Programming Assignment-1: Camera Pipeline

Sanjana Prabhu EE17B072

February 28, 2021

1 Introduction

In this assignment, we have been given three raw images and we have to process these following the pipeline in a traditional camera. The steps include demosaicing, white balancing, tone mapping and finally, denoising.

2 Part 1: Demosaicing

Demosaicing in a processing technique by which the missing RGB channel values of an image are computed by interpolation. In this section, demosaicing is done on the first image using bilinear and bicubic interpolation, as well as using an in-built MATLAB function called "demosaic". The results are compared in the following figures(Fig 1, 2, 3).

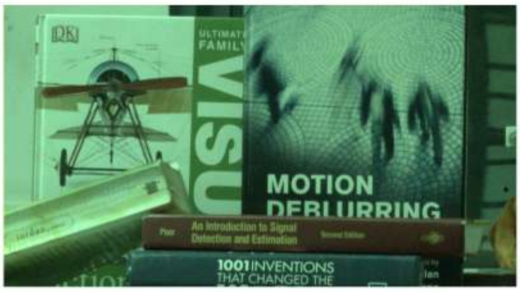

Figure 1: Bilinear Interpolation

Comparison of the methods of interpolation
Bilinear and bicubic interpolation was generating almost the same type of image, whereas the inbuilt MATLAB function generated a slightly sharper image.
What assumptions are you making while interpolating the missing pixel values and what will happen when the assumptions do not hold?
While using interpolation to compute the missing pixels, we assume that

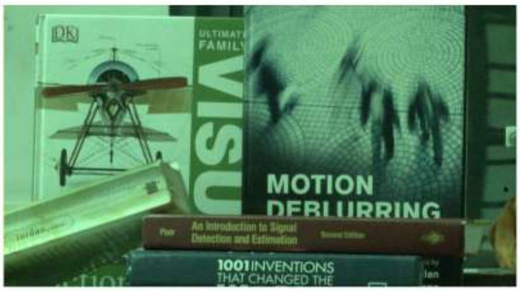

Figure 2: Bicubic Interpolation

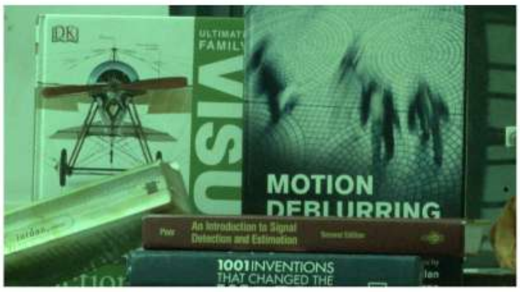

Figure 3: Using demosaic function

there are no edges in the image, that is, there is not much high frequency content in the image, as these lead to artifacts in the interpolated image due to aliasing of the high frequency content(edges). This can be seen in the next section.
Comparison of the demosaiced and actual version of the image from the Kodak dataset
The following figure(Fig 4, 5, 6, 7) compares the demosaiced and actual version of the image from the Kodak dataset. As we can see, there are some artifacts(color fringes or Moire patterns) present near the fence and the left-most building in the demosaiced image. These artifacts arise due to the fact that we are performing interpolation at the edges and that might generate some unexpected pixel values at these locations. Edges give rise to higher frequency content and these artifacts occur due to aliasing.

3 Part 2: White Balancing and Tone Mapping

White balancing is done to negate the effects of the colour of the light source and tone mapping is done to obtain better contrast, i.e., to ensure that all intensity values are represented uniformly. For white mapping, I chose the

