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

Dimitar Tasev

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

This paper presents an approach to image enhancement, which combines multiple operations on the image, Automatic Contrast Adjustment, Luminosity Histogram Equalisation and Average Value Blur filter. This algorithm is hard to implement, and is expensive to compute, because there are multiple operations on the image, performed in sequence. The algorithm is tested on five test images and the results show that it can fix contrast and brightness issues, and improve the quality of the image. However, in some circumstances the results may not be satisfactory. These aspects are discussed in this paper.

I. INTRODUCTION

THE image enhancement to meet human visual perception is an important issue in general image processing, coding, machine vision. Nowadays, image enhancement is also widely adopted for medical image processing, texture synthesis, and speech recognition[1]. Image enhancement in computer graphics, the process of improving the quality of a digitally stored image by manipulating the image with software. It is quite easy, for example, to make an image lighter or darker, or to increase or decrease contrast. Advanced image enhancement software also supports many filters for altering images in various ways[2]. Image editing encompasses the processes of altering images, whether they are digital photographs, traditional photochemical photographs, or illustrations. Traditional analog image editing is known as photo retouching, using tools such as an airbrush to modify photographs, or editing illustrations with any traditional art medium. Graphic software programs, such as Adobe™ Photoshop® and GIMP[1], are the primary tools with which a user may manipulate, enhance, and transform images. Many image editing programs are also used to render or create computer art from scratch.

The problem presented is to create an algorithm for automatic image enhancement. The area has been developing recently, the biggest contributor being the creation of Instagram™, and the filters that they use to automatically enhance pictures, which have become very popular. Another popular automatic enhancement application is Google Photos [2], also providing a variety of filters, and automatic picture enhancement.

This paper is concerned with a new method to enhance a given image. The idea behind the approach is to merge the effects of three different operations, Automatic Contrast Adjustment, Luminosity Histogram Equalisation and Average Value Blur filter, each acting on a different area of the picture. In theory this should correct any contrast issues, any brightness issues, and remove noise artefacts, if such are available.

In this paper the methodology for the proposed image enhancement algorithm will be given at both high and low

level in Section II. The results are shown in Section III and then discussions and conclusion are drawn in Section IV.

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[3], 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

1: **function** NORMALHISTOGRAM(I) $\triangleright$ where $|I| = m \times n$,
2: K = MAX value of the colour space, 256 for RGB, 101 for LAB's L channel
3:
4: $\quad H = [0, K-1]$
5:
6: $\quad$ **for each** p **in** I **do**
7: $\quad\quad H(p) = H(p) + 1$
8: $\quad$ **end for**
9:
10: $\quad$ **return** H
11: **end function**

B. Detailed Description

The proposed enhancement method is a combination of filter and point-based histogram operations. To simplify the description of the operations, the algorithms assume that the picture

