

# Novel Histogram Processing for Colour Image Enhancement

Jiang Duan and Guoping Qiu

School of Computer Science, The University of Nottingham, United Kingdom

{jxd | qiu} @cs.nott.ac.uk

## Abstract

*In practice histogram equalization often produces images with unnatural appearances and visually disturbing artefacts. One of the reasons for these unwanted effects is that histogram equalization attempts to force the output image to have a uniform pixel distribution regardless of what the original image's pixel distribution may be. In this paper, we present a novel histogram processing algorithm which takes into account the original image's pixel distribution in the equalization process. The method uses a single parameter to control the degree of contrast enhancement to ensure that the output have an enhanced appearance which is also faithful to that of the original image and is free of unwanted visually disturbing artefacts. We first develop the algorithm for the luminance channel and then extend the method to the colour components. We present experimental results to demonstrate the better performances of our new method over established methods in the literature.*

## 1. Introduction

Histogram equalization for greyscale image contrast enhancement is a well-known technique in the literature of image processing. Given a greyscale image  $I$  with grey levels in the range  $[0, L-1]$ , its normalized histogram is a discrete function  $H(l)=n_l/n$ , where the  $l$  is the  $l$ th grey level,  $n_l$  is the frequency of occurrence of the corresponding grey level and  $n$  is the total pixel population in the image. Histogram equalization can be achieved by using a Cumulative Distribution Function (CDF), and its discrete version is as follows:

$$S_l = T(l) = (L-1) \sum_{j=0}^l H(j) = (L-1) \sum_{j=0}^l \frac{n_j}{n} \quad (1)$$

The equation uniformly distributes the pixel population of an image to all the available grey levels of visualization devices, and thus maximizing the contrast. With the rapid development of colour media, the requirement for enhancing colour image has become more and more demanding. Histogram equalization, originally designed for greyscale images has been extended to enhance colour images [2, 3, 4, 5,]. Reference [3] proposed an algorithm to generate an almost uniform colour histogram, which, however, would change the ratios of the RGB components and thus producing hue-shifting related artefacts. As the

authors claimed, the algorithm was more suitable for better visualization of pseudo-colour scientific pictures than for ordinary image enhancement. Hue is the most basic attribute of a colour and normally the change of hue would result in unacceptable artefacts for ordinary colour image enhancement and even scientific colour images visualization. In order to avoid hue-shifting related problems, other ideas [2, 4, 5] uniformly spread the histogram under the constraint that keeps the hue of each pixel unchanged. That is to say, the equalization process is applied only to the luminance and saturation components of images.

In practice, however, it is well known that the appearances of the resulting image from the equalization process are not visually pleasing for most of the cases. When the equalization process is applied to greyscale images or the luminance component of the colour images, the maximized contrast usually leads to visually annoying artefacts. For colour images, these artefacts become more obvious when the saturation component joins in the equalization process even though the hue component is left untouched. The visually unsatisfactory result of equalization is not unexpected. The cumulative distribution function performed in the equalization process forces the output pixels to have a uniform distribution *independent* of the input pixels distribution. However, generally speaking, individual images are characterized by their own specific pixel distributions and do not share a common property of uniform pixel distribution. For instance, a dark image gives us a "dark" sensation because pixels with low intensity levels are dominant in the picture. If these pixels were evenly scattered to all the intensity levels using histogram equalization, the dark sensation of the image would be destroyed. Therefore, the original pixel distribution is a significant attribute in conveying and characterizing the pictorial information for an image. In most cases, ignoring this factor by the histogram equalization process causes the processed images to have unrealistic visual appearances and unwanted artefacts.

In this paper, we present a novel algorithm that elegantly strikes a balance between histogram equalization (contrast enhancement) and the maintenance of the image's original pixel distribution (to be faithful to the original image's visual appearances) to produce realistic and good-looking pictures.

## 2. The development of the algorithm for the luminance component

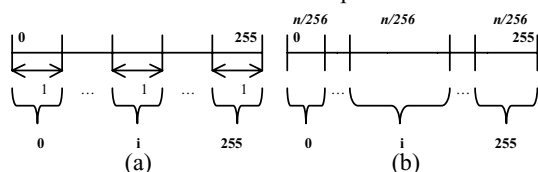
In this section, we focus on the investigation of the algorithm for the luminance component of colour images. The luminance  $Lu$  (continuous valued) of a colour image can be computed as a weighted sum of RGB components.

$$Lu = 0.299*R + 0.587*G + 0.114*B \quad (2)$$

When the equalization process is applied to the luminance component, the schematic of this process can be illustrated in the Figure 1 (b). The computed luminance values of an image ranges from 0 to 255. Divide this range into 256 intervals such that each interval has the same number of pixels falling into, and pixels falling into the same interval are represented by the same display intensity level.

