A Novel Framework for Optimal RGB to Grayscale Image Conversion

Yi Wan, Qisong Xie
School of Information Science and Engineering
Lanzhou University
Lanzhou, China
Email: wanyi@lzu.edu.cn, 905556586@qq.com

***Abstract*—Nowadays most images are shot as color images. Yet in situations such as printing or pattern recognition they also need to be converted to grayscale images. The most important problem in this conversion process is to preserve the image contrast. In this paper we make two contributions. In the simple yet popular line projection approach, which is also adopted in Matlab, we propose an entropy-based optimization framework to choose the optimal line direction so that all the pixel color vectors in an image have the most spread-out projections, thus increasing the grayscale image contrast. Secondly, we make use of histogram specification on all the projection points to further increase the image contrast. Experimental results show that the proposed framework produces enhanced results compared to typical other methods.**

***Keywords*-color image; grayscale image; color to grayscale image conversion; decolorization**

I. Introduction

Color to grayscale image conversion finds applications in areas such as image editing, printing, pattern recognition, etc. Because of fewer luminance representation bits, the converted grayscale image usually has poorer contrast than the original color image. This can cause poorer perception and other image processing performance (e.g., [1]). The goal is to preserve as mush contrast information as possible.

One simple yet popular color to grayscale image conversion approach is the line projection described as

$$I = \alpha_r R + \alpha_g G + \alpha_b B, \tag{1}$$

where the non-negative coefficients $\alpha_r, \alpha_g, \alpha_b$ satisfy

$$\alpha_r + \alpha_g + \alpha_b = 1. \tag{2}$$

Different choices of the coefficients have been proposed. For examples, $(\alpha_r,\ \alpha_g,\ \alpha_b) = (1/3,\ 1/3,\ 1/3)$ is used in [9], $(\alpha_r,\ \alpha_g,\ \alpha_b) = (0.3,\ 0.59,\ 0.11)$ is proposed in [10] and used in the image editing software GIMP and the popular numerical computing software Matlab, $(\alpha_r,\ \alpha_g,\ \alpha_b) = (0.2126,\ 0.7152,\ 0.0722)$ is used in HDTV [9]. Although these choices make use of some psycho-visual characteristics such as the logarithmic nature of human brightness perception, they all suffer drawbacks such as the lack of adaptability for difference images and the lack of optimality to make full use of the grayscale intensity levels.

Besides the above simple line approach, more sophisticated methods have also been proposed. In [2] an eigenvalue-weighted linear sum of subspace projections method is applied after converting the RGB color space to the YCbCr color space. In [3], a weighted intensity component and a weighted chrominance component is first adaptively combined while maximizing the overall Sobel gradient of the resulting image. Then it applies a second stage of local contrast enhancement to further improve the result. [4] uses the bilateral filter to try to preserve the multiscale contrast.

The rest of the paper is organized as follows. In Section II we present a new framework that tries to find the optimal line direction so that the projections from all color pixel vectors are the most widely spread-out. In Section III we propose a histogram specification framework that further acts on the grayscale image obtained in Section II. Experimental results are presented in Section IV and we draw conclusions in Section V.

II. Optimal Line Direction Selection through Entropy Optimization

In a common color image each pixel value is described by the R, G and B components, which can be visualized as vectors in a unit cube. One common and fast simple approach to this conversion problem is to choose a unit direction vector $\alpha = (\alpha_r, \alpha_g, \alpha_b)$, as illustrated in Fig. 1, and project the RGB vector onto this line α as described in (1) and (2). Classical methods of choosing the vector α usually only considers the psycho-physical properties of the human visual system (HVS). Although such choice is intuitively plausible, it cannot adapt to the vast variations across different color images.

