CS24110 Assignment: Automatic Enhancement of Digital Images via Histogram Equalisation on LAB Colour Space, Contrast Enhancement on RGB and Average Value Blur Filter

Dimitar Tasev
Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK

This paper presents a novel approach to image enhancement based on the alignment of an image's intensity profile to a specified ideal value. This mid-mean alignment algorithm is easy to understand and fast to compute. The algorithm is tested on five test images and the results show that, in some circumstances, the algorithm is able to improve the quality of an image. However, the algorithm does contain some properties which means that the results may not always be satisfactory. These aspects are discussed in this paper alongside possible points for improvement.

I. Introduction

THE increased use of computers, along with the rise in social networks and the ubiquitous nature of digitals cameras means that society is experiencing a flood of digital images. Twenty years ago, digital cameras and digital image processing was still the domain of the professional; however, now nearly everyone has a camera in their phone and most programs that deal with digital images have some kind of image processing aspect.

The ability to alter, or manipulate, an image is not something that is new in the digital age. Developers of traditional film camera photos were able to modify their images in the dark room, a technical and often laborious task. In contrast, editing photographs in the digital age could not be simpler. Dedicated programs such as Adobe™ Photoshop® or Gimp[1] allow anyone to quickly and easily modify their images. These programs contain sophisticated algorithms to change the tone, colour, brightness, contrast, and many other aspects of the image. They also allow for multiple images to be "merged" and so the rise of the term "Photoshopping an image", meaning to alter the image's content in some way.

Although the algorithms in these software suites are powerful, more often than not they require some form of user input to guide the image modification process. Some products contain auto enhancement methods, but the ability to automatically enhance an image without any user input is still an ongoing area of research.

In recent times, this area of auto-enhancement has taken a different direction due to the rise of the social network Instagram™. Whereas, "Photoshopping" became synonymous with editing a photogram, Instagram has become synonymous with automatically enhanced images through the use of an Instagram filter. These filters will often produce extreme effects so that the image appears like an old Polaroid camera, or is changed to a high contrast black and white image. Whatever method is used the process is the same: alter the image in a specified way to produce a new, enhanced, image.

This paper is concerned with a new method to enhance a given image without any use input. The basic idea behind the approach is to match the intensity of the input image so that it is aligned to the middle intensity value. In theory, this should correct any brightness defects in the image.

The remainder of the paper is organised as follows: in Section II the methodology of the approach is given at both a high and low level. The results are given in Section III and then discussions and conclusions are drawn in Section IV.

[1]http://www.gimp.org

II. Methodology

In this section the methods behind the Average Value Blur Filter, Colour Space Conversion, Histogram Equalisation, and Contrast Adjustment is described. A high-level overview of how the method works, a mathematical outline and pseudocode are also provided.

A. High-level Overview

The idea behind the proposed automatic enhancement approach is to do a contrast adjustment to remap the intensity value so that the whole range of possible values is used, convert to LAB(CIELAB) colour space and perform a Histogram Equalisation on the Luminosity channel to enhance the picture, then convert back to RGB, and apply an Average Value Blur Filter for noise removal. The enhancement will follow the flowchart on figure 1.

Algorithm 1 Normal Histogram

function NormalHistogram(I) $\triangleright$ where $|I| = m \times n$,
$K = 256$
    $H = [0, K-1]$

