EQ2330 Image and Video Processing

EQ2330 Image and Video Processing, Project 2

Yuqi Zheng yuqizh@kth.se
Ernan Wang ernan@kth.se
Shuyi Chen shuyic@kth.se

December 7, 2021

Summary

This project investigates two image transforms of image compression algorithms, the discrete cosine transform (DCT) and the fast wavelet transform (FWT), respectively. Their performances are evaluated and a comparison is made.

For the DCT, the most prevalent DCT-II is applied. DCT and inverse DCT of block size 8×8 are implemented and the value of matrix A is obtained. For the fast wavelet transform (FWT), the one-dimensional two-band analysis and synthesis filter bank function are realized. The Daubechies 8-tap filter is used for studying the FWT and the inverse FWT of an arbitrary scale. To evaluate two algorithms, the uniform quantizer of the coefficients is performed and the quality of the reconstructed images is measured by the Peak Signal to Noise Ratio (PSNR).

By evaluating the PSNR, we conclude that, although both algorithms work well in decomposing and reconstructing images, DCT transform performs better than FWT transform in terms of reconstruction quality because it is closer to the ideal KLT transform.

1 Introduction

In the last decades, the demand for image compression has surged. Image compression aims at eliminating data redundancy and irrelevancy to decrease the storage and transmission costs. Transform converts the input image into another kind of data in which the coefficients are encoded. In this project, two image compression techniques that we investigate are DCT and FWT. They are the most widely used algorithms because of their ability to compress image with a small number of coefficients.

The DCT represent a sum of different cosine functions by a finite sequence of data points. It is a Fourier-related transform and similar to the Discrete Fourier transform (DFT), but only using real numbers. The DCT-II is probably the most commonly used form, often simply referred to as the “DCT”.

The FWT is a computationally efficient form of the discrete wavelet transform (DWT). Based on an orthogonal basis of wavelets, it turns a signal into a sequence of coefficients in the time domain. It can be easily extended to multidimensional signals like images.

Via quantizer, all value in a single step-sized interval can be rounded to the representative interval value. We quantize the coefficients using a mid-tread uniform quantizer without threshold characteristic. The quality of the reconstructed images is measured by the Peak Signal to Noise Ratio (PSNR).

PSNR express the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of its representation.

2 System Description

2.1 DCT-based Image Compression

In this project, we use the DCT-II of a block size $M \times M$ to process images ($M = 8$). It is separable and orthonormal. The $M \times M$ transform matrix A contains elements

$$a_{ik} = \alpha_i \cos(\frac{(2k+1)i\pi}{2M}) \quad \text{for } i, k = 0, 1, ...M-1, \tag{1}$$

with

$$\alpha_0 = \sqrt{\frac{1}{M}}, \tag{2}$$

$$\alpha_i = \sqrt{\frac{2}{M}} \quad \forall i > 0. \tag{3}$$

The DCT transform coefficients of a $M \times M$ signal block x can be expressed by $y = AxA^T$ and the inverse DCT (IDCT) transform can be expressed by $x = A^T yA$.

2.2 FWT-based Image Compression

We choose the direct implementation of the filter bank for the FWT. At first, we implement a one-dimensional two-band analysis filter bank function as described in Figure 1. This step produces two sets of coefficients: approximation coefficients cA_1, and detail coefficients cD_1. These vectors are obtained by convolving s with the low-pass filter Lo-D for approximation, and with the high-pass filter Hi-D for detail, followed by dyadic decimation.

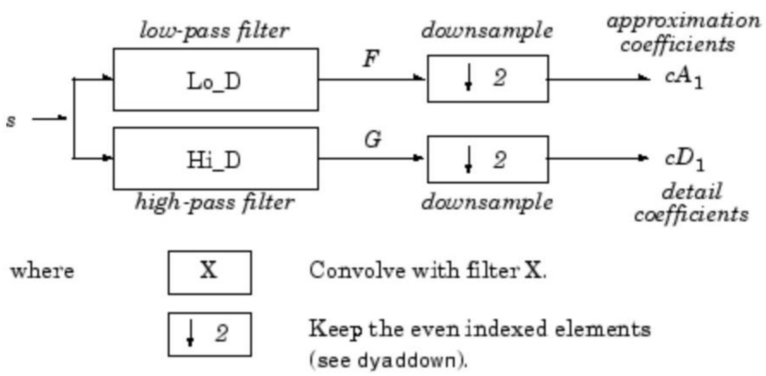


Figure 1: 1-D DWT

Conversely, starting from cA_j and cD_j, the IDWT reconstructs cA_{j-1}, inverting the decomposition step by inserting zeros and convolving the results with the reconstruction filters, as described in Figure 2.