In this section we propose a new framework to adaptively choose the vector α so as to make full use of all the grayscale levels, generally 256. Toward this goal we use the entropy of the histogram of all the projected I values. Suppose a color image has N pixels. For any given α satisfying (2), (1) would give N grayscale values $I_1, I_2, \cdots, I_N$. In this paper we use the most common 8 bit depth for storing the grayscale intensity I values, resulting in $K = 256$ levels. Then we can obtain the histogram of the N grayscale values $p_1, p_2, \cdots, p_K$, i.e.,

$$p_k = N_k/N, \quad k = 1, 2, \cdots, K. \tag{3}$$

978-1-5090-0768-4/16 $31.00 © 2016 IEEE
DOI 10.1109/IHMSC.2016.201

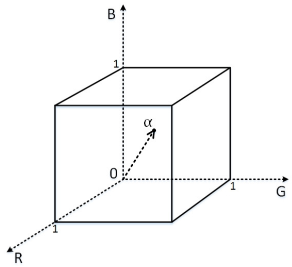

Figure 1. Illustration of the line direction α onto which all color pixel value RGB can project to obtain the grayscale value.

where N_k is the number of elements in the set $\{n \in \{1, 2, \cdots, N\} : I_n = k\}$.

Recall that the entropy of the probability distribution $p_1, p_2, \cdots, p_K$ is defined as [11]

$$H = -\sum_{k=1}^{K} p_k \log p_k. \tag{4}$$

H is a concave function on $(p_1, p_2, \cdots, p_K)$ and achieves its maximum when $p_1 = p_2 = \cdots = p_K = 1/K$. This property realizes our intention of spreading the histogram as much as possible. The analytic formula for the maximum of H in (4) is difficult to find. So we can discretize α_r and α_g and carry out an exhaustive search (Note that $\alpha_b = 1 - \alpha_r - \alpha_g$.) In Fig. 2 we show the results on three test images. We see that the proposed method adaptively finds the optimal vector α that produces visually better grayscale images when compared with other methods. From the histograms we can also see that the enhanced results come from flatter histograms. In addition, we see that even with optimal choice of α, the resulting histogram is still not truly flat (see the 6th row in Fig. 2). In the next section we develop a histogram specification method to further flatten the histograms.

III. Histogram Specification for Further Contrast Enhancement

Histogram specification has been used for image contrast enhancement (e.g., [8]). The key is to design an ordering rule. In the context of color to grayscale image conversion, since the key thread in the proposed framework is to make full use of all the grayscale levels, we can just spread out the histogram obtained in the previous section evenly across the grayscales $b_0, b_0 + 1, \cdots, b_1$ where $0 \le b_0 < b_1 \le K - 1$ while maintaining the grayscale value orders, where the choice of the two parameters b_0 and b_1 will be discussed in the next section. This can be done easily as follows. First order the grayscale intensity values $I_1, I_2, \cdots, I_N$ in the ascending order to obtain $I'_1 \le I'_2 \le \cdots \le I'_N$. Then do