The concept of maintaining the original pixel's distribution is in fact leaving the original data unchanged. In order to correspond to the 256 discrete values as equalization process, these computed floating values of the luminance can be quantized to 256 discrete values. This process is illustrated in Figure 1 (a), where the  $[0, 255]$  luminance range is equally divided into 256 intervals and the values falling into the same region would be assigned the same display intensity level. These quantized values still faithfully maintain the original pixels distribution with (small) quantization error.

Enlightened by these two diagrams, if we cut the  $[0 - 255]$  luminance range in such a way that the 255 cutting points fall in between the cuts that evenly distribute the pixel population and the cuts that maintain the original pixel distribution, the resulting images would blend the characteristics from both of them and thus will not only retain their original sensation but also will have an improved contrast.



**Figure 1.** (a) Divide luminance range into 256 equal length intervals. (b) Divide the luminance range into 256 intervals such that the total number of pixels  $n$  is evenly distributed into each interval. All pixels falling into the same interval are mapped to the same display value.

### 2.1. The algorithm

Let  $Lu(x, y)$  be the luminance input from Eq. (2). A high resolution histogram,  $\{H[k] = \text{Prob}[Lu(x, y) = k]\}$ , is first constructed to ensure the accuracy of the algorithm. In this study, a 1,000,000-bin histogram is used. Based on the constructed histogram, the

algorithm then divides the luminance range  $[0, 255]$  into 256 intervals using a hierarchical division procedure. First, a control parameter  $\alpha$ ,  $0 \leq \alpha \leq 1$ , is defined (this is the only user defined parameter in the algorithm, and its meaning will be explained shortly). We then find a value  $\beta_0$ ,  $0 \leq \beta_0 \leq 255$ , such that pixel populations on both sides of the value are equal:

$$\sum_{k=0}^{\beta_0} H[k] = \sum_{k=\beta_0}^{255} H[k] \quad (3)$$

We then divide the dynamic range into 2 segments by finding a cutting value,  $C_0$ :

$$C_0 = \frac{0+255}{2} + \alpha \left( \beta_0 - \frac{0+255}{2} \right) \quad (4)$$

The dynamic range is now divided into two intervals:  $[0, C_0]$  and  $[C_0, 255]$ . We then perform the procedure recursively for each of the intervals and divide each into two segments. After 8 iterations, the dynamic range would have been divided into  $256 = 2^8$  intervals. Pixels falling into the first interval are then represented by the same display value of 0; pixels falling into the second interval are assigned the same display value of 1 and so on.

The only control parameter the user has to set in the algorithm is  $\alpha$ . If  $\alpha = 0$ , the mapping leave the original image unchanged, if  $\alpha = 1$ , the mapping is histogram equalized. Setting  $0 \leq \alpha \leq 1$ , we control the mapping between equally distributed pixel population and the maintenance of the original pixel distribution in a very simple and elegant way. With the single parameter  $0 \leq \alpha \leq 1$  we can strike a balance between maximizing contrast and preserving the visual sensation of the original images. In fact our experiences showed that setting  $\alpha = 0.4$  to  $\alpha = 0.6$  worked very well for a variety of images. Note that a similar procedure has been successfully used for the display of high dynamic range images [6].

### 2.2. Experimental results

The proposed algorithm has been applied to many images and shows good results. Figure 2 shows one of these images and its versions from histogram equalization and our algorithm. In the figure, their histograms are also shown for easy comparison. Although the resulting image from histogram equalization improves the contrast compared with the original, it loses the naturalness of the original image and introduces unwanted artefact (See the blue speckles scattered in the leaves). The image computed by our algorithm produces a histogram that not only follows the shape of the original histogram but also is broadened across the whole range. The result appropriately follows our analysis and fulfils our expectation. The effect of the specially broadened histogram is that the computed image shows more contrast and visibility and at the same time the

calculated image looks more realistic and avoids artefacts.

