EQ2330 Image and Video Processing - Project 2

Stefano Imposcopi
imoscopi@kth.se

Denis Kulicek
kulicek@kth.se

December 9th, 2015

SUMMARY

The goal of this project is to investigate image transforms and their energy compaction capabilities for the goal of compressing images. A comparison is made between the block Discrete Cosine Transform (DCT) and the Digital Wavelet Transform (DWT), which are the basis for the JPEG and JPEG2000 compression algorithms respectively.

Uniform quantization is then applied to the coefficients in the transformed domain to find a relationship between number of bits required and quality of reconstruction. To estimate the number of bits required we use entropy as a measure, so we obtain a lower bound of the bitrate required with an ideal Variable Length Coder.

DCT-BASED IMAGE COMPRESSION

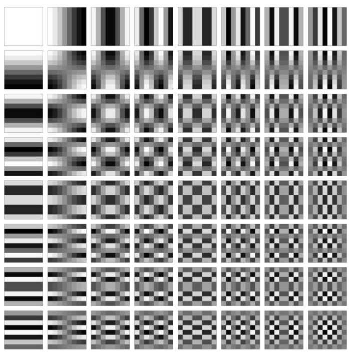
The DCT is the transform at the core of the compression algorithm JPEG. The image is divided into blocks of size 8x8 and each block is processed separately. Since the DCT is a Linear transform, it can be implemented as a matrix multiplication $T = A\,I\,A^T$, and because it is orthonormal transform it is perfectly reversible (that is before the quantisation is performed).

Matrix A is the DCT transform matrix of size 8x8. Code for generating DCT coefficients is shown in the Appendix.

$$A = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

Figure 1.1 - DCT coefficients for transform matrix A

In the essence this transform actually represents coding 8x8 blocks of an input image as summation of DC and AC coefficients shown on the righthand side[1].



As seen in Figure 1.2 below - most of the energy of every block is compacted in the DC and low frequency coefficients. This makes the DCT an ideal transform for image compression.

[1] https://en.wikipedia.org/wiki/Discrete_cosine_transform#/media/File:DCT-8x8.png

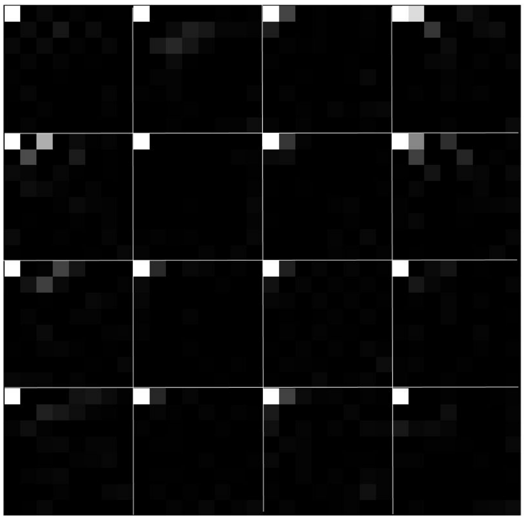


Figure 1.2: 16 samples of DCT transform blocks. The DC component lie in the upper left corners

FWT-BASED IMAGE COMPRESSION

For the wavelet part we implemented analysis and synthesis filter banks and we decided to use a Daubechies 8-tap filter to generate orthonormal filters. We followed the filter and downsample approach for analysis and upsample and filter for synthesis. To extend to the 2D case we used the block scheme in Fig 2.1.