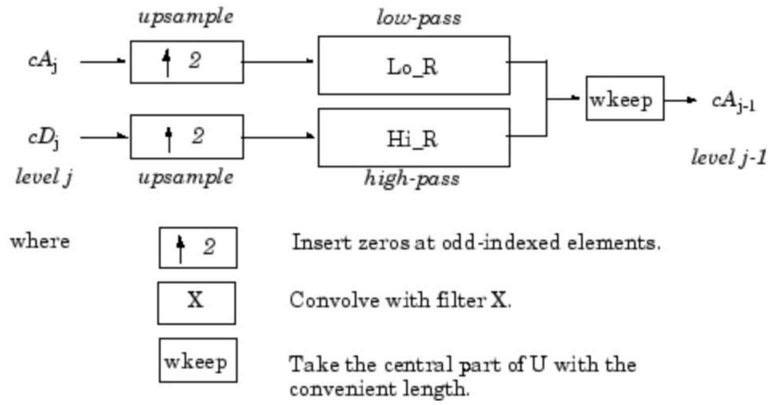


Figure 2: 1-D IDWT

We implement a two-dimensional FWT (inverse FWT) by recursive application of the analysis (synthesis) filter bank designed before and with the Daubechies 8-tap filter. The 2-D FWT and 2-D IFWT are presented as the two following Figures 3 and Figure 4. This kind of 2-D DWT leads to a decomposition of approximation coefficients at level j in four components: the approximation at level $j+1$ and the details in three orientations (horizontal, vertical, and diagonal).