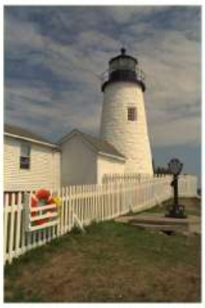

Figure 4: Demosaiced image - Bilinear Interpolation

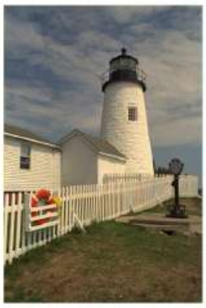

Figure 5: Demosaiced image - Bicubic Interpolation

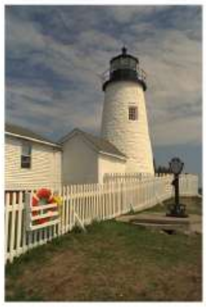

Figure 6: Demosaiced image - Using demosaic function

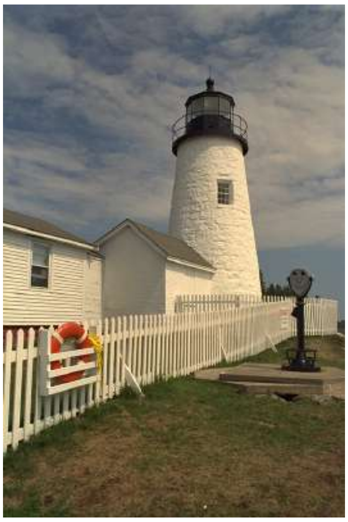

Figure 7: Actual image

outputs of bicubic interpolation.
There are different ways in which one can do white balancing, such as :

- Assuming that the average color of the image is grey. The results obtained using this method are shown in the following figure(Fig 8, 9, 10).

- Using specular highlight: In this technique, the brightest pixel is assumed to be white and the image is scaled accordingly. The results are shown in Fig 11, 12 and 13.

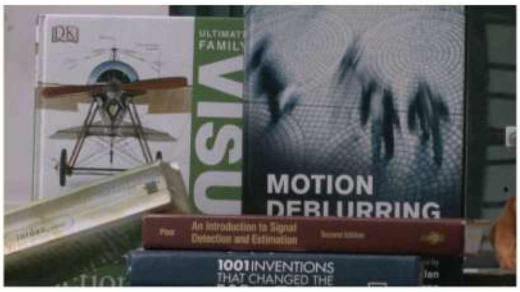

Figure 8: White balancing on image 1

Figure 9: White balancing on image 2

Figure 10: White balancing on image 3

- Assuming that some part of the scene is neutral: In this technique, one of the pixels is assumed to be neutral and the image is scaled accordingly. The results are shown in Fig 14, 15 and 16.

Two techniques of tone mapping have been shown. For tone mapping, I selected the outputs of the second white mapping technique.

- Histogram equalisation: In this technique, the RGB image is first converted to HSV and histogram equalisation is applied only to the V

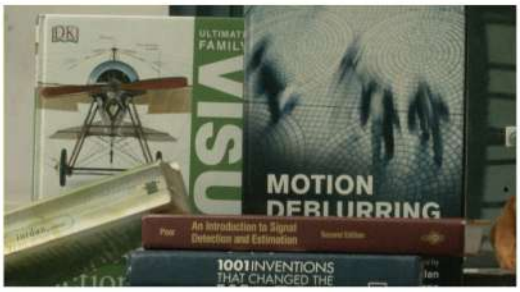

Figure 11: White balancing on image 1

Figure 12: White balancing on image 2

Figure 13: White balancing on image 3

channel(intensity channel). The results are shown in Fig 17, 18 and 19.

- Gamma correction: In this technique, the RGB image is first converted to HSV and the gamma equalisation is applied on the V channel. The results are shown in Figures 20 to 28, for three different values of gamma.

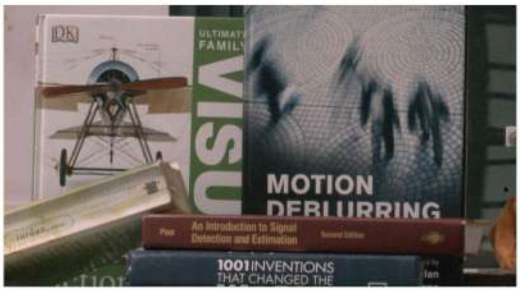

Figure 14: White balancing on image 1

Figure 15: White balancing on image 2

Figure 16: White balancing on image 3

4 Part 3: Image denoising

To perform denoising, we use bilateral filtering, i.e., performing an averaging operation on the pixels. The extent of averaging depends on both the noise variance and the difference in the intensity levels. For image denoising, I chose the outputs of histogram equalisation based tone mapping.
To estimate the noise variance for Image 1 and Image 3, I chose a relatively constant region of 50×50 and computed the standard deviation of the pixel intensities of these regions. These values will roughly correspond to the

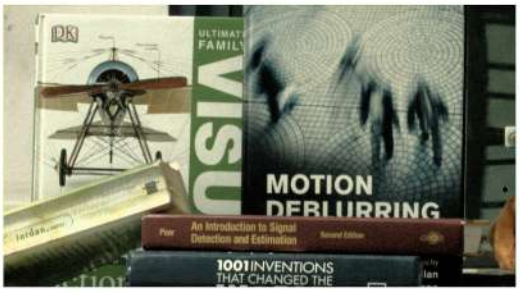

Figure 17: Tone mapping on image 1

Figure 18: Tone mapping on image 2

Figure 19: Tone mapping on image 3

standard deviation of the noise in the image. The results of denoising are shown in the figure below(Fig 29, 30, 31).

