Contrast Maximizing and Brightness Preserving Color to Grayscale Image Conversion

Min Qiu, School of Mathematical Sciences, South China University of Technology, Guangzhou, China
Graham D Finlayson, School of Computing Sciences, University of East Anglia, Norwich, United Kingdom
Guoping Qiu, School of Computer Science, University of Nottingham, Nottingham, United Kingdom

Abstract

This paper presents a novel variance (contrast) maximizing and brightness preserving color to grayscale image transformation method. We formulate the problem of converting color to grayscale image as a constrained optimization problem where we maximize the variance (contrast) of the grayscale image subject to the constraint that it preserves the brightness of the color image. Our algorithm has no free parameter and has a closed-form solution. We present experimental results to demonstrate the effectiveness of the new technique and compare it with other recent methods.

1. Introduction

Even though color imaging has become ubiquitous, there is still demand for high quality grayscale images in printing and in various image analysis tasks. In digital video and photography, the luminance channel or the Y component of the image is often used as the grayscale image [1]. One of the major problems for this simple approach is that in isoluminant regions, visual contrasts that are visible in the color image will become invisible in the grayscale image. That is, two objects with different chromaticities but the same brightness will become indistinguishable in the grayscale image. Ideally, the grayscale image should convey the same amount of visual information as the original color image or should preserve as much as possible the visual information contained in the color image. One way to convey the maximum information of the color image is to project the colors in a direction that has the maximum variance. This can be done using the popular principal component analysis (PCA). However, such approach may result in an image that is visually distorted. For example, sometimes the image can be inverted since an inverted image and the correct image will have the same variance. Another problem is that it may be difficult to map the dynamic range of the projections correctly since we don't know what is black and what is white. Recent methods try to produce grayscale images with visual contrasts follow that of the color contrasts [2 – 4]. These methods are problematic. First, in methods such as [2, 3], the number of variables range from hundreds to thousands which render the algorithms slow and can be trapped in local minima. Secondly, although faster solutions to these ideas have been proposed [4], these methods have many free parameters and require user intervention to produce sensible results. We should say that the default parameters of the algorithm of [4] work reasonably well.

In this paper, we propose a novel solution to the color to grayscale transformation problem. Our method finds a linear transform that converts a color image to a grayscale image in such a way that the variance of the transformation is maximized at the same time the grayscale image preserve the brightness of the color image thus avoiding situations such as the transformed image become inverted. Finding the transform coefficients is formulated as a constrained quadratic optimization problem where we maximize the grayscale image's variance subject to the constraints that the grayscale image preserve the average brightness of the original image and that the transformation is energy preserving. Our method has no free parameter and is guaranteed to converge to a global minimum. Results show that the new method produces grayscale images that not only have excellent contrasts but are also visually pleasing.

2. The Method

Let $I = \{I(x, y)\} = \{R(x, y), G(x, y), B(x, y)\}$ is a color image, we want to find three transform coefficients, α, β, and γ, to transform I into a grayscale image $L = \{L(x, y)\} = \{\alpha R(x, y) + \beta G(x, y) + \gamma B(x, y)\}$, where (x,y) is the spatial co-ordinate of the pixels. We want L to have the maximum variance in order to convey the maximum information. However, only maximizing the variance has no guarantee that L will be visually pleasing or even visually meaningful. Therefore extra constraints are necessary in order to produce visually plausible maximum variance grayscale images. There maybe various ways to set such constraints, we believe they should include the followings. First, the transform should be energy preserving. We should not amply the image's energy but rather should preserve the energy in the grayscale. What this means is that we want to have $\alpha + \beta + \gamma = 1$. Second, the grayscale image should preserve the brightness of the original image. This is an important condition. A given image and its negative will have the same variance. Preserving the brightness of the original image should remove such an ambiguity thus producing a visually correct image. The problem can therefore be formulated as

Transform I to L according to

$$L(x,y) = \alpha R(x,y) + \beta G(x,y) + \gamma B(x,y) \tag{1}$$

The transform coefficients is found as

$$(\alpha, \beta, \gamma) = \underset{\alpha,\beta,\gamma}{\arg\max}\left(\mathrm{var}\left(\{L(x,y)\}\right)\right) \tag{2}$$

Subject to

$$\alpha + \beta + \gamma = 1 \tag{3}$$

and

$$\sum_{\forall(x,y)} L(x,y) = \frac{1}{3}\sum_{\forall(x,y)}\left(R(x,y) + G(x,y) + B(x,y)\right) \tag{4}$$