    for each p **in** I **do**
        $H(p) = H(p) + 1$
    end for

    return H
end function

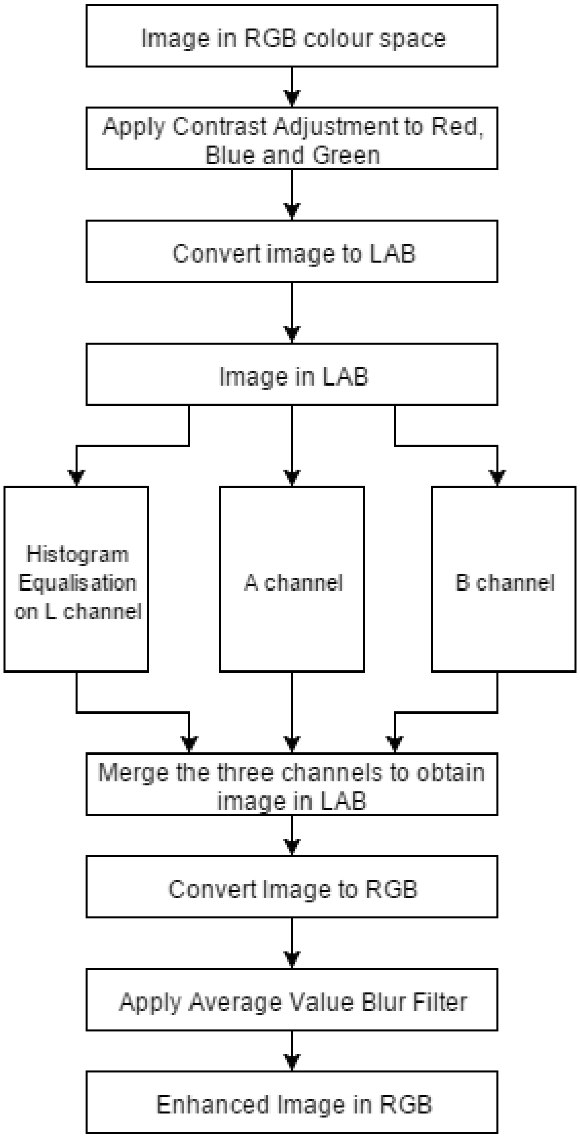


Fig. 1. A flowchart of the proposed enhancement method. It shows the steps that will be performed in the automatic enhancement.

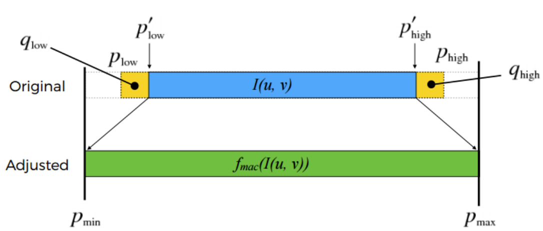
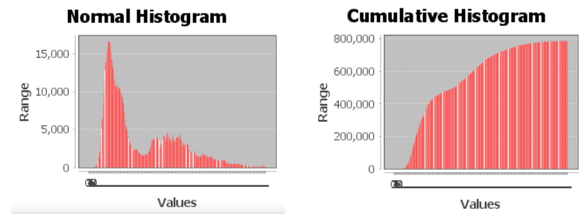


Fig. 2. An image showing how the range of values will be set when applying contrast adjustment to the picture.

B. Detailed Description

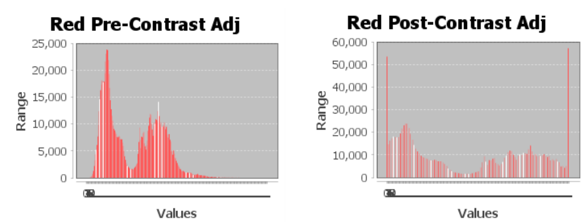
The proposed enhancement algorithm is a combination of filter and point-based histogram operations. To simplify the description of the operations, the algorithm assumes that the picture is grayscale, however the final algorithm works on RGB images, by repeating the functions for each colour channel. The first two operations both rely on histograms to perform their operations. A histogram is a graphical representation of a distribution of numerical data[1][2]. Figure 3 shows both a normal histogram and a cumulative histogram, both of which are used. The pseudocode for creating a **normal histogram** 1, and the pseudocode for creating a **cumulative histogram** from the normal histogram 2 are included in the paper. The equation for deriving a cumulative histogram from a normal histogram can be simply written as:



(a) original (b) modified

Fig. 3. Two images showing a normal histogram and a cumulative histogram of it.

$$H(i) = \sum_{j=0}^{i} h(j) \quad for(0 \leq i < K) \tag{1}$$



(a) original (b) modified

Fig. 4. Two images showing a normal histogram before and after being contrast adjusted.