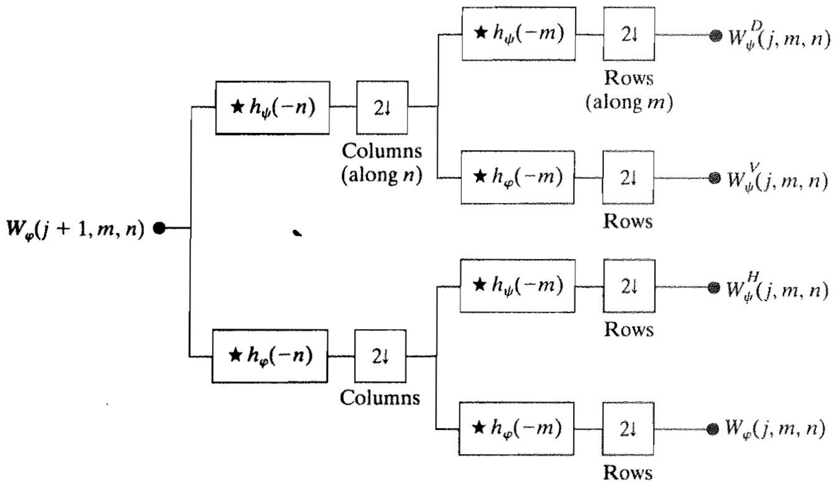


Figure 2.1 - Block scheme for image DWT using filter and downsample approach

Where h_φ and h_Ψ are the scaling and wavelet function to extract approximation and detail coefficients respectively. The result of 4-scale analysis using our implementation is visible in the Figure 2.2. The values are very saturated due to stretching of the values to highlight the details coefficients that would otherwise be barely visible. Most of the energy is packed in the very low resolution low-pass part of the image, while details coefficients at higher resolutions are mostly black images that can be compressed efficiently with techniques like run-length coding.

The Daubechies filters are orthonormal and thus allow for perfect reconstruction in theory. In practice this is achieved only when edge problems arising from convolution of finite length signals are handled using padding. In our implementation we used **symmetric padding** on both sides of the signals. Despite this we could not get perfect reconstruction, after downsampling by 2 problems arise that lead to shifts between the reconstructed image and original image, as well as problems at the edges of the image objects. We experimented with many different paddings and other ways, but could not fix the problem.

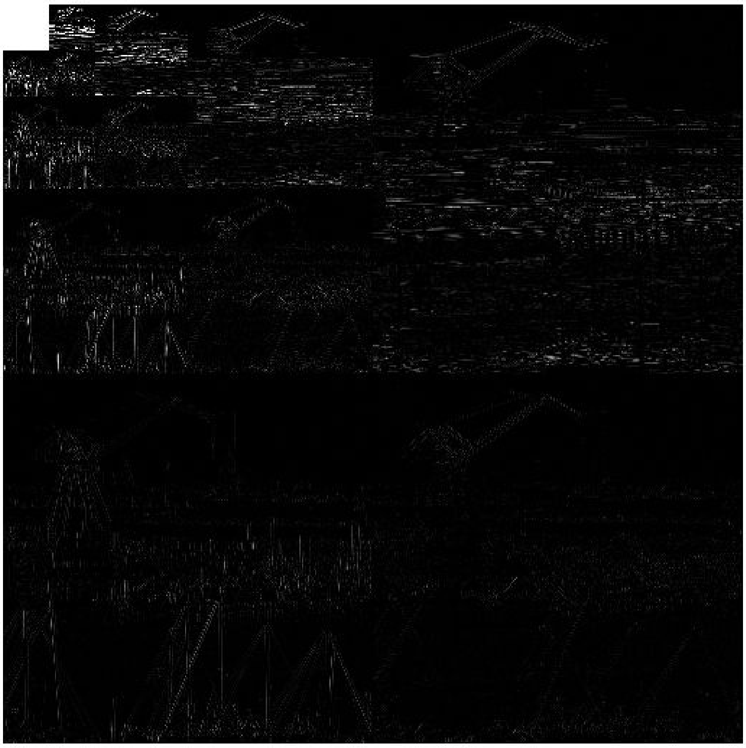


Figure 2.2. 4-scale wavelet decomposition of the Harbour image

The problems arising in reconstruction with our implementation are visible in Fig 2.3

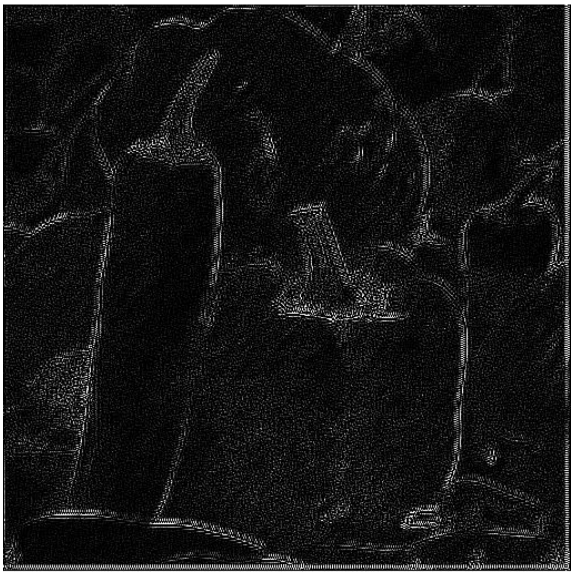


Figure 2.3: Thresholded error image showing the edge problems in reconstruction with our implementation.