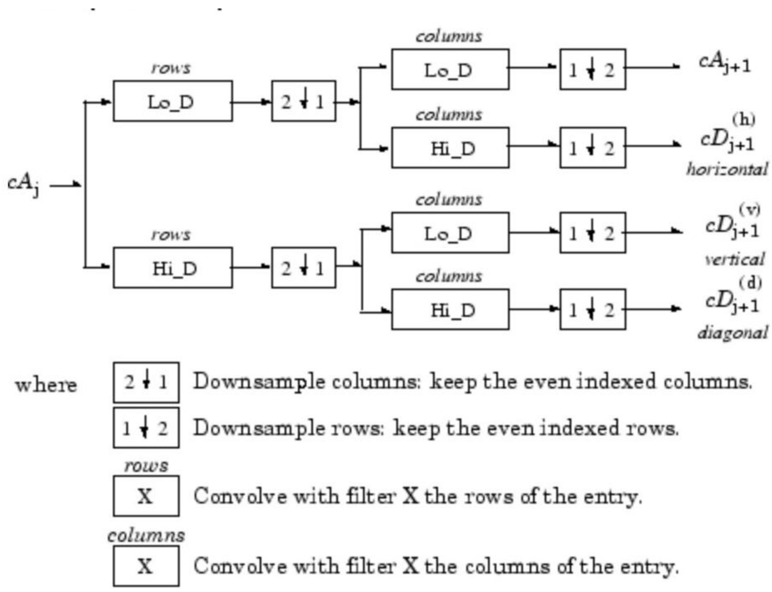


Figure 3: 2-D DWT

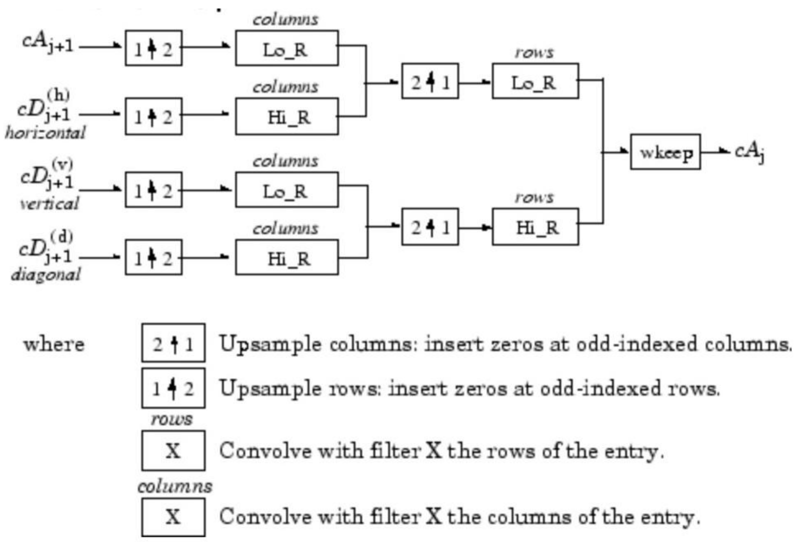


Figure 4: 2-D IDWT

To implement the FWT and the inverse FWT of arbitrary scale, we should repeatedly apply 2-D DWT and 2-D inverse DWT to the image. For example, if we want to implement the FWT and the inverse FWT of n scale, we should apply n times 2-D DWT and 2-D inverse DWT to this image. So, when the scale is equal to 2, the 2-D wavelet tree has the following form [1].

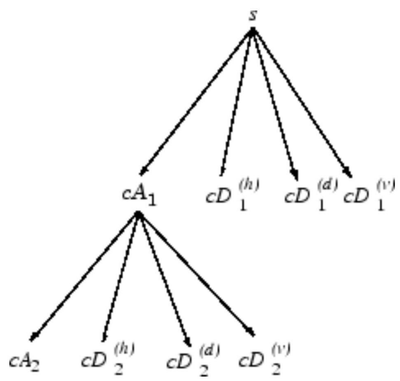


Figure 5: 2-D wavelet tree, scale=2.

To eliminate the border effects that occur when filtering, we firstly extend the signal periodically and truncate the output coefficients finally. The extension of the signal can be presented as follows.

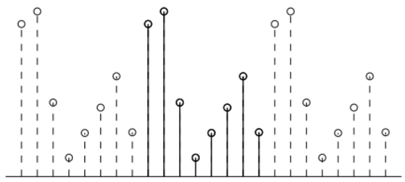


Figure 6: The extension of the signal.

In MATLAB, we use 'wextend' function and set the third argument in this function to be 'per'.

2.3 Quantization and Evaluation

To quantize the transform coefficients, we utilize a uniform mid-tread quantizer which has a zero-representative level without threshold characteristic. The quantizer maps all values in a range to the midpoint of this range. Besides, all quantization steps are of equal size. A smaller step is corresponding to a finer quantization. The quantized element can be expressed by

$$x_{quantized} = \left\lfloor \frac{x}{\delta} + \frac{1}{2} \right\rfloor \cdot \delta, \tag{4}$$

where $\lfloor . \rfloor$ denotes a function which maps a number to the largest integer no larger than its argument and δ denotes the step size.

To evaluate the quality of the reconstructed 8-bit images, we introduce the Peak Signal to Noise Ratio (PSNR), which is defined as follows,

$$PSNR = 10 \log_{10}(\frac{255^2}{d}) \quad [\text{dB}], \tag{5}$$

where d denotes the mean square error (MSE) between the original and the reconstructed images.

To evaluate the performance of the compression, we need to consider the bit-rate required as well. To estimate the bit-rate to encode the quantized transform coefficient, we assume that we use the ideal code word length of a variable length code (VLC), which leads to the lower bound of all possible bit-rates. According to $Shannon's\ Information\ Theory$, the entropy of the coefficients, i.e., the entropy of the code word length, is the ideal bit-rate, since the possibility of the coefficients is mapped to the length of the corresponding code.

In DCT, we use different VLCs for each coefficient within each block while the pairwise coefficients use the same VLC in each block. On the other hand, in FWT, we encode each coefficient individually. For each of the subbands, we use different VLCs.

3 Results

3.1 DCT-based Image Compression

Firstly, we generate a 8 × 8 DCT coefficients matrix shown as below.