---

[1]http://www.gimp.org
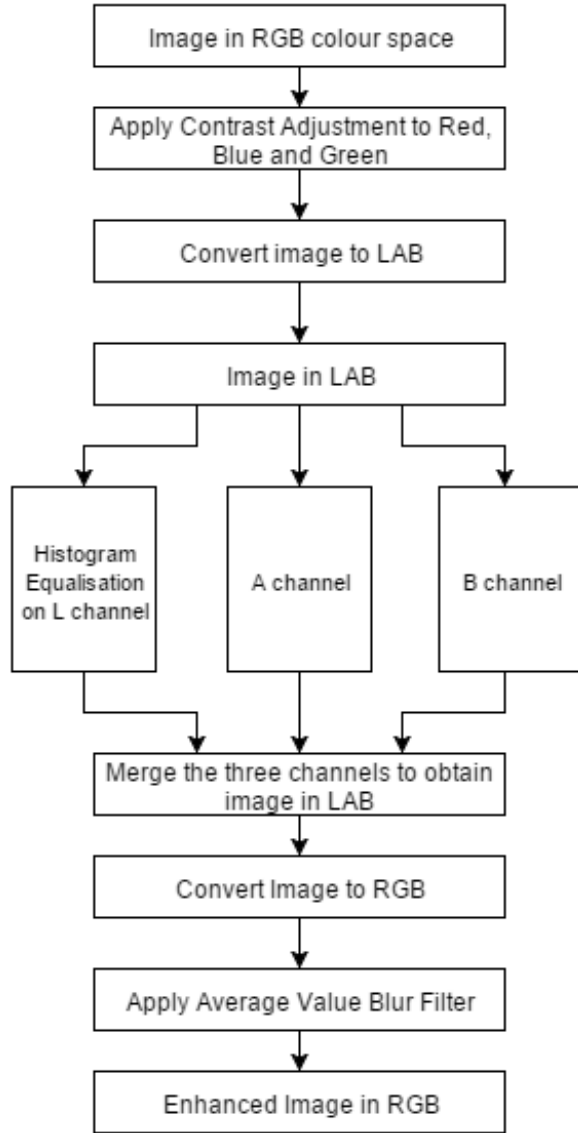[2]https://photos.google.com/

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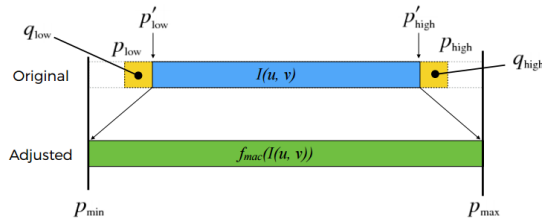

Fig. 2. An image showing how the range of values will be set when applying contrast adjustment to the picture.

is grayscale, however the final algorithms work 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[4][5]. In this paper the histograms are created using *JFreeChart*[6]. Figure 3 shows both a normal histogram and a cumulative histogram, both
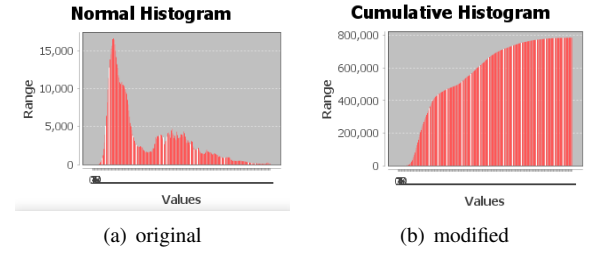

Fig. 3. Two images showing a normal histogram and a cumulative histogram of it.
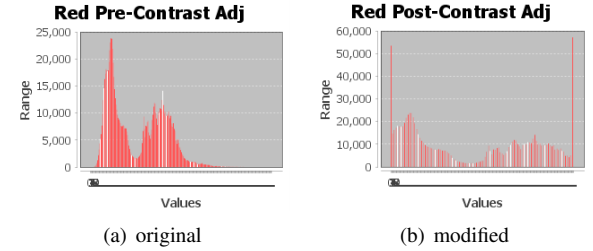

Fig. 4. Two images showing a normal histogram before and after being contrast adjusted.

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:

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

The first 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[7]. We have to identify two quantiles at the low p'_{low} and high p'_{high} end of the intensity spectrum, and map the pixel values inside them to the extreme values, the other pixels are then linearly mapped to the interval $[p_{min}, p_{max}]$ as seen on Figure 2. To calculate

---

Algorithm 2 Cumulative Histogram

1: **function** CUMULATIVEHISTOGRAM(I)
2:    ▷ where K = MAX value of the colour space, 256 for RGB, 101 for LAB's L channel
3:
4:    H = NormalHistogram()
5:    $CH = [0, K-1]$
6:
7:    **for each** v **in** H **do**
8:       $CH[v] = CH[v-1] + H[v]$
9:   **end for**
10:
11:    **return** CH
12: **end function**

the two quantiles we set a range for ignored pixels q. Using that range we can calculate the quantiles using the cumulative histogram:

$$p'_{low} = \min\{i \mid CH(i) \geq m \cdot n \cdot q_{low}\}$$
$$p'_{high} = \max\{i \mid CH(h) \leq m \cdot n \cdot q_{high}\} \quad (2)$$

