Lab 02

2D Signal Processing

Aaron Dinesh - 20332661

25 February 2024

Declaration

I hereby declare that this report is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university. I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`. I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `https://libguides.tcd.ie/academic-integrity/ready-steady-write`. I consent to the examiner retaining a copy of the report beyond the examining period, should they so wish. I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. **Please consult with your supervisor on this last item before agreeing, and delete if you do not consent**

Signed: Aaron Dinesh Date: 25 February 2024

1 Separable Filters

1.1 Creating a Convolution Mask

In this section, we were given the following transfer function:

$$\mathcal{H}_0(z_1, z_2) = \frac{1}{20.25}(z_1^{-1}z_2^{-1} + 2.5z_1^{-1} + z_1^{-1}z_2 + 2.5z_2^{-1} + 6.25 + 2.5z_2 + z_1z_2^{-1} + 2.5z_1 + z_1z_2) \quad (1)$$

Since this is a Linear Shift Invariant system, this transfer function can be implemented as a convolution between the input signal and a convolution mask. A mask can be generated simply by looking at the transfer function. First, we establish a few conventions. $z_1^0 z_2^0$ will define the center of the convolution mask. Powers z_1 control the position of the coefficient in the first dimension relative to the center, with positive powers going down along the first dimension and negative powers going up along the first dimension. Similarly for z_2, but along the second dimension of the mask. Using these conventions we can generate the convolution mask that can be seen in eq 2

$$ConvMask = \frac{1}{20.25}\begin{bmatrix} 1.00 & 2.50 & 1.00 \\ 2.50 & 6.25 & 2.50 \\ 1.00 & 2.50 & 1.00 \end{bmatrix} \quad (2)$$

1.2 Applying the Convolution Mask

Applying the mask is simple. I called the MATLAB conv2 function passing in the image and the mask as inputs. It is important to note that the image should be normalized to be in the range [0-1] before passing it to conv2. The output of the convolution can be seen in Figure 1

Figure 1: Image before and after convolution. Mask is also shown.

1.3 Separating the Mask

Applying the convolution mask was fairly easy. However, this operation is computationally expensive. Given an $M \times N$ image with an $R \times C$ convolution mask, we would need to perform $M \times N \times R \times C$ operations which can get expensive when considering large images and large masks. However, if we could apply a 1D mask to the rows and then another 1D mask to the columns and achieve the same output as the 2D mask, we could cut down the number of operations to $M \times N \times (R + C)$. Separable filters allow us to do this. A filter is considered to be separable if it can be written as the outer product of two vectors. That is to say, $mask \equiv u \otimes v \equiv uv^\top$ for a row vector u and v. The mask in eq 2 can be decomposed into the following row and column vector.

$$u = \frac{1}{4.5}\begin{bmatrix} 1.0 \\ 2.5 \\ 1.0 \end{bmatrix} \qquad v^\top = \frac{1}{4.5}\begin{bmatrix} 1.0 & 2.5 & 1.0 \end{bmatrix} \quad (3)$$

We can perform the outer product of these two vectors to show that it gives the same result as the

original mask

$$\frac{1}{4.5}u \times \frac{1}{4.5}v = \frac{1}{4.5}\frac{1}{4.5}\begin{bmatrix}1.0\\2.5\\1.0\end{bmatrix}\begin{bmatrix}1.0 & 2.5 & 1.0\end{bmatrix} \tag{4}$$

$$\frac{1}{20.25}\begin{bmatrix}1.0\times1.0 & 1.0\times2.5 & 1.0\times1.0\\2.5\times1.0 & 2.5\times2.5 & 2.5\times1.0\\1.0\times1.0 & 1.0\times2.5 & 1.0\times1.0\end{bmatrix} \tag{5}$$

$$\frac{1}{20.25}\begin{bmatrix}1.00 & 2.50 & 1.00\\2.50 & 6.25 & 2.50\\1.00 & 2.50 & 1.00\end{bmatrix} \tag{6}$$

Which we can clearly see is equal to eq 2 meaning that we can separate our 2D convolution mask into 2 smaller 1D convolution masks.