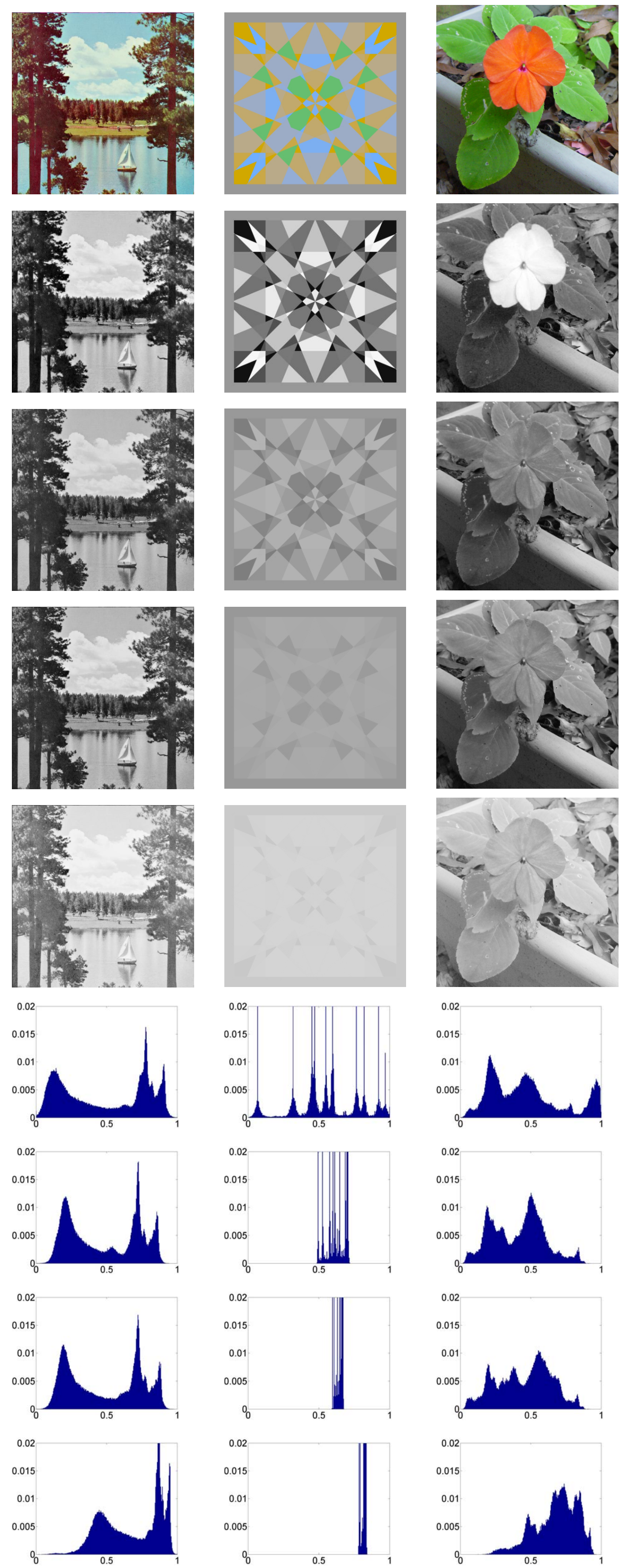

Figure 2. Illustration of the optimal selection of α. First row: three test color images. 2nd through 5th rows: Results by the proposed method and $\alpha = (1/3,\ 1/3,\ 1/3)$ [9], $\alpha = (0.2126,\ 0.7152,\ 0.0722)$ used in Matlab [10] and $\alpha = (0.3,\ 0.59,\ 0.11)$ [9]. 6th through 9th rows: Corresponding histograms of the previous results.

the following mapping

$$I'_n \;\rightarrow\; I''_n = \left\lfloor n \cdot \frac{b_1 - b_0}{N} \right\rfloor + b_0 \tag{5}$$

where $\lfloor \cdot \rfloor$ is the floor function.

Now combining the entropy-based optimization for α and the histogram specification just described above, we arrive at the complete algorithm as follows.

Complete details of the proposed method:

Step 0: Discretize the space of $\alpha = (\alpha_r, \alpha_g, \alpha_b)$ to get $\alpha_m, \quad m = 1, 2, \cdots, M$.

Step 1: Given any color image of N pixels with RGB values $(R_n, G_n, B_n), \; n = 1, 2, \cdots, N$. For each m, compute

$$I_{m,n} = \alpha_{m,r} R_n + \alpha_{m,g} G_n + \alpha_{M,b} B_n. \tag{6}$$

Step 2: Compute the histogram p_m for each I_m as

$$p_{m,k} = N_{m,k}/N \tag{7}$$

where $N_{m,k}$ is defined similar as N_k in (3).

Step 3: Compute the entropy H_m from p_m using (4).

Step 4: Set

$$I_n = I_{m^*,n} \tag{8}$$

where

$$m^* = \underset{m}{\operatorname{argmax}}\, H_m \tag{9}$$

Step 5: Sort $\{I_n\}_{n=1}^{N}$ in the ascending order to get $I'_1 \le I'_2 \le \cdots \le I'_N$ with $I_n = I'_{n'}$.

Step 6: Obtain the mapping

$$I_n \;\rightarrow\; I''_n = \left\lfloor n' \cdot \frac{b_1 - b_0}{N} \right\rfloor + b_0 \tag{10}$$

where I''_n is the final grayscale intensity value for pixel n.

IV. Experimental Results

In this section we present the experimental results of the proposed method and compare it with several other published methods. All codes are written in Matlab. For the proposed method, three parameters need to be determined. The first is M. To strike a balance between accuracy and speed, we let α_r to vary from 0 to 1 with step 0.1 and α_g to vary from 0 to $1 - \alpha_r$ with step 0.1 also. This results in $M = 66$. For the parameters b_0 and b_1, we tried different choices of b_0 and b_1 near 0 and 255 respectively. Empirically we find that the visual perception difference is minor. So in the following we fix $b_0 = 0$ and $b_1 = 255$.