Our method is computationally very simple; the parameter can be controlled at an interactive speed even for very large size images thus making the effects of changing the parameter instantly visible. To process the image of 768 x 512 pixels in on a Pentium 4 with 1800MHz CPU using C programming language, the whole computational process takes about 0.7s.

### 3. The extension of the algorithm to colour component

Colour equalization is used as a powerful technique to fully exploit the colour space to achieve rich colour effects in the resulting images. To avoid hue-shifting related artefacts, methods in [2, 4, 5] kept the hue component unchanged and performed the equalization process on the saturation component. Although these methods make the computed images colourful and eliminate hue-shifting related artefacts, the colour equalized images still present visually disturbing artefacts in most of the cases. Similar to histogram equalization applied only to the luminance component, these artefacts arise from the mechanism of equalization that completely ignores the original data distribution. In this session, the developed method for luminance component in section 2 is extended to the colour component, which attempts to strike a balance between keeping the original colour distribution and achieving a uniform colour distribution.

When humans see a colour, we interpret it by its hue, saturation and brightness [1]. However the RGB colour space does not correspond well to how we interpret colours in such way. Hence the processing of colour image requires a different colour space or colour model that intuitively describes our interpretation of colours. HSI colour space that is widely used in the colour image processing is chosen in our study.

In HSI colour space, similar to other colour equalization algorithm, our algorithm is only applied to the saturation component in the divided hue regions and the hue of each pixel is left unchanged to avoid the hue-shifting related artefacts. In our algorithm, the computation needs the attainable maximum saturation of display devices that in fact is a function of different luminance in the same hue plane region [5] as shown in Figure 3. If the equalization process is carried out in the whole hue plane as the method in [2] that ignored the relationship between luminance and the attainable maximum saturation range, cross-session artefacts would be introduced.

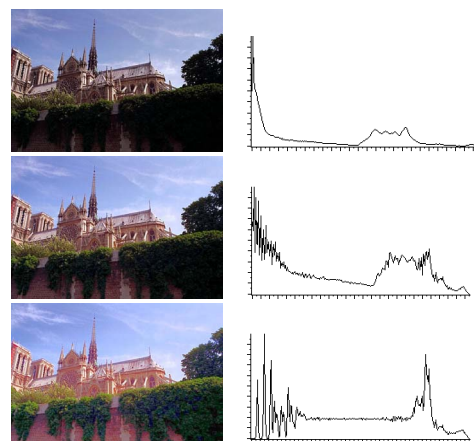
Our algorithm applied to colour component is as follows:

1. The HSI colour space is divided into  $M$  hue regions and each hue region is then divided into  $N$

luminance regions. Now the whole colour space is segmented into  $M \times N$  regions. Similar to the work in [5], the maximum realizable saturation  $Sat_{max}(m,n)$  for  $m=1,2,3,\dots,M$  and  $n=1,2,3,\dots,N$  are found by comparing all the saturation values for all possible RGB combinations falling into the region and choosing their maximum saturation value as the maximum realizable saturation values for the corresponding region. Once these values have been found, they can be saved for future repeated use by our algorithm.

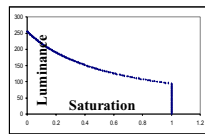
2. The pixel RGB values of an image are converted to HSI values and our algorithm introduced in Section 2 is performed on the saturation component in each region. The specific operation is: Find all the pixels belonging to a region  $(m, n)$ ; A high-resolution histogram in terms of their saturation values is created; the saturation range  $[0, Sat_{max}(m,n)]$  is then divided hierarchically into  $K$  intervals using the procedure introduced in section 2.1; similarly  $\alpha$  controls the mapping between equally distributed saturation and the maintenance of the original saturation distribution in a very simple and elegant way. After the division, the saturation falling into the  $r$ th interval would be grouped together and represented by the same saturation value  $r * Sat_{max}(m,n) / K$  for  $r = 1, 2, 3, \dots, K$ .

3. The processed HSI values are converted back to RGB values. Since each of the maximum saturation is not decided from an infinitesimal region, the converted RGB values might be out of realizable RGB space. In such kind of circumstances, these RGB values for the out-of-space pixels can be brought back by clipping or scaling by  $255/\max(R,G,B)$ . In our study, the latter is used because it keeps the RGB ratio and thus leaves the hue component unchanged.



**Figure 2,** From top to bottom: The original image and its histogram. The computed image by applying our algorithm to the luminance component ( $\alpha=0.45$ ) and its histogram. The computed image by applying histogram equalization to luminance component and its

histogram. The original image is courtesy of Raanan Fattal, Dani Lischinski and Michael Werman.



