SF2 Image Processing

First Interim Report

---

1. Simple Image Filtering

This section introduced fundamental image filtering techniques. It was noted that the filter output borders gradually faded to black due to the assumption that all pixels outside the region defined by the image are black. To alleviate this problem, the image was reflected symmetrically about horizontal and vertical axes.

2D filtering was achieved by filtering rows and columns independently in turn. It was noted that the order in which rows/columns were filtered affected the 2D-filter output. The maximum absolute pixel difference between the row-column and column-row filtered cases was found to be 1.1369e-13 for a half-cosine filter of size 15.

Next, it was decided to observe the changes in the low and high-pass images due to varying the filter size. The energy of the images was evaluated so as to objectively track the changes in the images. It was observed that increasing the filter size decreased the energy in the low-pass images and increased the energy in the high-pass images. Visually it was noted that the low-pass images became blurrier as the filter size was increased while the high-pass images included more content from the original image. Overall, it was noted that the low-pass images had 18 to 50 times the energy in the high-pass images. The upper end of the range is for the smaller filter sizes while the lower end of the range is for the larger filter sizes.

2. Laplacian Pyramid Scheme

The low-pass images were decimated with little loss of information due to their low bandwidth. This led to compression by use of a Laplacian pyramid scheme.

2.1. Quantisation and Coding Efficiency

Important Note: In this report compression rates are defined as (total bits for compressed scheme / total bits for reference scheme). Where applicable, the quantiser was adjusted so that the compressed scheme gave the same rms error as the reference scheme.

To compress image data, it must first be quantised. As such the sub-images produced by the pyramid scheme were quantised before their entropies were computed. When quantised to a step size of 17 the entropies of images X, X1 and Y0 were found to be 3.4910, 3.4021 and 3.3100 bpp (bits per pixel) respectively. This gave data compression rates of 70.79%, 62.31%, 60.41% and 60.03% for 1-, 2-, 3- and 4-stage pyramids respectively. It can be seen that there are significant improvements (8%) in transtioning from 1- to 2-stage pyramids though adding more stages after this yields somewhat marginal improvements.

However it was observed that it would not make much sense to compare compression rates across pyramid sizes if the quality of the resulting reconstructed image is not similar. The quality was measured using the rms (root mean square) error metric. The rms errors were found to be 5.4165, 6.0639, 6.9232 and 7.7572 for 1-, 2-, 3- and 4-stage pyramids respectively, when quantised with step size of 17. The direct quantisation yielded an rms error of 4.9340 i.e. a smaller error yielded than the pyramid scheme. This was expected as the image reconstruction in a pyramid scheme involves a sum of sub-images so the quantisation error of each sub-image accumulates to form a larger error in the reconstructed image. Therefore it is expected that the rms error increases with pyramid size. Visually, it was noted that direct quantisation yielded a grainy image while the pyramid scheme yielded a smoother appearance - noticed especially in the clouds. As the size of the pyramid increased so did the blurring. The single-stage pyramid subjectively yielded the best appearance.

To make comparison of compression rates more fair, a function was written to find a constant quantisation step size in the pyramid that yielded the same rms error as the result of direct quantisation at a different step size. A golden-section search was used to find the optimum value. Later this function was modified to allow for variation of quantisation step size across levels in the pyramid. This was done so that quantisers in each layer would achieve equal MSE (Mean Squared Error) in the reconstructed image. A code snippet for this function is given in appendix B.

Constant Step Size: To yield the same error as a direct quantisation at step size of 17, constant step sizes across all stages for 1-, 2-, 3- and 4- stage pyramids were found. These step sizes were found to be 15.5562, 13.4917, 11.8081 and 10.4716 respectively. These results imply that larger pyramids would need to be quantised more finely which was expected since there are more sub-images to be summed and therefore more opportunities for error accumulation. With these step sizes, compression rates of 0.7418, 0.7119, 0.7469 and 0.7912 were calculated for 1-, 2-, 3- and 4-stage pyramids. Notice that the 2-stage pyramid offers the best compression rate under this constant step size scheme. This meant that the larger pyramids require many bits to store them since they are quantised more finely.