We compare five other methods with the two proposed methods, one using stage 1 (Step 0 through 4) only, which can be viewed as an adaptive line projection method; the other using the additional Steps 5 and 6 as outlined in Section III. Two of the five comparing methods follow the line projection approach, namely the average method and the Matlab method [1]. Three others are from references [2], [6] and [3]. We tested all seven methods on various test images and find that the proposed methods produces the overall best visually appealing results. We list the results for five of the test images in Fig. 3. We see here that both of the methods proposed in this paper produces the overall best quality grayscale images. The striking example is the test image Mosaic 2. Here we see that the proposed methods generates the resulting pattern of 45 that is almost unnoticeable in the original color image, while all other comparing methods completely fail at doing so. We also see from the two mosaic test images that the Proposed 2 method produces results that has more contrast content than the Proposed 1 methods. This is the result from the histogram specification operation that tends to make full use of all the grayscale intensity levels. An interesting case is the Flower test image. Here the middle red flower is rendered much whiter by the proposed two methods and the method in [3]. We think this is more desired in practice, since the results produced by the other three methods tend to mislead the viewers to think that the flower and the leaves are of the same color in the original color image. In the Girl test image, we see that both the proposed methods produces the circle around the girl's eye on the left with less contrast compared with others methods. We leave this problem to future work.

Several objective quality assessment for this color to grayscale image conversion problem have also been proposed in the literature. Here we use two of them for objective performance comparison. One is the C2G_SSIM index proposed in [5] and the other is the E-score index proposed in [7]. We list the results in Table I and Table II. We see that by both measures the Proposed 1 method has

Table I
C2G_SSIM INDEX COMPARISON

Image	Scene	Mosaic 1	Flower	Girl	Mosaic 2	Ave.
Prop. 1	0.88	0.94	0.68	0.87	0.81	**0.84**
Prop. 2	0.82	0.85	0.65	0.79	0.67	0.76
Average	0.87	0.82	0.73	0.90	0.77	0.82
Matlab	0.88	0.80	0.74	0.91	0.75	0.82
[2]	0.89	0.86	0.71	0.90	0.55	0.78
[6]	0.78	0.60	0.46	0.79	0.62	0.65
[3]	0.79	0.87	0.71	0.87	0.44	0.74

Table II
E-SCORE INDEX COMPARISON

Image	Scene	Mosaic 1	Flower	Girl	Mosaic 2	Ave.
Prop. 1	0.92	0.85	0.87	0.94	0.92	**0.90**
prop. 2	0.84	0.57	0.80	0.89	0.86	0.80
Average	0.92	0.75	0.88	0.93	0.91	0.88
Matlab	0.93	0.52	0.89	0.93	0.90	0.83
[2]	0.92	0.78	0.80	0.91	0.80	0.84
[6]	0.66	0.17	0.45	0.53	0.68	0.50
[3]	0.90	0.83	0.90	0.93	0.58	0.83

