intensity image and with the SVD-computation of A. Your task is to use svd-compression with one, two and three singular values.

   b) The Matlab code in `kallesvd.m` shows data compression of the 31×30-matrix kalle.mat; in `calvinsvd.m` the matrix calvin.mat is much larger, 200 rows and 200 columns. How many singular values are needed around to get an image with reasonably good sharpness?

   c) Now, you shall try Cleve Moler´s function imagesvd with the call `imagesvd('Flodhastar.jpg')` (it works better on a PC than on a Unix computer). Only one singular value is used in the first blurred image (rank 1). With 10 listed singular values (rank 10) the image is almost visible! We encourage you to test the function imagesvd on your own jpg-photo.

2. **Discrete Fourier Transform (DFT)**

   a) Calculate with pen and paper the DFT-vector of the vector:
   `x = 8*[1 1 0 0 0 2]'`.

   b) Perform the DFT of `x = 100*[0 0 0 1 1 1 1 0]'` by the Matlab-function `Y=fft(x)`. Form Yapp with the frequency components 4, 5 and 6 put to zero (DFT low-pass filter). How good is the approximation of the x-vector – what is `xapp=ifft(Yapp)`? (Let the filtered xapp be rounded to an integer.)

   c) The discrete Fourier transform of a matrix A is obtained by first performing DFT on the rows of A, and then applying DFT on the

columns of the just obtained matrix. (Matlab has the function `Y=fft2(A)` to make it all.)
The file `dft2mini.m` deals with the same 6×6-matrix as `svdmini`, but now with DFT. As before, compression is obtained by using only low frequencies.
Modify the code and show the result of the compression when the elements `(3:5, 3:5)` of the DFT-matrix are set to zero.

3. **Discrete Cosine Transform (DCT)**

   a) Perform DCT on `x = 100*[0 0 0 1 1 1 1 0]'` Form yapp by putting the last four DCT-components to zero (DCT low-pass filter). How good is the approximation of the x-vector – what is `xapp=C'*yapp` ? (Let the filtered xapp be rounded to an integer.)

   b) Study `dctNbild.m` that performs two-dimensional DCT on an 8×8-matrix with two strategies: low-pass filtering and linear quantization.
   Compare the DCT-compressed N-image to the DFT-compressed N-image in `dftNbild.m`; you may also study and compare the Matlab codes. How is the outcome of the comparison?

   c) In `hundhussedct.m` DCT-compression is applied on a 16×16 matrix, (the image matrix is found in (`Hundhusse.mat`). Study the Matlab code, so that you can understand and explain the algorithm.
   This is the basic idea of JPEG-compression: the image is divided into a number of 8×8 pixel blocks and DCT-compression with quantization is applied on each block.

   d) Your last task is to try the program `flodhkoll3.m` which compresses the hippo photo (used above in imagesvd, now black-white) with our three basic methods: SVD with truncated singular values, DFT with truncated frequency components, and DCT with quantization. Test other choices of truncation/quantization on the hippo. As above we encourage you to try the algorithms here on your own photo and compare the quality of the compressed images.

Name: ....

Lab3 approved!   Date, teacher:..