The second operation, the Contrast Adjustment, is performed on the image in RGB colour space, so no conversion is needed. In images often the full range of values isn't used. Automatic contrast adjustment remaps the intensity values so that they occupy the full range of possible values[3]. We have to identify two quantiles at the low p'_{low} and high p'_{high} end of

Algorithm 2 Cumulative Histogram

```
function CUMULATIVEHISTOGRAM(I)        ▷ where K = 256
    H = NormalHistogram()
    CH = [0, K − 1]

    for each v in H do
        CH[v] = CH[v − 1] + H[v]
    end for

    return CH
end function
```

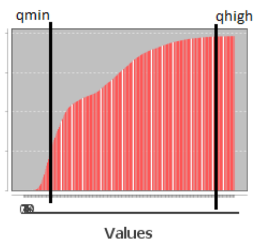
the intensity spectrum, and map the pixel values inside them to the extreme values, the other pixels are then linearly mapped to the interval $[p_{min}, p_{max}]$ 2. To calculate the two quantiles we set a range for ignored pixels q. Using that range we can calculate the quantiles using the cumulative histogram:

$$\begin{aligned} p'_{low} &= \min\{i|H(i) \geq m \cdot n \cdot q_{low}\} \\ p'_{high} &= \max\{i\ |H(h) \leq m \cdot n \cdot q_{high}\} \end{aligned} \tag{2}$$

Given that q follows:

$$0 \leq q_{low}, q_{high} \leq 1, , q_{low} + q_{high} \leq 1 \tag{3}$$

After applying the formula we get the following ranges:



Then we loop through the pixels in the picture and apply the following equation to each pixel:

$$f_{\mathrm{mac}}(p) = \begin{cases} p_{\mathrm{min}} & \text{for} \quad p \leq p'_{\mathrm{low}} \\ p_{\mathrm{min}} + (p - p'_{\mathrm{low}}) \cdot \frac{p_{\mathrm{max}} - p_{\mathrm{min}}}{p'_{\mathrm{high}} - p'_{\mathrm{low}}} & \text{for} \quad p'_{\mathrm{low}} < p < p'_{\mathrm{high}} \\ p_{\mathrm{max}} & \text{for} \quad p \geq p'_{\mathrm{high}} \end{cases}$$

We can see the difference in the histograms after adjusting the contrast 4, the value range now occupies the whole spectrum of possible intensities.

The last operation of the algorithm is Histogram Equalisation on the Luminosity channel in LAB(CIELAB) colour space. In order to do that we first have to convert to LAB. This cannot be done directly, so the RGB picture has to be converted to XYZ colour[4] space first, and then converted to LAB[5]. Then a normal and cumulative histograms have to be created on the Luminosity (L) channel, using the methods described earlier 1, 2. From the cumulative histogram we can equalise the pixel distribution, so that the histogram of the image approximates a uniform distribution. The goal of histogram equalisation is to shift the pixels in the image, so that the resulting cumulative histogram is approximately linear. [3] Figure 5 is showing the cumulative histogram of the L channel before equalisation, and after equalisation. Histogram equalisation on the cumulative histogram is done using the algorithm 3, implementing the following equation:

$$f_{he}(p) = \lfloor H(p) \cdot \frac{K-1}{mn} \rfloor \tag{4}$$

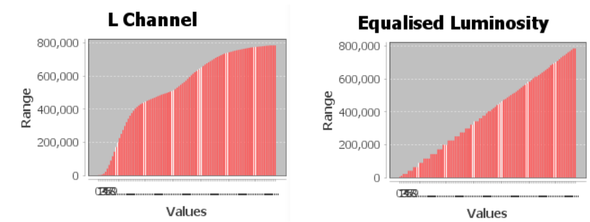
The last operation in the algorithm, the Average Value Blur is a filter, because it does not rely solely on a single pixel's value. The filter loops through the picture and acts on each pixel, but in the calculation the values of the neighbouring pixels are also included. The effect of is to reduce the noise

Algorithm 3 Histogram Equalisation

```
function HISTOGRAMEQUALISATION(I)                ▷ where K = 256
    CH = CumulativeHistogram(I)
    EQH = [0, K − 1]

    for each v in CH do
        EQH(v) = CH(v) · (K−1)/(m×n)
    end for

    return EQH
end function
```



(a) original (b) modified

Fig. 5. Two images showing a normal histogram and the equalised histogram of it.

in the picture, by applying a blur effect.[6] The described filter operation acts on each channel in the same manner, using a filter operation function f. That is:

$$I'(u,v) = f(\frac{1}{9} \cdot \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)) \tag{5}$$

C. Implementation

The detailed description section focuses on a single colour channel; however, since the algorithm needs to work on RGB images, the above methods needs to be repeated for each of the colour channels. The three enhancement operations are used in sequence in the image enhancement algorithm. Each of them will be looked at separately, in the order that they are applied in the algorithm, because if the order is changed, the final result will also be different. At the end of each of the operations, a clamping operation will be used to ensure that the modified intensities do not fall out of the displayable range.

