Report Assignment 2: Image Pyramid

Shantanu Chaudhary (2010CS502095)

Introduction:

The purpose of this report is to report the observations of the assignment 2 regarding implementation of image pyramid (Gaussian as well as Laplacian). Image pyramids are an interesting data structures for storing information about an image. The pyramids in turn can be used for image blending (stitching), image compression etc. This assignment consists of two parts which will be described in the following sections. We start with some preliminary information about the operations of image pyramids.

Image Pyramid – The Building Blocks:

There are two main operations which lie at the centre of the pyramid encoding scheme. These are called REDUCE and EXPAND. These operations do not perform image compression/expansion on their own. They form a part of the whole algorithm to construct the image pyramid.

- **Reduce:** Reduce takes an N X N image and reduces it to an N/2 X N/2 image. In other words, it reduces an image to 1/4 its original size in number of pixels. Reduce operates by taking a weighted average of pixel values over a block of pixels to produce a single pixel.
- **Expand:** Expand is just the opposite: it takes an N X N image and expands it to N*2 X N*2. Expand works by interpolating to create new pixels in the regions between pixels.

Reduce Operation – Implementation:

In this assignment, I have basically implemented the `impyramid` MATLAB function. By referring the manual pages of MATLAB, I have used the following kernel:

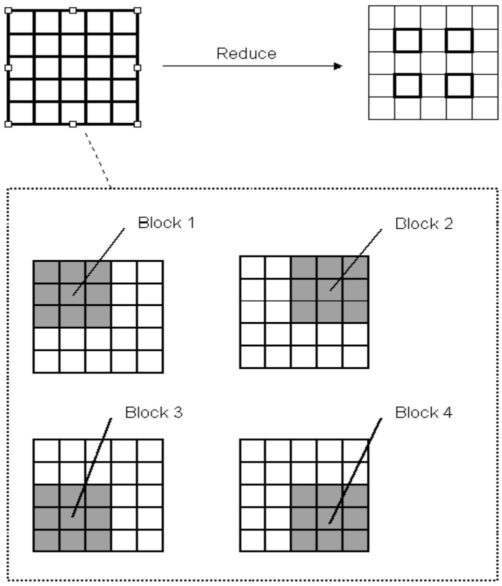
With a = 0.375

$$w = \left[\frac{1}{4} - \frac{a}{2}, \frac{1}{4}, a, \frac{1}{4}, \frac{1}{4} - \frac{a}{2}\right]$$

As mentioned in the paper, ‘a’ should be small and separable. The above value of a gives the filter a Gaussian shape. This kernel is then used to generate a filter with 5x5 dimensions which in turn is applied over a 5x5 block of image. The size discrepancies have been handled by additional padding (by replication, or zero/one padding). Figure 1 shows diagrammatically the reduce operation.

Expand Operation – Implementation:

Like the previous operation, this operation has also been implemented on the lines of `impyramid` matlab function. Above Figure 2 represents four pixels that are expanded to a 5 X 5 block. Pixels a, b, c, and d will retain their values after the expansion. Pixel x will be interpolated by averaging together pixels a and b. Pixel y will be interpolated by averaging together pixels a, b, c, and d.



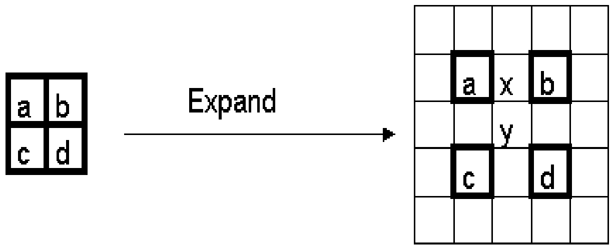


Figure 1, Figure 2

Fundamentals – Part1

The foremost step is the construction of image pyramids using the above operations. Next, the implementation can employ various ways to deal with colour images (for example compute in RGB space, or use YUV space and do stitching in Y component). The principal behind image stitching is that an image consists of many high frequency and low frequency components which can be stitched together without distorting the image to form a new composite image.

PYRAMID CONSTRUCTION:

```
L = I - Expand (Reduce (I))
```

IMAGE RECOVERY:

```
I = L + Expand (R)
```

Where I: Original Image, L: Image Pyramid, R: Reduce (I) operation (as referred in the paper)

Algorithm – Part1

The implementation in this assignment works for both colour as well as grey scale images. In case of grey scale images, there is 1 image pyramid for an image. In case of a coloured image, a composite image pyramid has been constructed for each of the three primary colours (RGB). Now, using the image pyramid, the image is stitched (includes use of Gaussian filters for smoothening) for each component to obtain the composite image.