Because of this we used the wavelet analysis and synthesis filters provided in the matlab wavelet toolbox for the remaining parts of the project.

3 UNIFORM QUANTIZATION AND BIT-RATE ESTIMATION

To compress the image we down-quantized the image, introducing big rounding artifacts over all the coefficients using a mid-term uniform quantizer. The input-output relation of the quantizer is show in the following Figure 3.2

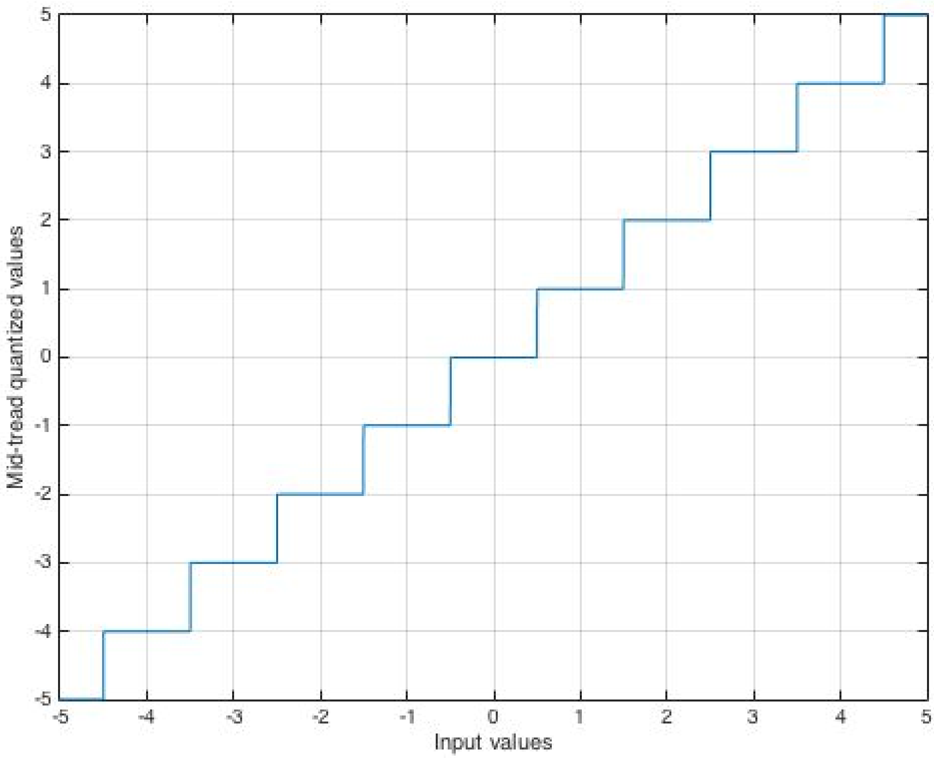


Figure 3.2 - The example input-output characteristic function of our quantizer

All values in a range are mapped to the midpoint of the range, and all quantization steps have the same length and thus are equally spaced. This is the simplest and most naive way of quantizing a signal. For the distortion and bitrate estimation we used quantization steps ranging from 2^0 to 2^9. A smaller step corresponds to finer quantization, and thus more levels, higher bit-rate, and intuitively higher quality of reconstruction. Figure 4.2. shows how different quantization levels affect reconstruction PSNR. More granular quantizer values are the ones with higher PSNR.

Then to evaluate the bit-rate required for every mode of quantization we assumed the use of an ideal Variable Length Coder that approaches the lower bound given by information theory. So we used the Shannon Entropy to estimate the number of bits required. The **Shannon Entropy** of the source is the lower-bound of all achievable rates, so it would be the **ideal average bit-length**. The bit-length can be fractionary even if codewords have an integer amount of bits, because of the different probabilities of codewords (given by different probabilities of values for the transform coefficients).

In the DCT case coefficients are encoded separately but with the same codewords for every block, so we computed the total probability distribution of DCT8 coefficients and estimated the entropy for different quantization steps. Intuitively **high frequency coefficients should have a lower entropy** being almost always very close to 0. This was confirmed by our experiments and is shown in Figure 3.3 below. Upper left coefficient represents the DC coefficient, which has the highest entropy overall, hence carries most information.