Given that q follows:

$$0 \leq q_{low}, q_{high} \leq 1, q_{low} + q_{high} \leq 1 \quad (3)$$

After applying the formula to the cumulative histogram we will get ranges, which can be visually represented like this:
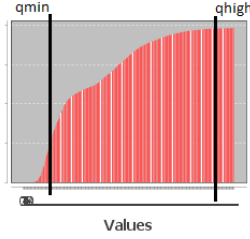
Then we loop through the pixels in the picture and apply the following function f to each pixel:

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

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 second operation of the algorithm is Histogram Equalisation on the Luminosity channel in LAB(CIELAB) colour space. It adjusts the contrast and brightness to appear as they would in a well exposed picture. As we can see in Figure 8, if the brightness of the picture is low, i.e. the picture is dim, the Luminosity channel is concentrated in the lower end of the spectrum. If the picture is bright, it is concentrated in the high end of the spectrum. In order to equalise the Luminosity channel, we first have to convert the image to LAB. This cannot be done directly, so the RGB picture has to be converted to XYZ colour[8] space first, and then converted to LAB[9]. 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. [7] 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 algorithm 3, implementing the following equation:

$$f_{he}(p) = \left\lfloor CH(p) \cdot \frac{K-1}{mn} \right\rfloor \quad (4)$$

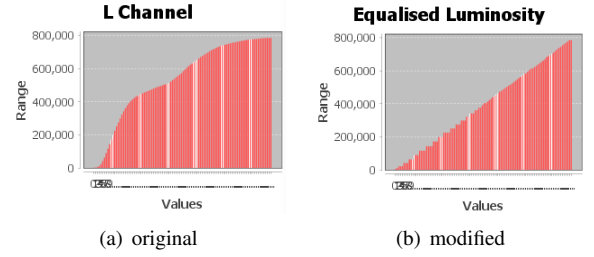(a) original   (b) modified

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

After the Equalised Histogram has been created, we have to loop through each pixel in the image and set its new value, which is stored in the Equalised Histogram, at the position of the pixel's current value:

$$P(v) = EQH(P(v)) \quad (5)$$

Algorithm 3 Histogram Equalisation

1: **function** HISTOGRAMEQUALISATION(I)
2:     ▷ where K = MAX value of the colour space, 256 for RGB, 101 for LAB's L channel
3:
4:     CH = CumulativeHistogram(I)
5:     $EQH = [0, K-1]$
6:
7:     **for each** v **in** CH **do**
8:         $EQH(v) = CH(v) \cdot \frac{K-1}{m \times n}$
9:     **end for**
10:
11:     **return** EQH
12: **end function**

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. Its effect is to reduce the noise in the picture, by applying a blur effect.[10] The strength of the blur, however, heavily depends on the size of the picture. Small pictures (i.e. 640x480 resolution) will have very strong blur, to the point of ruining the picture, and very big pictures (i.e. 1920x1020 resolution) will have nearly no visible change. The described filter operation acts on each channel in the same manner, using a filter operation function f. That is:

$$I'(u,v) = f\left(\frac{1}{9} \cdot \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)\right) \quad (6)$$

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 is **Automatic Contrast Adjustment** on the RGB channel. To achieve that the implementation will create and store a Normal Histogram 1 for each of the channels. When the Normal Histograms have been calculated, we can use them to create the Cumulative Histograms 2, repeated for each of the channels. Then we have to pick appropriate q_{low} and q_{high}, following the equation 3. Now we can calculate p'_{low} and p'_{high} for each of the colour channels using the Cumulative Histograms of the channels 2. After we have calculated p'_{low} and p'_{high}, we apply the function II-B to each pixel to determine it's new value. After the Contrast Adjusted image is produced, we perform the second operation from the algorithm, the Luminosity Histogram Equalisation.

