Transformer Theory – Image Compression

Emil Gustafsson, MT2A and Samuel Rising, MT2B

Jan 29, 2016

1 Introduction

This report covers the fundamentals of image compre-ssion, the algorithms and math-processes that occur behind the scenes. The specific concepts analyzed are the general idea of transform coding, Fourier trans-forms and low-pass filtering using the Manhattan distance function.

2 Background

The general idea of transform coding, or more specifically digital signal processing, is to perform an operation on the image while it is in the frequency domain. The main rea-sons for performing transform coding is for a more efficient quantization and data compression. Something to note is that the transformation is almost perfectly reversible without any major loss of data until the coefficients are manipulated at the frequency domain state.

In the tests presented in this report, Matlab was used to produce fast results with little to no preparation. Matlab contains most of the functions necessary to test Fourier transformations, as well as easy methods of displaying the results. Calling these functions allows for processing and compressing of image files and displaying the result at different stages in the process. The functions and the function names are from the *TNG032* booklet[1], the functions used are *fft2*, *ifft2* and *fftshift*.

3 Method

3.1 Transform coding with *fft2* and *fftshift*

The function *fft2* produced a matrix of Fourier coefficients corresponding to the values of the input matrix, which is the pixel values of the input image. Next, the function *fftshift* was used to shift the low-frequency coefficients from the edges to the center of the matrix. This is necessary when performing a low-pass filtering of the matrix given the structure of the filter.

3.2 Manhattan distance

The Manhattan distance function was used to calculate the distance between two integer-coordinates located on a grid. The Euclidean distance can be described as the straight line between two coordinates, however the Man-hattan distance function was used when the coordinates were placed on a grid similar to a map with city blocks. Under this circumstance, it is not possible to travel a straight line as *figure 1* shows.

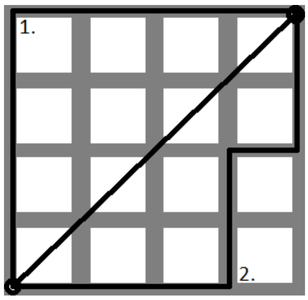

Figure 1: A visual representation of how the Manhattan distance function that has two-eligible paths with the same distance. Also, the non-eligible way is represented with the Euclidean way.

3.3 Low-pass filtering

By creating a filter that let signals pass through a given threshold, unwanted coefficients can be discarded. To set this threshold the matrix that represents the image was multiplied with the given filter-matrix. The next step was to reverse the process by shifting and inverse Fourier transforming the image to present the result.

The low-pass filtering method used in this lab is based on the Manhattan distance function. The values in the *16x16* filtering matrix were generated by calculating the Man-hattan distance from origin. When the threshold was set as the value *k* then all coefficients located at points in the mask-matrix with values greater than *k* were set to zero. Since the result of *fftshift* puts the lower coefficients closer to center, these were not discarded, even at low *k* values.

3.4 Implementation of transform coding

First the *fft2* function were used to produce Fourier coeffi-cients on separate *16x16* portions of the image, the *fftshift* function prepared each for masking. Then each section was filtered using the mask matrix with a threshold value of *k*. Finally, the separate sections were shifted back using *fftshift* and converted again using *fftshift*. This method is similar to how the jpeg compression algorithm works.

4 Results

As previously explained, the function *fftshift* is used to shift the low-frequency coefficients from the edges to center. Below is an example of an image before and after *fftshift*. *See figure 2.*

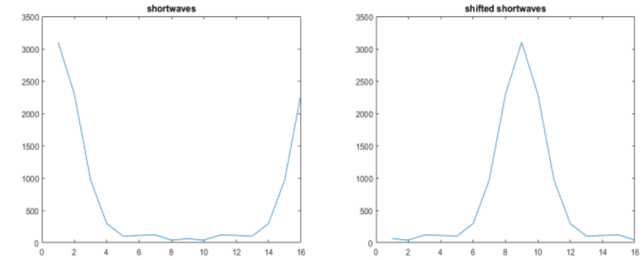

Figure 2: A representation of what happens when the low-frequency coefficients is shifted from the edges to origin.

The Manhattan distance mask image shows how the image was filtered. *See figure 3*. Every coefficient located outside the specified threshold value on the mask, was set to the value 0.

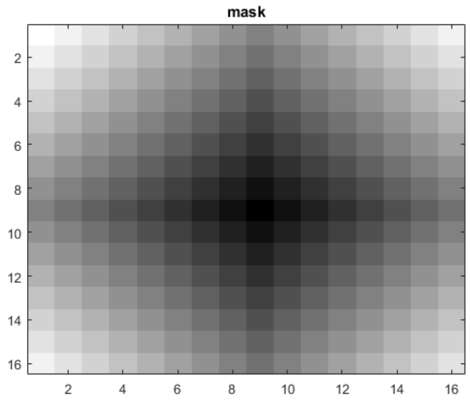

Figure 3: A mask that represent how the image will be filtered if this mask were applied to the image.

The result of an image compression shows a concrete example of how the image quality changes post-compression, *See figure 4.* The picture to the left is the original image while the one to the right is the compressed one with a threshold value of 1.

Figure 4: A visualization of how the image quality is deteriorated when the image size is compressed.

5 Discussion

Using the lower frequency coefficients of the Fourier transform it is possible to create a good approximation of the original pixels. Since the threshold value of the low-pass filter is easily modifiable for each section, it would be possible to have a variable threshold value. Parts of the image with more detail could have a higher threshold value or even be divided into smaller sections. This could potentially increase the perceived picture quality.

Figure 4 is an example of what effect a lower threshold value has on a picture. Overall the compressed picture is similar to the original but has a distinct lack of detail. Comparing areas with sharp transitioning from dark to light pixel values on both pictures, this is where much of the detailing is lost. Looking at areas with less contrast, it is harder to tell the difference between the original and the compressed version. This would be a good situation to use variable threshold values. The sections of the image with higher contrast should be compressed using a higher threshold value to save more of the details.

6 References

[1] R. Lenz, K. Nordström, O. Pedersen and O. Svensson (2015). *Laborationer i Tillämpad transformteori TNG032.* Norrköping: LiU-tryck.