1.4 Applying the Separated Masks

To apply the separated masks we just need to call conv2 twice, with the output of the first convolution step being fed into the second convolution step and changing the mask from u to v. Since we only care about the area where the image is defined, the padding option was set to 'same'. This can be seen in Figure 2.

Figure 2: Image shown using 2D conv mask and separated masks. Histogram of errors is also shown

1.5 Calculating the Mean Absolute Error

As we can see from the histogram, the absolute error is very low. The mean absolute error for the whole image was calculated to be 4.41691e−17.

2 Image Gradients

2.1 Calculating Gradients

The horizontal gradient of the image can be determined using the following formula:

$$H_x(z_1, z_2) = z_2 - z_2^{-1} \tag{7}$$

Using the same conventions defined before, we can arrive at the following convolution mask:

$$H_x = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{8}$$

The vertical gradient, similar to the horizontal one, has its own transfer function and therefore a convolution mask can also be derived. The steps of which are shown below.

$$H_y(z_1, z_2) = z_1 - z_1^{-1} \tag{9}$$

$$H_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

These can be convolved with the image using the conv2 function to produce a gradient map for each axes. A bias of 0.5 was added to each image so that derivative values that are close to 0 would appear gray, while highly negative and positive derivatives would appear black and white respectively. Once the horizontal and vertical derivatives are calculated, we can calculate and plot their magnitude, which had to be rescaled into the range of 0-1. The derivatives and the magnitude can be seen in Figure 3 below.

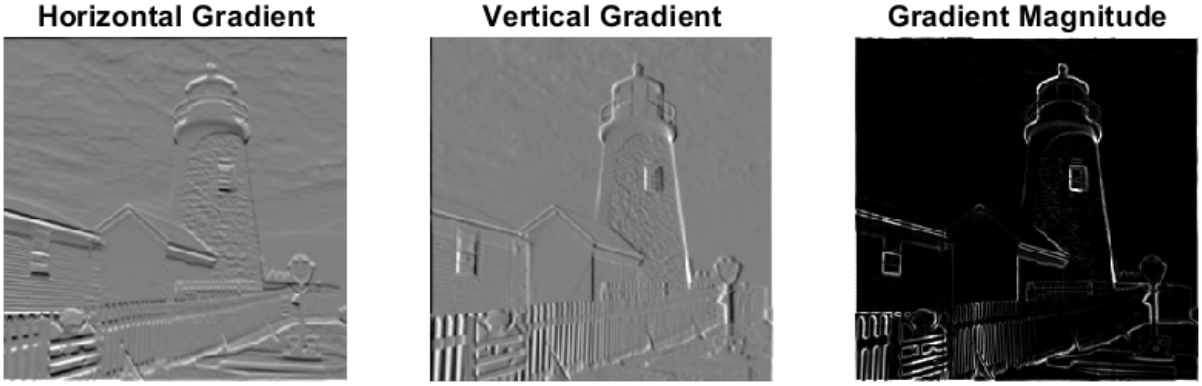

Figure 3: Horizontal and Vertical Gradients and the Magnitude

3 Downsampling

3.1 Simple and Complex Downsampling

An image can be easily downsampled by a factor of N by only considering every N^{th} pixel. In MATLAB this can be done by indexing the image array using a range, I(1:N:end, 1:N:end). However, this naïve implementation has a few problems. Since downsampling has the effect of compacting the images of the baseband signal in the frequency domain, a phenomenon known as aliasing occurs. This happens when the images of the signal overlap with the baseband signal. The moire patterns that can be seen in the middle image in Figure 4.

The proper way to downsample an image is to first apply a lowpass filter to the image. In the frequency domain, this removes the high frequency information thereby creating a larger separation between the baseband signal and its images. This then lowers the chance of aliasing when we downsample the image. The improvement over the naïve downsampling approach is evident when looking at the last image in Figure 4.