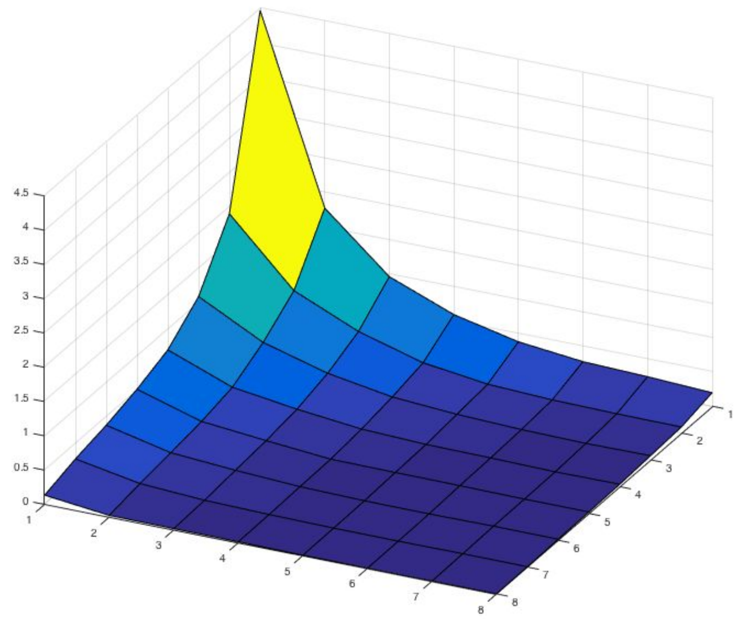


Figure 3.3 - Average entropy of all DCT coefficients of images: boats, harbour and peppers.

In the FWT case we assumed different codes for every subband, so we estimated the entropy over every subband (approximation, vertical, horizontal and diagonal details) and then averaged the results from the subbands together to get the total entropy of the DWT coefficients. In this case as well the **approximation coefficients have higher entropy than the detail coefficients**. For example without quantization the entropy of the approximation coefficients is 7,678, while the diagonal details have a significantly lower entropy of 4,232.

Estimated the bit-rates required, we reconstructed the images for all different quantization steps and computed Mean-Squared-Error[2] and Peak-to-Signal-Noise-Ratio[3] as a measure of quality in the reconstructed images.

Compare d with the mean squared error between the original and the quantized DCT coefficients. How do they compare and why?
The DCT is an orthogonal transform that preserves the energy between space and frequency domain, thus the results of MSE are the same both on reconstructed images and quantized DCT coefficients. So technically there is no need of transforming back to the space domain, but we did it to inspect the reconstructed images visually

4 COMPARISON OF RESULTS

The reconstructed images for both transforms are compared in fig 4.1. For both transforms a quantization step of 64 was used, corresponding to a quite coarse quantization. The DCT outperforms the DWT in terms of perceptual quality. Also the estimated bit-rate required for the coefficients is lower in the DCT case, suggesting a much better trade-off for the DCT.

---

[2] MSE description https://en.wikipedia.org/wiki/Mean_squared_error
[3] PSNR description https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

This is because the DCT is the transform that comes closer to the ideal KLT that packs the highest amount of energy in the smallest amount of coefficients. Thus it is ideal and performs best in the case of simple uniform quantization of coefficients.

Figure 4.1. DCT with step size 64 (left) vs. DWT with step size 64 (right) The Entropy estimation gives a lower bound of 0.25 (DCT) and 0.78 (DWT) bits/coefficient respectively