In order to create a **histogram** on the **Luminosity channel**, we have to convert each pixel to LAB colour space, before performing any operations. As this cannot be done directly from RGB, we first need to convert it into XYZ [8] colour space, and then into LAB [9] colour space. The implementation has to loop through each pixel, converting it to LAB colour space, and creating a Normal Histogram 1 of the Luminosity (L channel for short). When the Normal Histogram is calculated, a Cumulative Histogram 2 is created from it, and finally an equalisation 3 is performed on the Cumulative Histogram. After the Equalised Histogram has been completed, the implementation loops through all of the pixels, converting each one to LAB, and then adjusts the intensities 5, using the new values from the Equalised Histogram. When the pixels' values are adjusted, the implementation has to convert the pixel from LAB back to XYZ [11], and then to RGB [12]. A clamping operation is performed as well, before finally setting the new value of the pixel.

The last operation performed by the enhancement algorithm is the **Average Value Blur** filter. It works in RGB colour space, so it does not need to convert the picture. The filter is easy to implement, it calculates the average value of the pixel, plus the surrounding pixels 6. After it loops through every pixel, a blurred image is produced, which is also the final product image of the algorithm.

This Image Enhancement algorithm is very expensive to compute, and will be visibly slow when executed on older computers. It's implementation can be optimised by using an efficient pixel accessing mechanism, such as Java's *WritableRaster*, but it's run time will still be noticeably long. This is due to combining multiple operations, each of them acting on the *whole* image, in sequence.

III. Results

The proposed image enhancement method is tested on the five provided images. The results of performing the algorithm are shown in Figure 6.

The image enhancement algorithm causes a change in all of the images. The least change is seen in Image 5 (Figure 6 (i,

j)). The most natural looking images are Image 2 and Image 5; in the rest of the images there are large colour changes, bringing out unnatural colours. The blur from the filter is noticeable in all of the images.

The modified version of Images 1, 3 and 4, all have unnatural colours appear. This is due to the Automatic Contrast Adjustment remapping the images to use all of the available intensities, when the original picture pixels' intensity was mostly concentrated on a particular range. This can be seen when looking at the histograms. Figure 8 shows the histogram before and after processing Image 1. We can see how before the algorithm, the image's pixel intensity values were focused mostly in a particular range, but after being processed by the algorithm, they are spread out, bringing out colours, that would not be very well visible in the original picture.

The second operation in the algorithm, the Histogram Equalisation on the Luminosity channel, is responsible for the pictures looking clearer than the original, and the brightness change that can be seen in all of them. Images 1 and 2 have had their brightness increased after being processed, due to the original picture being dim. Images 4 and 5, have only a slight difference from the original picture's brightness. However, since they appear to be a bit washed out in the original pictures, after equalisation it can be see that they are clearer and less washed out. The same effect appears on Image 3, but it is weaker. Images 3, 4 and 5 have not been significantly changed by the Luminosity channel Histogram Equalisation, because the original pictures are fairly well exposed and have good brightness.

The final operation in the algorithm, the Average Value Blur filter is responsible for the visible blur in the pictures. The blur might be a little bit too strong than desirable, because the pictures are small in size.

IV. Discussion & Conclusions

The results show that the proposed method works best when the picture isn't heavily focused on one colour, causing the Automatic Contrast Adjustment to bring out unnatural colours, that are not seen in the original pictures. If pictures appear to be dim, or washed out, the method will adjust them to appear as if they were well exposed, brighter/less washed out. The Average Value Blur filter can be very strong on small pictures, reducing their quality rather than enhancing it, and unnoticeable on large pictures, making it rather unnecessary. The picture sizes are small, that is why the blur effect is a bit stronger than what would be optimal.