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

Table 1: 8×8 DCT coefficients matrix.

Then, we design a uniform mid-tread quantizer without threshold characteristic. The graph of the quantizer function is shown in Figure 7.

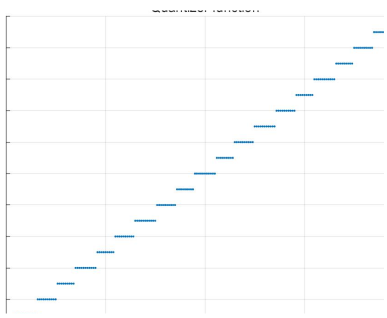


Figure 7: Quantizer function.

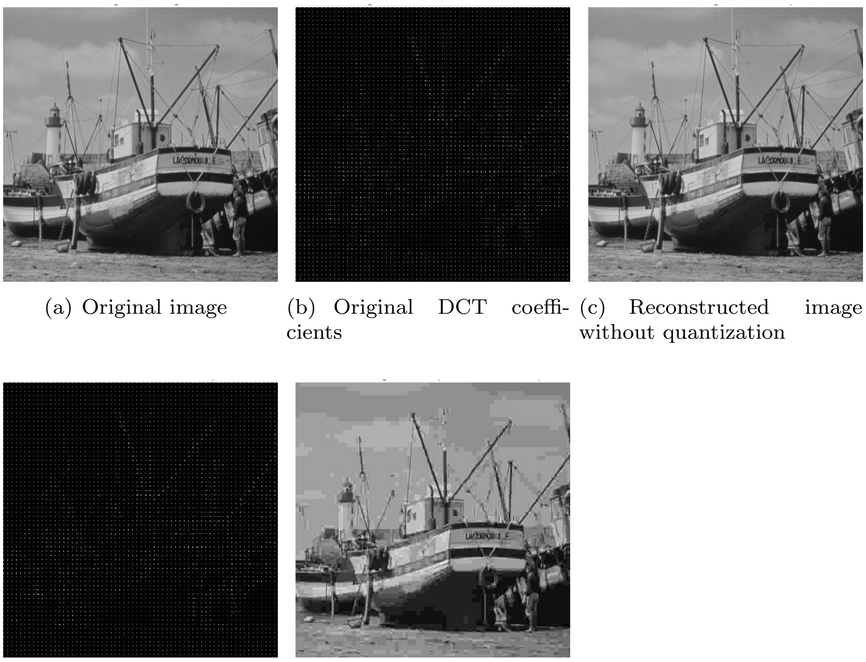
Then, we look into a specific image. We implement the DCT of block size 8×8 for the original image and thus obtain the DCT coefficients of it. Next, we utilized the quantizer as above to quantize the DCT coefficients with the step size 128. We reconstruct the image with the original coefficients and the quantized coefficients, respectively. Figure 8(a) to Figure 8(e) show the results.

We can observe significant degradation caused by quantization, which is also shown by the average distortion d. In this case, d=126.4270. In addition, we find that d is equal to the MSE between the original and the quantized DCT coefficients. The explanation is following.

Let $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}'$, $\mathbf{y}'$ denote the original image, original coefficients, reconstructed image and quantized coefficients. The transformation is given by $\mathbf{y} = \mathbf{A}\mathbf{x}$ and $\mathbf{x} = \mathbf{A}^T\mathbf{y}$ We have

$$\begin{aligned} MSE &= \frac{1}{64}\left(\mathbf{x}-\mathbf{x}'\right)^T\left(\mathbf{x}-\mathbf{x}'\right) \\ &= \frac{1}{64}\left(\mathbf{y}-\mathbf{y}'\right)^T \mathbf{A}\mathbf{A}^T\left(\mathbf{y}-\mathbf{y}'\right) \\ &= \left(\mathbf{y}-\mathbf{y}'\right)^T\left(\mathbf{y}-\mathbf{y}'\right). \end{aligned} \tag{6}$$

This relation holds because DCT is an orthonormal transform. In the perspective of energy conservation, it is also reasonable as the only loss of energy is introduced by quantization. Therefore, it is sufficient to evaluate the quantization error when evaluating MSE of reconstructed image. This proof is true for FWT because FWT is orthonormal as well.