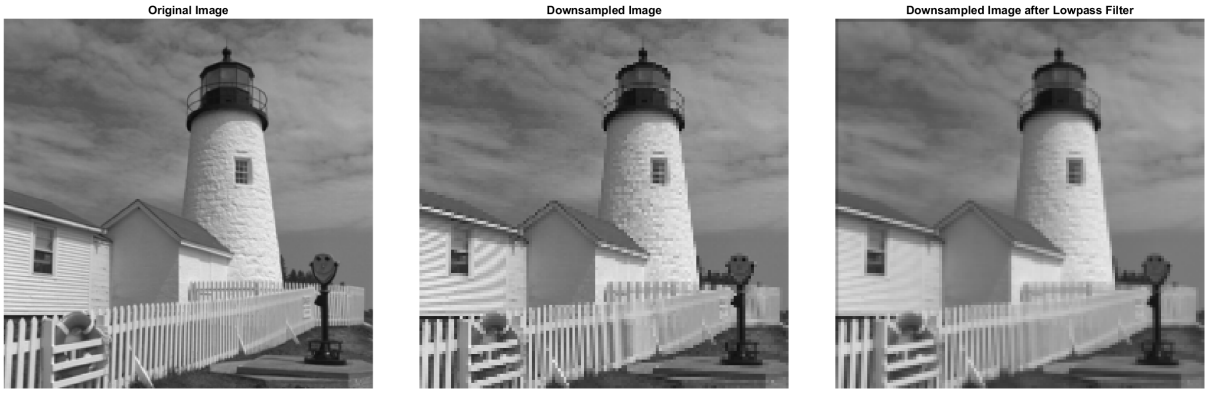

Figure 4: Original and Downsampled Images

4 Unsharpen Mask

4.1 Defining and applying the Unsharpen Effect

Applying this filter counterintuitively has the effect of sharpening the image. We first begin by passing the image I through a low pass filter to get I_l. This can then be subtracted from I to leave us with a new image that only contains the high frequency components in this image. We can then apply a gain of α to this image to control how much of the high frequency components we want to add back to the image. This then gives us our final sharpened image. If we raise α too much, we end up introducing sharpening artifacts to the image. However, if α is too small we don't end up with the desired sharpening effects. This is demonstrated in Figure 5.

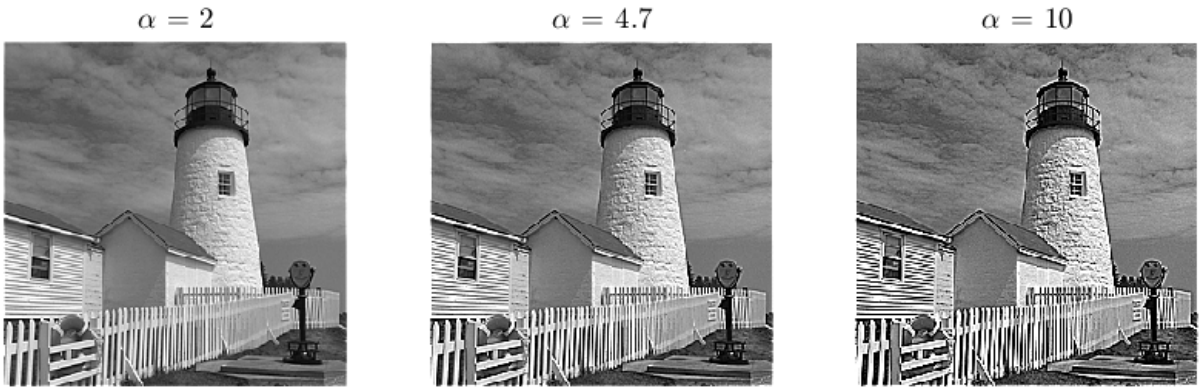

Figure 5: Sharpening effect with 3 values of α

Through trial and error I found that an α value of 4.7 provided good enough sharpening results without introducing too many artifacts. These artifacts are visible in the 3rd image as banding on the wall of the house and the fence.