Observations – Part1

In this part, we had to perform image blending using image and another without image pyramid (use low pass filter to smoothen the edge). The code was run on the sample images and following results were obtained:

Figure 1.1

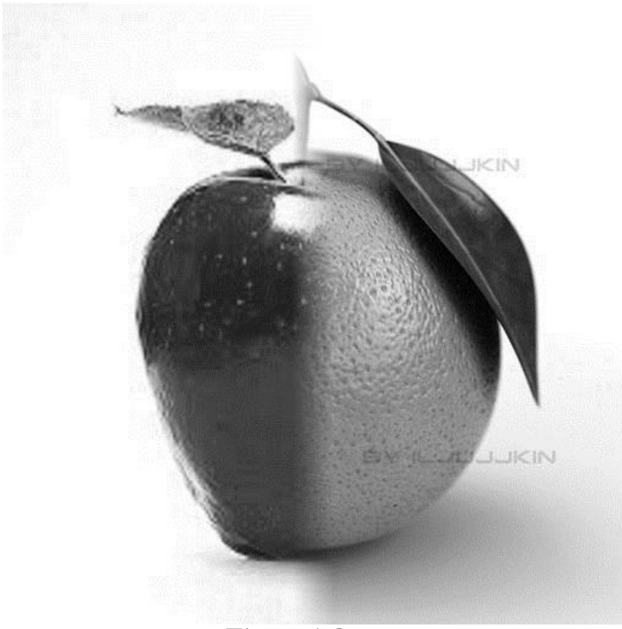


Figure1.2

Here, figure 1.1 shows results of blending with image pyramid while figure 1.2 shows image stitching with simple averaging filters. It is quite clear from the figure that there is a gradual change in intensity of colours in case of figure 1.1 than figure 1.2. Another evidence is the line (seam) that can be viewed in case of figure 1.2. Hence, blending through image pyramid tends to be finer.

Another example:

Figure 1.3

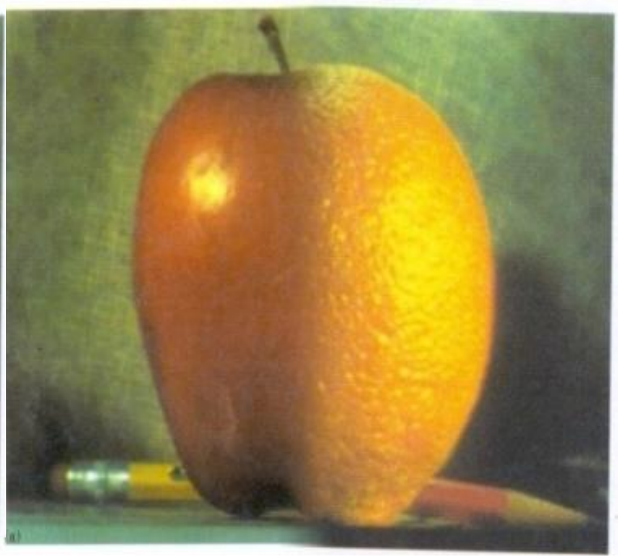


Figure 1.4

Figure 1.3 represents image blending with image pyramid.

Figure 1.4 represents image blending with smoothening masks.

Fundamentals – Part2:

To elaborate on the procedure to construct the Laplacian pyramid, we use the following figure.