For a given Gaussian kernel with variance σ, at what point will you choose to truncate the function and why?

I would choose to to truncate at 3σ on both sides of the mean, i.e., 6σ totally because most of the energy of the Gaussian kernel is contained in this interval.

For a given σ_n how would you choose the value for σ_r? What would happen

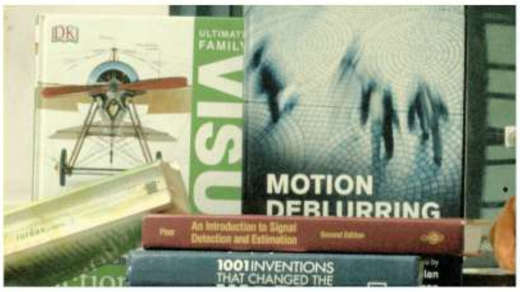

Figure 20: Tone mapping on image 1, gamma = 0.5

Figure 21: Tone mapping on image 2, gamma = 0.5

Figure 22: Tone mapping on image 3, gamma = 0.5

if $\sigma_r \leq \sigma_n$ or $\sigma_r >> \sigma_n$?

σ_r is a soft thresholding of what we consider a genuine edge value except the difference due to noise. Let us consider the two cases.

- If $\sigma_r \leq \sigma_n$, even a slight difference in intensity values might be considered as an edge and the averaging might not be carried out as we want. This slight difference can occur even due to noise. Especially in flat regions, small difference in intensity values will occur due to noise and it is important that we give the neighborhood pixels a good

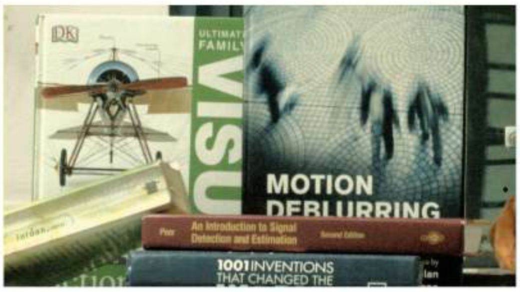

Figure 23: Tone mapping on image 1, gamma = 0.7

Figure 24: Tone mapping on image 2, gamma = 0.7

Figure 25: Tone mapping on image 3, gamma = 0.7

enough weight for averaging to take place. Choosing $\sigma_r \leq \sigma_n$ we can expect to very less smoothing effect.

- If $\sigma_r >> \sigma_n$, there will be considerable weights given even to pixels with very different intensity values from the current pixel. This will lead to a lot of averaging resulting in an image with its edges blurred.

A good choice hence is $\sigma_r \approx 2\sigma_n$.

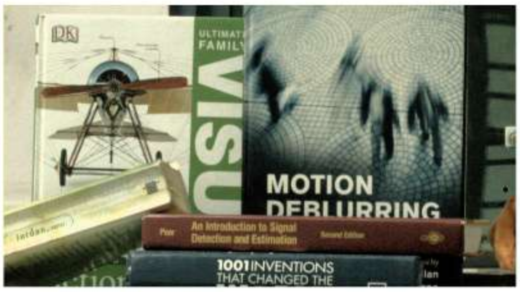

Figure 26: Tone mapping on image 1, gamma = 0.9

Figure 27: Tone mapping on image 2, gamma = 0.9

Figure 28: Tone mapping on image 3, gamma = 0.9

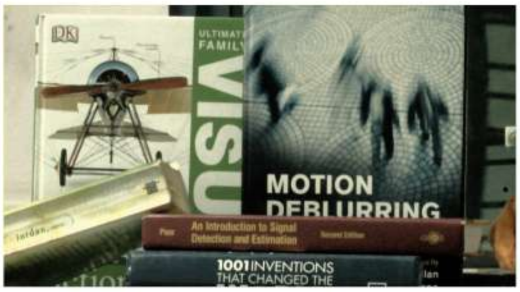

Figure 29: Denoising on image 1

Figure 30: Denoising on image 2

Figure 31: Denoising on image 3