**Figure 3**, The attainable maximum saturation at different luminance value for a fixed hue region  $0^\circ - 6^\circ$

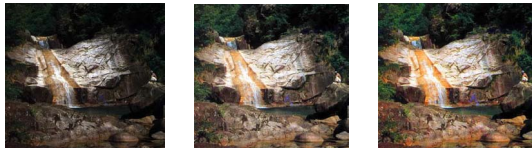
### 3.1. Results

In the implementation of the algorithm, a compromise should be made between the division of the HSI colour space into  $M \times N$  regions to apply our algorithm and the computational time. The more regions are divided, the less cross-session artefacts there are, but the more computational time is needed. In our experiments, we used the division of 60 hue regions and 17 luminance regions. To process the image of  $768 \times 512$  pixels, the *whole* computational process takes about 51.9s on an 1800MHz CPU PC.

Figure 4 shows an image and its versions from colour equalization algorithm by [5] and our algorithm. The algorithm by [5] is chosen because their work is similar to ours, but their algorithm is targeted for an equalized colour distribution. In order to facilitate comparisons, their algorithm is implemented in the same HSI colour space as ours but not in their C-Y colour space. The original image in Figure 5 shows good contrast but does not look colourful. Hence both our algorithm and [5]'s algorithm are only applied to the saturation component of this image. The method of [5] makes the processed image over-saturated and introduces disturbing artefacts, which makes the resulting image no longer look natural. On the other hand, the image computed by our algorithm appears natural, realistic, and more colourful, which also avoids disturbing visual artefacts. Figure 5 shows another example.



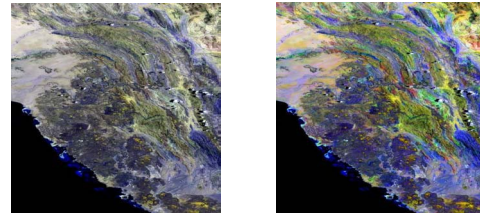
**Figure 4**, Original image (Left), result of our algorithm  $\alpha = 0.5$  (Middle), and colour equalization (Right) [5].



**Figure 5**, Original image (Left), result of our algorithm applied to the luminance only  $\alpha = 0.6$  (Middle), and colour

enhancement by applying our algorithm to the saturation component of the image in the middle  $\alpha = 0.5$  (Right).

A potentially useful application of our colour enhancement method is in the visualization of scientific images, where it may be useful to make the image have as many distinctive colours as possible and at the same time does not introduce visual artefacts. Figure 6 shows colour enhancement result on a Landsat image found on the Internet.



**Figure 6**, Left: original LandSat images. Right: Colour enhancement image by our algorithm.

### 4. Concluding Remarks

In this paper, a novel method that balances the requirements of both appearance enhancement and being faithful to the original appearance of an image has been proposed and applied to the enhancement of full colour images. Results have shown the effectiveness of our algorithm in improving the contrast and colourfulness of the original images without introducing artefacts caused by traditional equalization process.

### Reference

- [1] RC Gonzalez and R. E. Woods, "Digital Image Processing," 2nd Edition, Prentice Hall, 2002
- [2] I.M. Bockstein, "Color equalization method and its application to color image processing," J. Opt. Soc. Am., vol. 3, no.5, pp. 735-737, 1986.
- [3] E. Pichon, M. Niethammer and G. Sapiro, "Color histogram equalization through mesh deformation" ICIP 2003, 2:117-120
- [4] A.N. Venetsanopoulos P.E. Trahanias, "Color image enhancement through 3-D histogram equalization," in Proceedings, 11th International Conference on Image, Speech and Signal Analysis, vol. III, pp. 545-548, 1992.
- [5] Weeks, A. R., Hague, G. E. and Myler, H. R., "Histogram equalization of 24-bit color images in the color difference (C-Y) color space," Journal of Electronic Imaging, Volume 4, Issue 1, pp. 15-22, January 1995.
- [6] J. Duan and G. Qiu, "Fast Tone Mapping for High Dynamic Range Images", ICPR2004, 17th International Conference on Pattern Recognition, Cambridge, United Kingdom, 23 - 26 August, 2004
- [7] M. Faverau, S. Soca, J. Balon, and M.Cattoen, "Adaptive contrast correction using real-time histogram modification," SMPTE J., pp. 488-491, 1984