(a) Original image (b) Original DCT coefficients (c) Reconstructed image without quantization

(d) Quantized DCT coefficients (e) Reconstructed image with quantization

Figure 8: DCT-based image compression.

3.2 FWT-based Image Compression

We implement a 1-D two-band analysis filter bank and a 1-D two-band synthesis filter bank with the direct method. Then, we implement the 2-D FWT and IFWT by recursive application of the 1-D filter banks, using the Daubechies 8-tap filter. The wavelet coefficients for scale 4 of the image harbour is presented by Figure 9.

Figure 9: Wavelet coefficients at scale=4.

Then, at scale 4, we implement the FWT for the original image and thus obtain the wavelet coefficients of it. After quantizing the wavelet coefficients with the quantizer designed in **3.1** with the step size 2, we reconstruct the image with the original coefficients and the quantized coefficients by IFWT, respectively. Figure 10(a) to Figure 10(e) show the results.

We find that the average distortion d is also equal to the MSE between

(a) Original image

(b) Original FWT coefficients

(c) Reconstructed image without quantization

(d) Quantized FWT coefficients

(e) Reconstructed image with quantization

Figure 10: FWT-based image compression.

the original and the quantized wavelet coefficients. The proof is the same as that in **3.1**. In FWT, the energy is concentrated at low frequencies, i.e., the approximate part.

Then, we look into the difference between DCT and FWT. We implement these two transforms for three images, with the quantizer step size over the range 2^0, 2^1, 2^2, ..., 2^9. Then, we obtain the PSNRs and bit-rates. The rate-PSNR curve is as shown in Figure 11.

In general, the smaller the quantizer step size is, the higher bit-rate required is, the higher the PSNR is, and thus, the high quality of reconstruction is. From Figure 11, we observe that DCT outperforms DWT in terms of the reconstruction quality. That is because DCT is closer to the ideal KLT that packs the highest amount of energy in the smallest amount of coefficients. Also, it can be noticed that the gain in performance is of roughly 6 dB per added bit/pixel for large bit-rates, which is confirmed by $Shannon's\ Information\ Theory$.

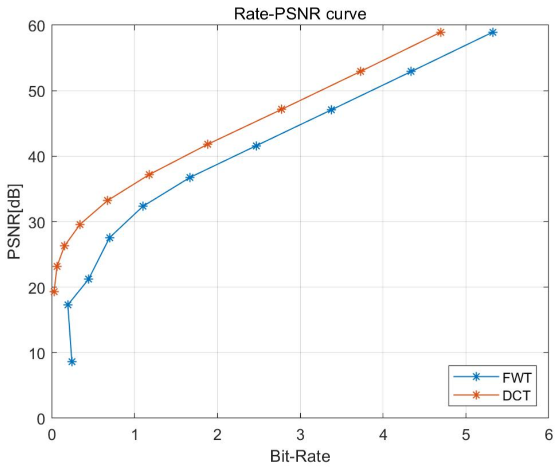


Figure 11: PSNR versus bit-rates curve of DCT and FWT transform.

4 Conclusions

In this project, we implemented DCT and FWT to perform the image compression, including the computation of the transform matrix, the quantization of the transform coefficients, the reconstruction based on the coefficients, and the evaluation of the transform in terms of PSNR and bit-rate. Both algorithms work, but overall, DCT performs better, which is mainly because it is closer to KLT. In other words, DCT compacts as much energy into as few coefficients as possible. However, FWT does have its merits as it divides the image into non-overlapping multiresolution subbands. In FWT, the information is concentrated on more components that can be affected less sensitively by quantization than in DCT. Furthermore, FWT avoids blocking artifacts that can happen by dividing the image into blocks in DCT. Their specific advantages make them both widely applied in corresponding situations.

Appendix

Who Did What

Report

Summary and Introduction: Shuyi Chen, Ernan Wang
System description and Results: Yuqi Zheng, Ernan Wang
Conclusion: Yuqi Zheng

Code

Yuqi Zheng

MatLab code

The MATLAB code is attached as images.rar.

References

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