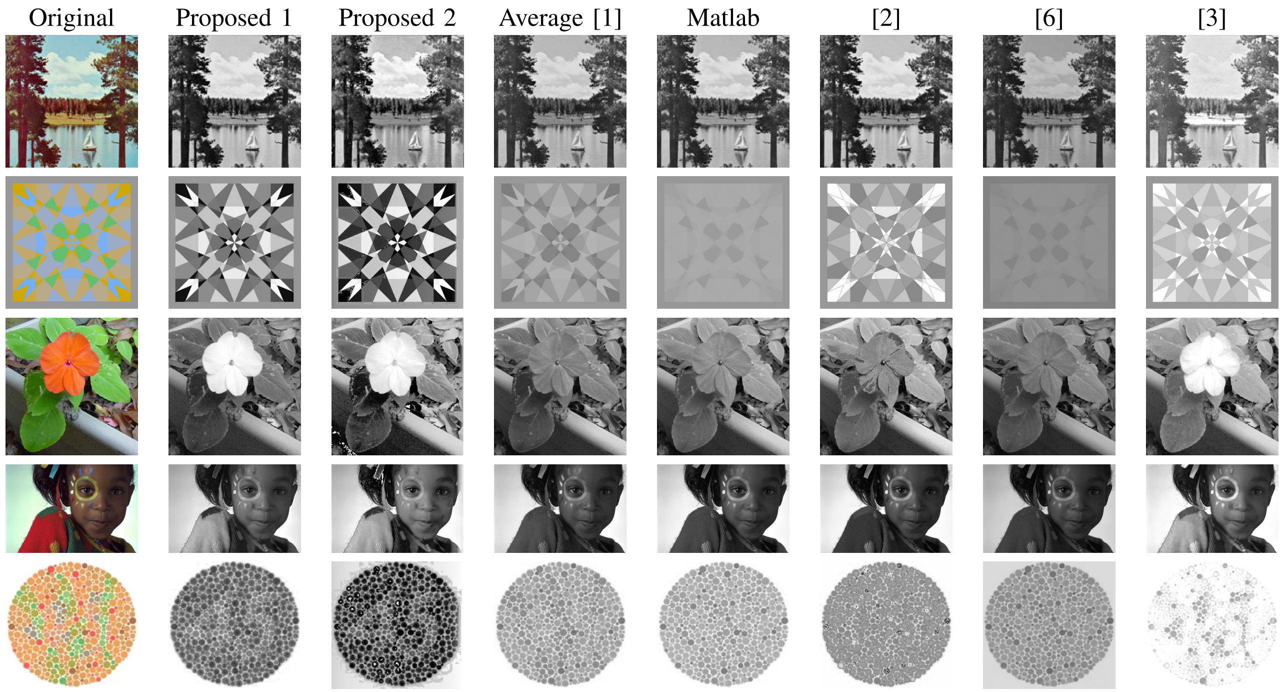

Figure 3. Visual performance comparison between the two proposed algorithms and five other methods. The third "Average" method is the line projection method when $\alpha = (1/3, 1/3, 1/3)$. The five test images used are the Scene, Mosaic 1, Flower, Girl, Mosaic 2 in sequence.

the highest average scores. This comes as a little surprise, since the Proposed 2 method seems to have no less subjective quality. After analyzing all the results, we think the objective measures may not be suitable in all situations and further research is needed to develop better ones.

V. Conclusion

In this paper we present a two-stage framework for converting a color image to a grayscale one. In the first stage we adaptively choose a direction vector and project all color pixel vectors onto this direction. The direction is chosen by maximizing the entropy of the histogram of all the projected points. This allows for better utilization of the full grayscale levels and performs better than the non-adaptive methods used in Matlab and other publications. Furthermore, in the second stage, to make full use of the grayscale levels, we present a simple histogram specification method so that all the grayscale level are evenly used by all the pixels. Experimental results confirm the enhanced performance of the proposed framework.

References

[1] C. Kanan, G. W. Cottrell, *Color-to-grayscale: Does the Method Matter in Image Recognition*, PLoS One, vol. 7, no. 1, 2012.

[2] J. W. Seo, S. D. Kim, *Novel PCA-based Color-to-gray Image Conversion*. in Proc. IEEE Int. Conf. Image Processing, pp. 2279–2283, 2013.

[3] C. Hsin, H.-N. Le, S.-J. Shin, *Color to Grayscale Transform Preserving Natural Order of Hues*, in Proc. Int. Conf. Electrical Engineering and Informatics (ICEEI), pp. 1–6, 2011.

[4] Y. Song, L. Bao, X. Xu, Q. Yang, *Decolorization: Is Rgb2gray() Out?*, ACM Siggraph Asia Technical Brief, pp. 1–4, 2013.

[5] K. Ma, T. Zhao, K. Zeng, Z. Wang, *Objective Quality Assessment for Color-to-Gray Image Conversion*, IEEE Trans. Image Processing, vol. 24, no. 12, pp. 4673–4685, 2015.

[6] R. L. de Queiroz, K. M. Braun, *Color to Gray and Back: Color Embedding into Textured Gray Images*, IEEE Trans. Image Processing, vol. 15, no. 6, pp. 1464–1470, 2006.

[7] C. Lu, L. Xu, J. Jia, *Contrast Preserving Decolorization with Perception-Based Quality Metrics*, int. J. Computer Vision, vol. 110, no. 2, pp. 222–239, 2014.

[8] Y. Wan, D. Shi, *Joint Exact Histogram Specification and Image Enhancement Through the Wavelet Transform*, IEEE Trans. Image Processing, vol. 16, no. 9, 2007.

[9] K. Jack, *Video demystified, 5th Edition*, Newnes, 2007.

[10] W. Pratt, *Digital Image Processing, 4th Edition*, Wiley-Interscience, 2007.

[11] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, Wiley-Blackwell, 2006.