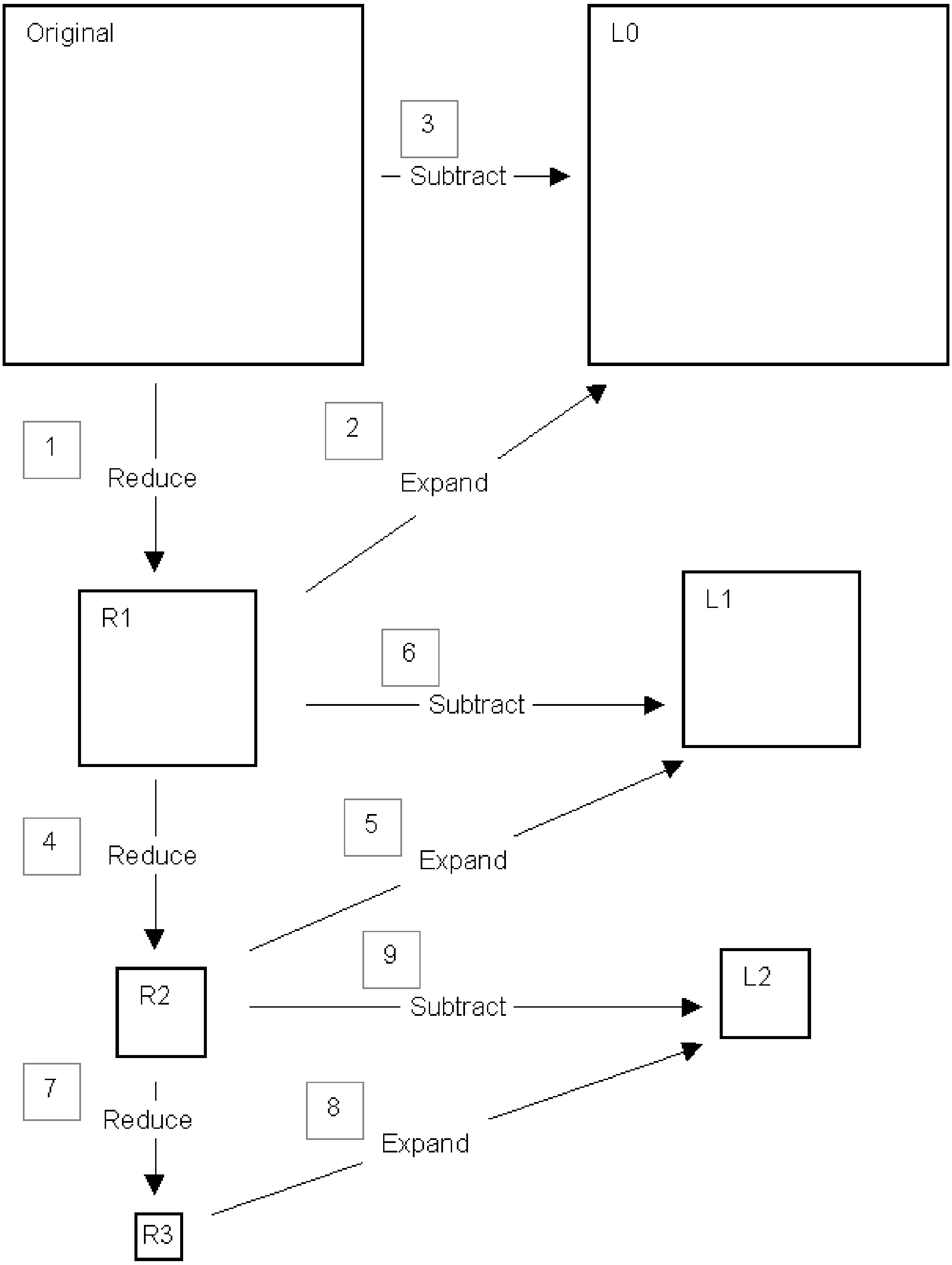


Figure 2.1

Above figure gives an elaborate description of how the pyramid is constructed. The original is reduced to R1 and then expanded again. The expanded image is then subtracted from the original to produce L0. Now the original can be reproduced using only L0 and R1. We can also encode R1. R1 gets reduced to produce R2. R2 is then expanded and subtracted from R1 to produce L1. R1 can now be reproduced using only R2 and L1. If R2 is subsequently encoded, the only images needed to restore it will be R3 and L2. Finally, the only images we will need to restore the original image are L0, L1, L2 and R3. This collection of images constitutes the Laplacian pyramid. L0 is the base or bottom level of the pyramid, and R3 is the top of the pyramid. Notice that R3 is very small; it's 1/64 the size of the original. Also, because L0, L1, and L2 are difference images they contain a high number of zeroes and have very peaked distributions. These characteristics enable the Laplacian images to be compressed very effectively. Also notice that the base level of the pyramid, because it is the largest, represents finer details in the image. It is therefore the image containing the highest frequencies. As the images get smaller, each of their pixels represent a greater area in the original image. Thus, higher levels in the pyramid represent lower frequencies in the image.

The process of compression can be complemented by the use of entropy encoding (Huffman and Arithmetic Encoding) and quantization. The use of quantization relies on choosing an appropriate value of threshold to truncate intensity values in the pyramid. Various heuristics can be used for this operation. Quantization done in a suitable way can compress image size without compromising on the information conveyed by the image.

Maximum information of low frequency component of image can be stored in a low resolution form as a smaller size image.

Also, the high frequency component which has been stored for each pyramid level comprises mostly of low levels (due to homogeneity in image and less variation in the Gaussian filtered image from the original one) and hence, it can be quantized up to 5 bits and information can still be maintained efficiently

Algorithm – Part2:

Same as in Part 1. But here we did not do quantization or entropy calculation. In the problem statement, we were asked to show how an image gets stored in the pyramid at various resolutions. So, once, the image pyramid was ready, we did an EXPAND operation on the successive layers starting from top to bottom. And wrote the image information in the form of an image file.