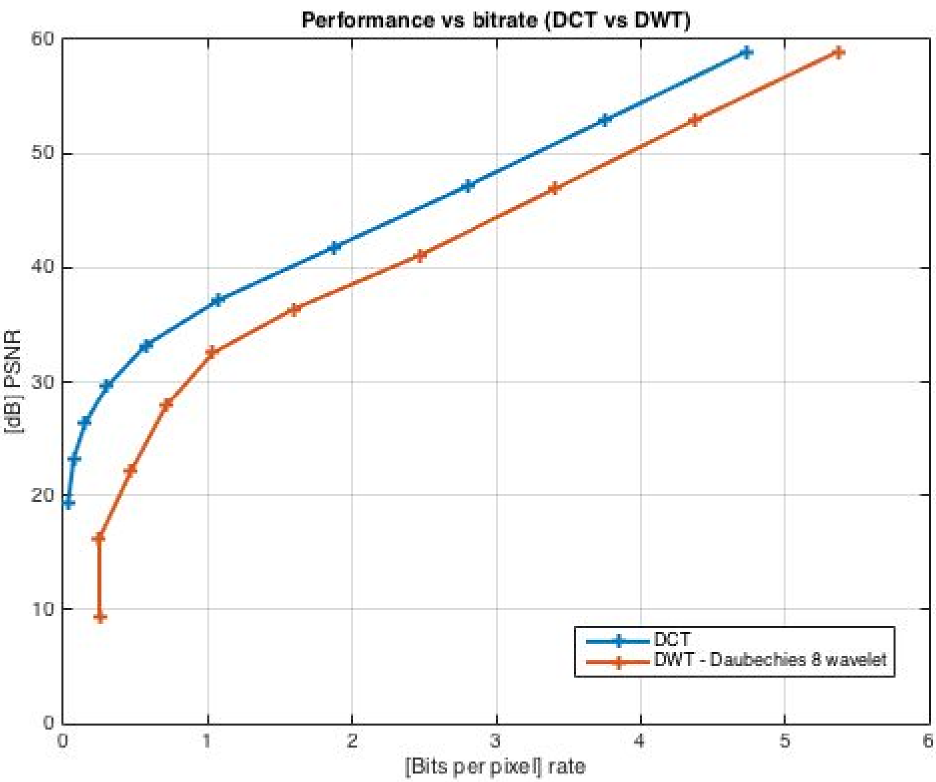


Figure 4.2. Performance-bitrate plot of DCT and DWT results

Performance in PSNR vs bitrate can be seen in Figure 4.2. Again, as expected the performance of the DCT is higher than the DWT and overall the tradeoff performance/bitrate weighs in favour of the DCT. Also it should be noted that the gain in performance is of roughly 6dB per added bit/pixel. This makes perfect sense for uniform quantization and confirms the result by Shannon for the Rate Distortion Curve of a gaussian source with MSE distortion.

The same dataset (composed of the all dct coefficients of images: *peppers*, *boat* and *harbour* combined) was also used in this calculation to maintain consistency.

SUMMARY

To conclude we proved the great usefulness of the DCT as a very energy-compact transform for image compression. Good perceptual quality images could be reconstructed with a significant reduction in bit-rate, but overall the DCT seemed to perform better. This should not in any way downplay the usefulness and superior results that can be achieved with DWT. The DWT is a multiresolution analysis transform that has the advantage of having the possibility of compressing coefficients selectively based on the resolution of interest. Moreover it is a global transform that does not need to split the input image in blocks. Thus it has been adopted in JPEG2000 where it can avoid the blocking artifacts of JPEG. Also the performance differ because we used a very basic uniform quantization to reduce bit-rate. We did not design a quantization method optimized for each type of transform. In that case the DWT based approach can outperform the DCT approach.

APPENDIX

The project was done collaboratively by both authors, while discussing the algorithms used throughout the process. Some of the learning outcomes are getting to know nuts and bolts of DCT and DWT image compression implementation details. Part of the code mentioned in the report is shown below, and the remaining source code is attached in a zip folder.

DCT coefficient generation

```
function[Y] = dctz2(X)
% Discrete Cosine Transform of 8x8 block
%   X input matrix
%   Y output matrix

% Making sure it works for both matrix inputs and structs (blockproc)
if(isstruct(X))
    M = X.blockSize(1);
    X = X.data;
else
    M = size(X,1);
end

if (M~=8)
    error('unexpected block size, should be 8x8');
    Y = [];
    return
end

for k=0:M-1
    for i=0:M-1
        alpha = sqrt(((i~=0)+1)/M);
        c = cos(((2*k+1)*i*pi)/(2*M));
        A(i+1,k+1) = alpha * c; %stupid matlab non-zero indexing
    end
end

Y = A*X*A';
% the difference between DCT2 and IDCT2 (its inverse) is this part:
% to get the x back we would have to do: X = A’*Y*A;
end
```

Some of the other functions used

```
uniform_quantizer = @(x,ssize) round(x/ssize)*ssize;
mse = @(x,y) sum(sum((y-x).^2))/(size(y,1) * size(y,2));
PSNR = @(D) 10*log10(255^2./D);
```

References

[1] *Rafael C. Gonzalez and Richard E. Woods*, Digital Image Processing, Prentice Hall, 2nd ed., 2002

[2] *Ruye Wang,* Definition of DCT - http://fourier.eng.hmc.edu/e161/lectures/dct/node1.html

[3] *Pascal Getreuer,* Image processing with Matlab - http://www.getreuer.info/tutorials/matlabimaging