The first operation The description in the previous section is focussed on a single colour channel; however, since the algorithm needs to work on RGB images, the above method will need to be repeated for each colour channel. To achieve this, the implementation will need to keep track of the actual mean values for the three colour channels separately and then ensure that the correct shift is applied to each colour channel. As well as this, a clamping operation will need to be employed so that the modified intensities do not fall out of the displayable range of intensities.

The pseudocode algorithm is given in Algorithm **??**. The function takes as input an RGB $m \times n$ image I and then starts by producing a blank output image of the same dimensions

Image	s_{red}	s_{green}	s_{blue}	δs
Image 1	$+53$	$+49$	$+46$	7
Image 2	$+63$	$+58$	$+55$	8
Image 3	$+49$	$+25$	-21	70
Image 4	-14	-24	-31	17
Image 5	$+12$	-1	$+4$	13

TABLE I
THE DIFFERENT SHIFTS FOR EACH COLOUR CHANNEL FOR EACH IMAGE. THE LARGER THE SHIFT THE GREATER THE EFFECT; THE LARGER THE DIFFERENCE IN SHIFTS BETWEEN EACH COLOUR CHANNEL THE GREATER THE COLOUR CHANGES WILL BE.

(line 2). The ideal means are then given and lines 4-10 loop through the input image summing each pixel channel and then dividing by the total number of pixels to produce the actual mean (note, this divide is an element-wise divide so that each element in the actual vector is divided by $m \times n$). Once the actual mean is calculated, the shift is found for each channel by subtracting the actual mean from the ideal mean (line 11). Finally, the output image is filled by adding the shift amount to each pixel value found in the input image making sure that the pixels are clamped into the range $[0, 255]$. The function returns the modified image (line 17).

The implementation described in Algorithm 1 requires two loops through the image but using an efficient pixel accessing mechanism (i.e. Java's `WritableRaster` [7]) then the run-time for this operation should be minimal.

III. RESULTS

The proposed method is tested on the provided four images. The results of performing the mid-mean alignment algorithm on each of these images are shown in Figure 6.

In all of the images the algorithm causes a change in the modified image, with the least significant change being seen in Image 5 (Figure 6 (i, j)). Although the change is least significant for Image 5, it is only Image 5 and Image 4 that have natural looking results; all the other images, when modified, appear either washed out or with large colour changes.

The modified versions of Images 1 and 2 both appear washed out. This can be explained when the histograms are examined. Figure 7 shows the histogram before and after processing for Image 1. The washed out effect occurs because all of the pixels within the shaded orange region in the original image are mapped to maximum intensity in the final image. As well as this, the full range of intensities are not used in the final image so that image will appear to have low contrast. This identifies an important property of the proposed algorithm, although it only performs a brightness adjustment, the algorithm can also modify contrast. This is due to the fact that large shifts in intensity (such as those seen in Figure 7) will cause a large block of pixels to be mapped to the extreme values. The remaining pixels will be shifted as normal but the range of intensities used will be decreased thus leading to a decrease in contrast.

The colour change that is seen in Image 3 (Figure 6 (e, f)) occurs because the shifting operation has a larger effect on a single colour channel. The proposed method shifts each colour channel by a separate amount, so if, say, the green channel has a larger shift than the red and blue channels then the modified image will display a marked change in the green component (either more green or less green). This effect can be further understood by examining the shift amounts for each colour channel for each image. Table I shows the shift amounts for each colour channel for each image as well as the size of the shift (measured as the range of the shift values for that image). As can be seen, the cases when there is a large δs correspond to cases when there is a significant change in colour. Similarly, cases where there is a large shift even though there is a small δs, correspond to cases where there are contrast changes.

IV. DISCUSSION & CONCLUSIONS

The results show that the proposed method works best when there is a small shift observed for each colour channel and the difference in shifts between each colour channel is minimal. If the shift amount is large, then it is likely that the image will appear washed out and the image's contrast will change. Also, if the shift amount is very different for each colour channel, then the modified image will contain a very different colour distribution than the original image.