To sum up:

- Take the image pyramid by Gaussian filtering,.
- Take the difference of Gaussian pyramid from original: this represents the high frequency component (Laplacian)
- Do this up to n levels.
- Quantize the Laplacian up to low loss levels.
- Back track for restoring the image.

Observations – Part2:

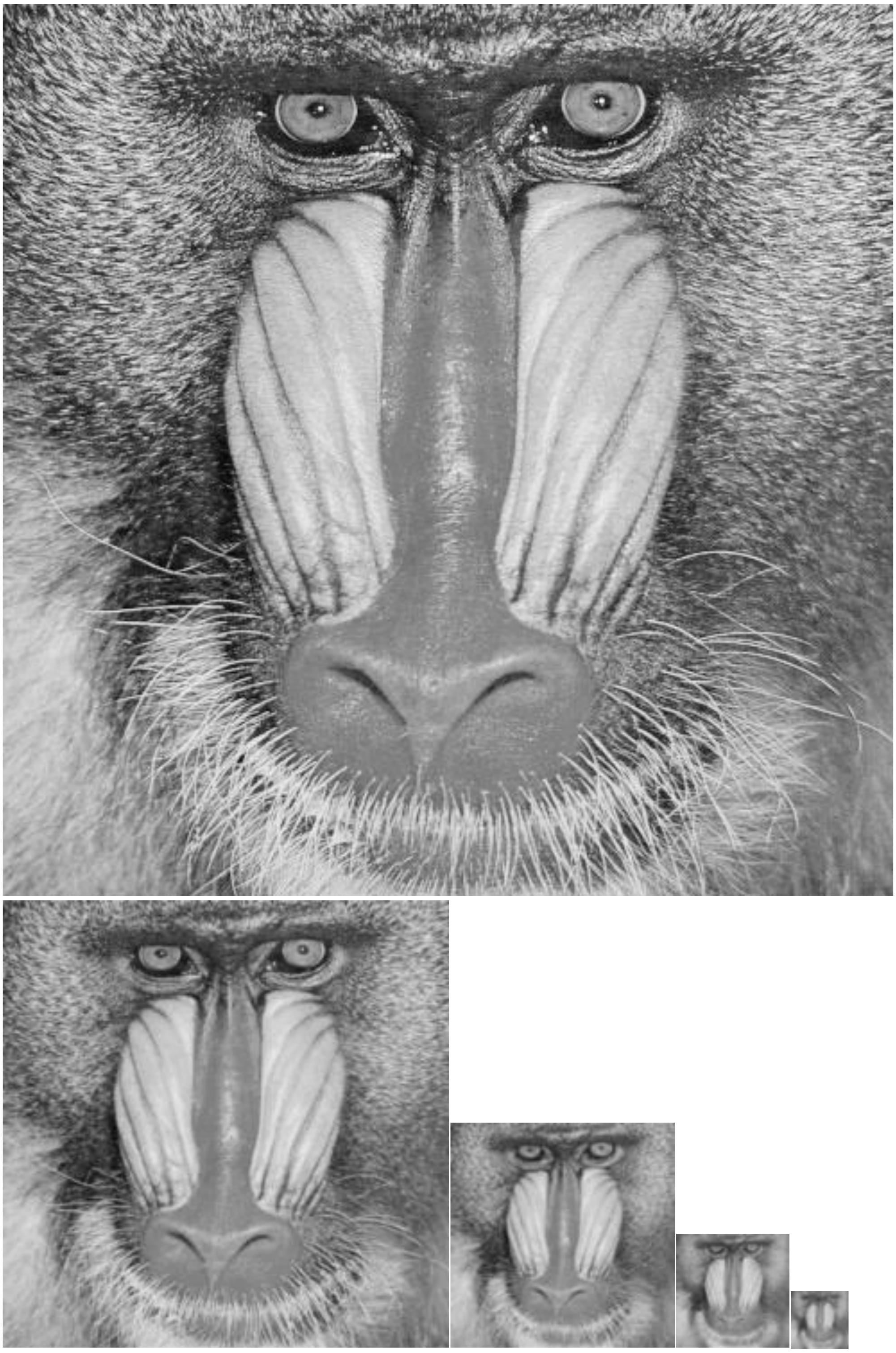
Following observations have been made. The image samples shown below have been derived from the successive expanded image pyramid levels.

Test Case 1:

Top: Input Image

Bottom: Successive levels of Image pyramid

Test Case 2:



Top: Input Image

Bottom: Successive levels of Image pyramid

Test Case 3:

Top: Input Image

Bottom: Successive levels of Image pyramid