From that we can conclude that the algorithm works well with pictures that do not have a single dominating colour. If the pictures happen to be underexposed and appear to be dim or overexposed and washed out, the method will work well, and will properly adjust them. However having the Average Blue Filter's usefulness depend on the picture size is impractical.

An improvement for the algorithm would be to add a different blur filter one which is consistent and independent of picture size. Also it would be beneficial to try and find a way to avoid the unnatural colours from the Automatic Contrast Adjustment. Adding an edge enhancement operation,

(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
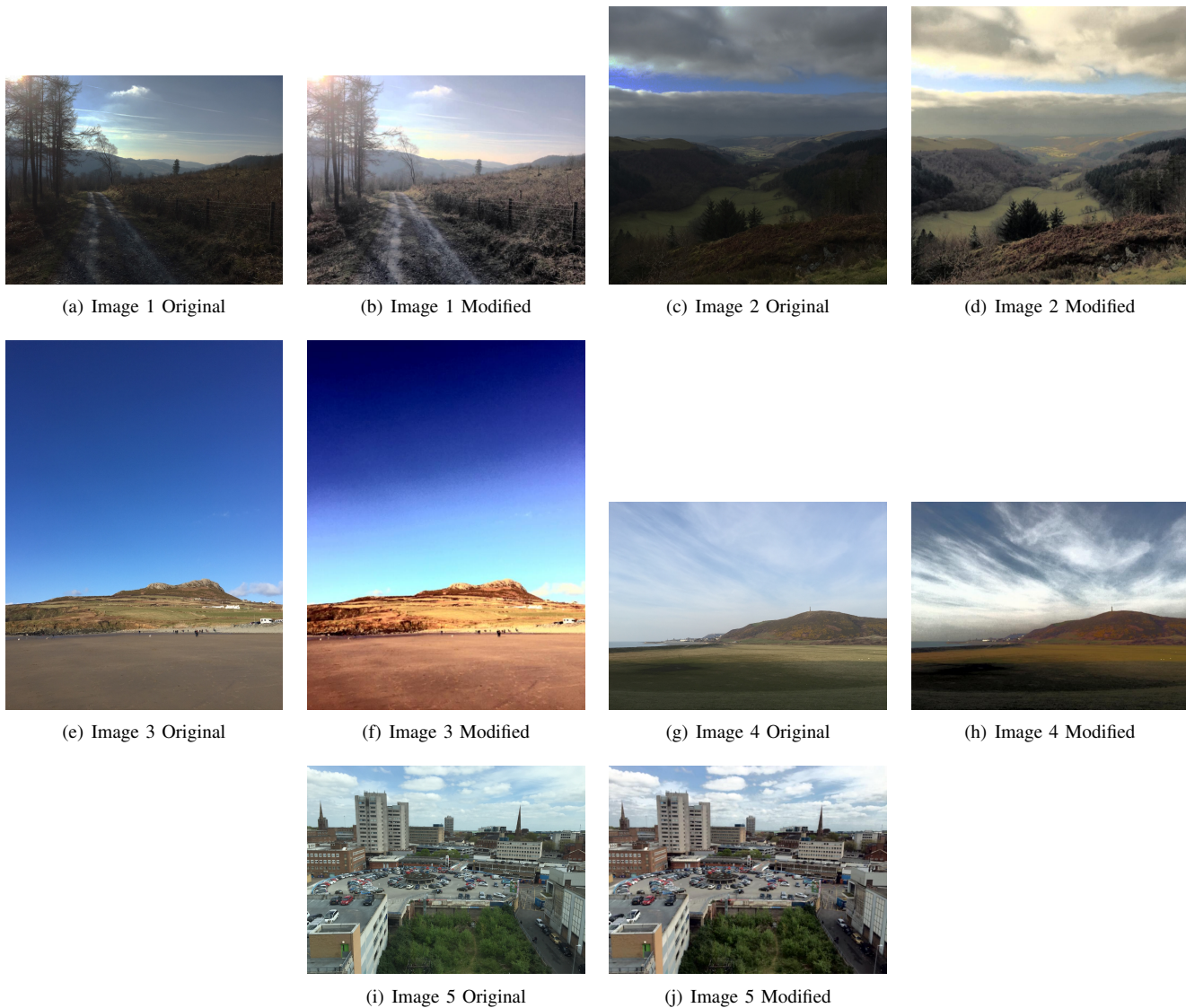
(i) Image 5 Original