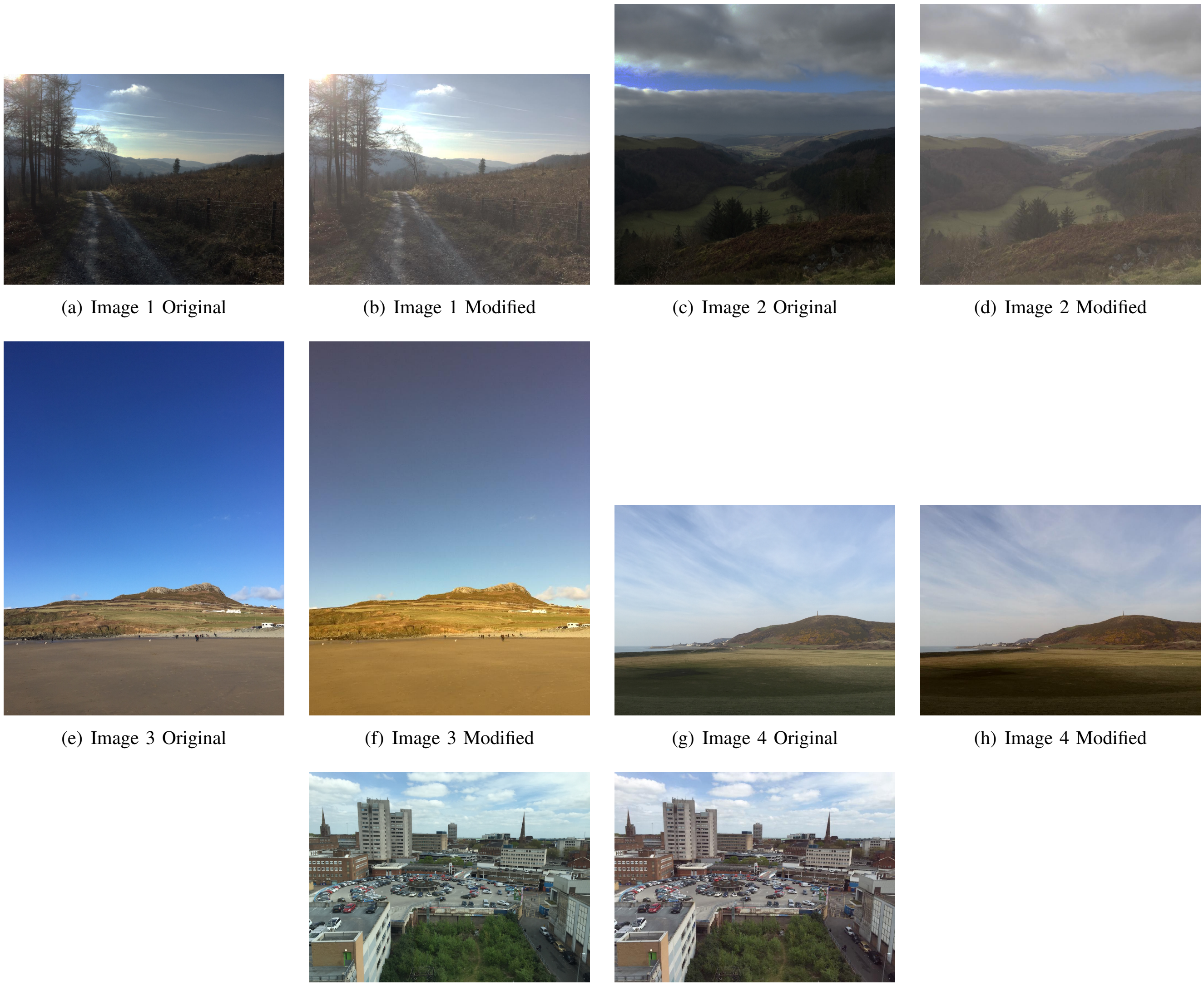
Therefore, a few key properties of the algorithm can be noted:

1) The algorithm works best when the shift factor ($\mu_{\text{ideal}} - \mu_{\text{actual}}$ in Equation 4) is small. If the shift factor is large then it is likely that a contrast change will be observed.
2) If the distribution of intensities across the different colour channels are very different, then the algorithm will produce an image with colour changes.
3) The algorithm will work best when the distribution of intensities are distributed in an approximately Gaussian distribution.

