**Report 7: Transform Image Compression**

**Ge Yunhao （116020910017）**

**[Problem 07]**

1. Investigate image compression based on DCT. Divide the image into 8-by- 8 subimages, compute the two- dimensional discrete cosine transform of each subimage, compress the test image to different qualities by discarding some DCT coefficients based on zonal mask and threshold mask and using the inverse discrete cosine transform with fewer transform coefficients. Display the original image, the reconstructed images and the difference images.

2. Investigate image compression based on wavelets.

**Program:**

**f7():**  Main function

**zonalMask():** Add a zonal mask to an image.

**threshold():** Add a threshold mask to an image.

**DCT():** Divide the image into subimages, compute DCT transform of each subimage, add a mask and then use IDCT with fewer transform coefficients.

**waveletTransform():**

Perform a wavelet transform based on specific wavelet. Compute the compressed image and difference image. Parameter ‘wavelet’ can be ‘haar’, ‘db4’, ‘sym4’ and ‘bior6.8’. Parameter ‘thre’ means the threshold to compress the image.

**Solution：**

In this report, we introduce three kinds of image compression methods respectively based on zonal mask, threshold mask and wavelets. And there are various wavelets and we present four classes.

The first two methods are based on discrete cosine transform, that is DCT. We firstly divide the original image into m-by-m subimages, and in experiments we specify 8 to m, then apply DCT operations. Next masks are used to replace some values in transformed images with 0. The zonal mask can be regarded as a selector that only choose upper left triangle elements because these values near the origin are usually larger than others. As for threshold mask, it preserves top larger values. In implementation, we can realize it by firstly sorting then selecting. Both two methods are aimed to choose values to conserve information as much as possible. And then, we use IDCT to transform processed result into spatial domain. To get an explicit comparison with original one, we depict a difference image about original one and compressed one.

The wavelet transform conduct convolution operations based on a specific form of waveform. In a recursion, an image is partitioned four parts, separately representing principle information and detailed information horizontally, vertically and diagonally. What we need to do is compress the detailed information which contains last three parts. In order to get a more compact result, we can conduct several recursions. Each recursion subdivide the image into four parts and compress three of them. And then we can use processed individuals to transform back to original size. Similarly, we make a difference image. In the experiment, we investigate four kinds or wavelets including Haar, Daubechies, Symlet as well as biorthogonal Cohen-Daubechies-Feauveau and decompose original image to 3 levels, then truncate coefficients in terms of a threshold. In Matlab, there are encapsulated functions and we use them directly for convenience.

**Result and Analysis**

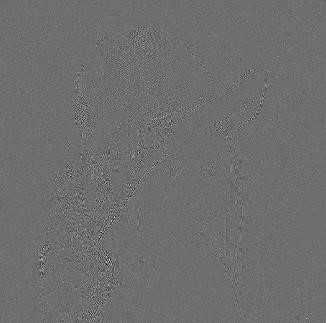
# (a)

Original image, image with threshold mask and difference image



Original image, image with zonal mask and difference image



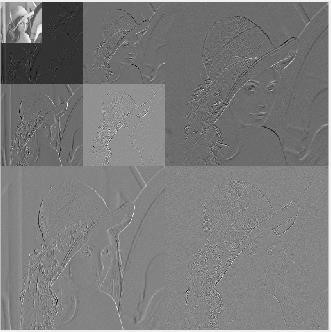


It’s obvious that the zonal mask has a better performance since its difference image contains less information, and this means that it losses less information during compression.

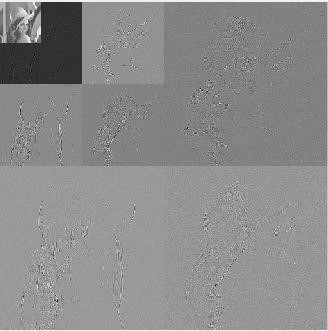
# (b)

Each wavelet corresponds with three images, wavelet transform of the image, reconstructed image and difference image. The input is just the same as above. Note that it may need some dimension operations when using other three wavelets except Harr.

# Harr

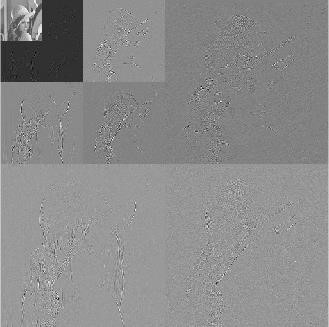


# Daubechies



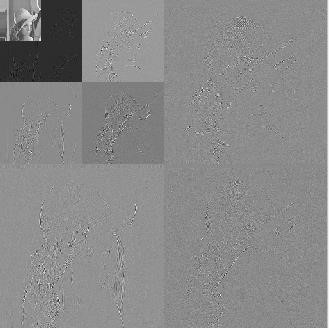


# Symlet





# Biorthogonal Cohen-Daubechies-Feauveau



We can find that Daubechies has the best performance because its reconstructed image is mostly similar to original one. Then the results of Symlet and biorthogonal CohenDaubechies-Feauveau are not bad. The worst case is Haar and there is distinct grid noise on image.