(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). Results of performing the described image enhancement algorithm on each of the five test images. The effect of the algorithm is noticeable in all of the images, they are all brighter and the blur is visible. For images (e, g) the images produced do not look natural (f, h). The algorithm's effect on image (a) has brought in colours in the sky that are not in the original picture (b). For images (c, i) the algorithm produced more natural-looking results.
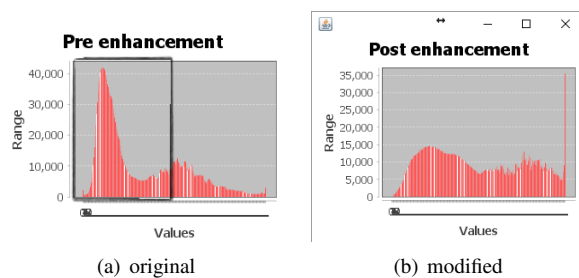

(a) original

(b) modified

Fig. 7. Average RGB histograms from image 1 (Figure 6 (a, b)). The unnatural colours occurs, because the original pixel intensity distribution is mostly focused in a particular range, surrounded by a black square, but after contrast adjustment the full range is used, bringing out unnatural colours.

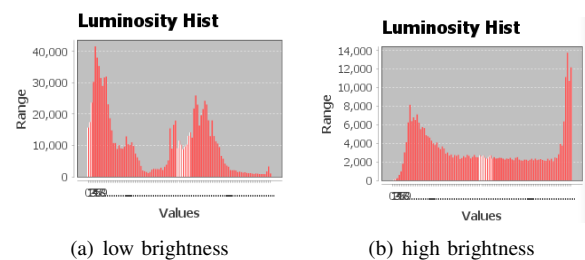(a) low brightness

(b) high brightness

Fig. 8. Luminosity channel histograms of a low brightness picture (left) and high brightness picture (right).

which will sharpen the picture might also be beneficial for this image enhancement algorithm.

In conclusion, this paper has presented a method for automatic image enhancement, incorporating multiple image enhancement operations, each acting on a different area of the picture, with one of the operations being performed in a different colour space. This makes it a hard algorithm to implement, and expensive to execute, as it has multiple operations, histograms and colour space conversions, all of which are necessary parts to make the algorithm work. In some cases the algorithm is able to improve the quality of the picture, adjusting the brightness to imitate a well exposed picture, good contrast range using all of the possible values and reducing the noise of the picture. However, in other cases the contrast adjustment can bring out unwanted unnatural colours, that are not visible in the original picture. The strength of the Average Value Blur filter is also of concern, because having the strength depend on the size of the picture is impractical.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*.

[2] Image enhancement. [Online]. Available: http://www.webopedia.com/TERM/I/image_enhancement.html

[3] G. Jeon, "Color image enhancement by histogram equalization in heterogeneous color space."

[4] 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

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

[6] Jfreechart. [Online]. Available: http://www.jfree.org/jfreechart/

[7] 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

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

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

[10] 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

[11] Colour space conversion lab to xyz. [Online]. Available: http://www.easyrgb.com/index.php?X=MATH&H=08#text8

[12] Colour space conversion xyz to rgb. [Online]. Available: http://www.easyrgb.com/index.php?X=MATH&H=01#text1