With these properties in mind it is possible to outline some possible areas for future improvement. Firstly, it would be beneficial to incorporate some kind of contrast adjustment *after* the brightness shift has occurred. This would enable the contrast changes seen in Figure 6 (b) and (d) to be alleviated. However, it is worth noting that performing a post-processing contrast adjustment would mean that the mean of the target image is no longer aligned to the middle value.

Secondly, the colour problem could be alleviated by performing the shifting operation in a different colour space. For example, using the HSV colour space would allow for the brightness information to be treated separately from the colour information. This would reduce the problem to be a single dimensional shift and so the changes in colour that can be seen in the current algorithm could be removed.

Thirdly, the shifting information shown in Table I could be used to quantify the effect of the algorithm and so could be used to help guide further processing. For example, if a low δs value is observed then it is likely that no further processing is needed; however, if a high δs value is observed then it is likely that the colour information has changed and so further processing, or even different processing, may be required.



(a) Image 1 Original (b) Image 1 Modified (c) Image 2 Original (d) Image 2 Modified

(e) Image 3 Original (f) Image 3 Modified (g) Image 4 Original (h) Image 4 Modified

(i) Image 5 Original (j) Image 5 Modified

Fig. 6. Results of performing the mean alignment algorithm on each of the five test images. For the first two images (a, c), the algorithm produces washed out results (b, d). Performing the algorithm on Image 3 (e) causes a drastic change in colour (f), whereas the results for Image 4 (g) are far more natural (h). The effect of the algorithm is least noticeable on Image 5 (i, j).

In conclusion, this paper has presented a method for automatic image enhancement based on shifting the pixel intensities for each colour channel so that they are mean-aligned to the middle intensity value. This mid-mean alignment algorithm is intuitive and easy to implement. In some cases the method is able to improve the quality of the image without any contrast or colour artefacts. However, in other cases the method, although simple, changes both the colour and the contrast of the image. This may not be a desirable feature and so further processing, or performing this algorithm in a different colour space, may help overcome these issues. It is expected that the proposed method could form the basis for a more advanced processing pipeline, and the shift information gained as a result of performing this mid-mean alignment algorithm could give important information to help guide the user in further enhancement processing.

References

[1] Histogram description. [Online]. Available: https://blackboard.aber.ac.uk/bbcswebdav/pid-686416-dt-content-rid-999089_1/courses/CS24110_AB0_2015-16/CS241_L07_LQ.pdf

[2] Histogram. [Online]. Available: https://en.wikipedia.org/wiki/Histogram

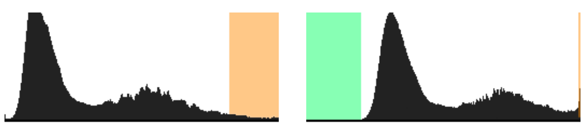
[3] Automatic contrast adjustment. [Online]. Available: https://blackboard.aber.ac.uk/bbcswebdav/pid-688647-dt-content-rid-1002902_1/courses/CS24110_AB0_2015-16/CS241_L09_LQ.pdf

[4] Colour space conversion rgb to xyz. [Online]. Available: http://www.easyrgb.com/index.php?X=MATH&H=02#text2

[5] Colour space conversion xyz to lab. [Online]. Available: http://www.easyrgb.com/index.php?X=MATH&H=07#text7

[6] Average filter function. [Online]. Available: https://blackboard.aber.ac.uk/bbcswebdav/pid-690794-dt-content-rid-1006217_1/courses/CS24110_AB0_2015-16/C241_L12_LQ.pdf

[7] "WritableRaster, Oracle Java SE7," http://docs.oracle.com/javase/7/docs/api/java/awt/image/WritableRaster.html, accessed: November 2015.



(a) original (b) modified

Fig. 7. Average RGB histograms from image 1 (Figure 6 (a, b)). The washed out effect seen in Figure 6 (b) occurs because all of the pixels highlighted in the orange region are mapped to maximum intensity. Also note the range of unfilled intensities in the darker region of the modified image (green bar).