Equal MSE: To yield the same error as a direct quantisation at step size of 17, varying step sizes were used across stages of various sizes of pyramids in order to contribute equal MSE to the reconstructed image. The values are given in table 1 below. Notice that the step size calculated for the single-stage pyramid is the same as the constant step size case, as expected. The compression rates achieved under this scheme were 0.7418, 0.6648, 0.6455 and 0.6397 for 1-, 2-, 3- and 4-stage pyramids. In this case, compression rates appear to improve as the size of the pyramid is increased.

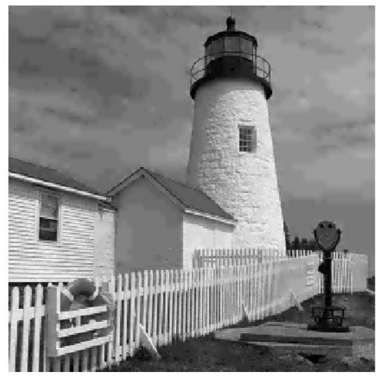
		Pyramid size			
		1	**2**	**3**	**4**
	1	15.5562	16.9852	17.8184	18.2145
Stage	**2**	-	11.3235	11.8789	12.1430
	3	-	-	6.4794	6.6234
	4	-	-	-	3.3887

Table 1: Quantisation step sizes required for equal MSE

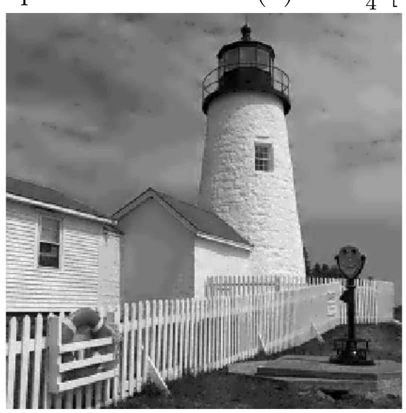
A larger filter as given in section 6.3 of the handout was used to obtain more blurred reconstructed images. In appendix A it can be seen that generally the pyramid scheme produces more blurred images than the direct quantisation scheme. It can also be seen that the clouds in figure 1(c) are more blurred than those in figure 1(b) due to the use of the second filter. The optimum step sizes and compression rates for this filter can be obtained just as above for the previous filter using the function get_optimum_step.m and h = $\frac{1}{16}$ [1 4 6 4 1].

A. Images

(a) h = $\frac{1}{4}$ [1 2 1], direct quantisation



(b) h = $\frac{1}{4}$ [1 2 1], 2-stage pyramid



(c) h = $\frac{1}{16}$ [1 4 6 4 1], 2-stage pyramid

Figure 1: Reconstructed images: Direct quantisation and 2-stage pyramid

B. Program Listings

```
function step = get_optimum_step(step_X, num_stages, X , h, tol, equal_mse)

    % low tol gives a more accurate search
    % equal_mse is a boolean indicating whether equal_mse criterion or
    % const. step size should be used
    % if equal_mse is used then the step size for layer 0 is found - others
    % may be obtained using this step size and get_equal_mse_ratios

    x1 = 1;
    x2 = 50;
```

```
    % objective function
    f = @(step) dist_to_rms_X(step, step_X, num_stages, X, h, equal_mse);

    step = golden_search(f, x1, x2, tol);

    return


function dist = dist_to_rms_X(step, step_X, num_stages, X, h, equal_mse)

    rms_X = get_rms(step_X, 0, X, h, 0);
    dist = abs(get_rms(step, num_stages, X, h, equal_mse) - rms_X);

    return


function x3 = golden_search(f, x1, x2, tol)

    % get f1 and f2 for initial points
    f1 = f(x1);
    f2 = f(x2);

    r = (sqrt(5) - 1)/2;
    x3 = r*x2 + (1-r)*x1; f3 = f(x3);

    % initial x1 and x2 must be picked such that f3 is less than
    % f1 and f2

    while abs(x1-x2) > tol
        x4 = r*x1 + (1-r)*x2; f4 = f(x4);
        if f4 < f3
            x2 = x3; f2 = f3;
            x3 = x4; f3 = f4;
        else % f4 > f3
            x1 = x2; f1 = f2;
            x2 = x4; f2 = f4;
        end
    end

    return
```