Let the average of the Red, Green and Blue channels be m_R, m_G, and m_B, respectively; the mean vector of the image $m = (m_R, m_G, m_B)$; let $u = (\alpha, \beta, \gamma)$. Let $R = (R(1, 1), R(1, 2) \dots, R(M, N))$ be the $M \times N$ dimensional array of the red pixels, where $M \times N$ is the dimension of the image. We define G and B similarly for the green and blue pixels. The covariance matrix of the color channels is

$$\xi = \begin{bmatrix} R - m_R \\ G - m_G \\ B - m_B \end{bmatrix} \left[(R - m_R)^T \quad (G - m_G)^T \quad (B - m_B)^T \right] \quad (5)$$

The variance of the transformed grayscale image L, $Var(L)$, and the average brightness, $Ab(L)$, are

$$Var(L) = u\xi u^T \qquad Ab(L) = mu^T \qquad (6)$$

Without loosing generality and assuming that the average brightness of the original color image is unity. The optimization problem now becomes

Maximizing $E(\alpha, \beta, \gamma) = u\xi u^T$

Subject to $\alpha + \beta + \gamma = 1$ and $mu^T = 1$ (7)

The cost function is quadratic and the constraints are linear, the problem has a unique solution and can be solved using a number of standard methods.

3. Implementation

Solving the constrained optimization problem of (7) can be done using standard methods such as constrained optimization and quadratic programming (more detail about the implementation is given in the Appendix). Our implementation was based on Quadratic Programming (QP) and was implemented in Matlab.

4. Experimental Results

We have tested our method on a variety of images. We have also compared our results with a recent technique [4]. Basically [4] is a fast implementation of the methods described in [2, 3] with several implementation innovations.

Figures 1 – 4 show examples of converting color images into grayscale by the NTSC system, the method of [4] (using their default parameters) and our new method. It should be noted that our method has no free parameters. From these results, it is seen that ours are at least as good as those of [4] and in some case better.

In the experiments, we also observed that the technique of [4] produced more noticeable visual artifacts as compared with our technique and such an example is illustrated in Figure 5.

5. Concluding Remarks

In this paper, we have presented a novel computational method for converting color images to grayscale. Our method aims to achieve maximum contrast at the same time preserve visual correctness. Our solution is based on a constrained optimization technique and it has a closed form solution and no free parameters. We have presented experimental results which demonstrated that our new method is very effective and out performs state of a state of the art technique.

Acknowledgment

This work is supported by the EPSRC grant number EP/E012248/1.

References

[1] R. Gonzalez and R. Woods, Digital Image Processing, Prentice Hall; 2nd edition, 2002

[2] K. Rasche, R. Geist, J. Westall, Re-coloring images for gamuts of lower dimension, Computer Graphics Forum 24 (2005) 423–432

[3] A. Gooch, S. C. Olsen, J. Tumblin, B. Gooch, Color2Gray: salience-preserving color removal, ACM Trans. Graphics 24 (2005) 634–639

[4] M. Grundland, N. Dodgson, "The decolorize algorithm for contrast enhancing, color to grayscale conversion", Technical Report, No. 649, Computer Laboratory, Cambridge University, October 2005

Appendix A: Implementation Detail

A1. Constrained Optimization: The optimization problem of (7) can be solved using constrained optimization.

According to theory of constrained optimization, we can get the following formulation:

$$\begin{aligned} E &= -u\xi u' + \lambda v(mu' - 1)^2 + \lambda_1(\alpha + \beta + \gamma - 1) \\ &= (\lambda v m_1^2 - c_{11})\alpha^2 + 2(\lambda v m_1 m_2 - c_{12})\alpha\beta + 2(\lambda v m_1 m_3 - c_{13})\alpha\gamma + 2(\lambda v m_2 m_3 - c_{23})\beta\gamma + (\lambda v m_2^2 - c_{22})\beta^2 \\ &+ (\lambda v m_3^2 - c_{33})\gamma^2 - 2\lambda v m_1 \alpha - 2\lambda v m_2 \beta - 2\lambda v m_3 \gamma + \lambda_1 \alpha + \lambda_1 \beta + \lambda_1 \gamma - \lambda_1 + \lambda v \end{aligned}$$

Hence we can get:

$$\begin{cases} \frac{\partial E}{\partial \alpha} = 2(\lambda v m_1^2 - c_{11})\alpha + 2(\lambda v m_1 m_2 - c_{12})\beta + 2(\lambda v m_1 m_3 - c_{13})\gamma + \lambda_1 - 2\lambda v m_1 = 0 \\ \frac{\partial E}{\partial \beta} = 2(\lambda v m_1 m_2 - c_{12})\alpha + 2(\lambda v m_2^2 - c_{22})\beta + 2(\lambda v m_2 m_3 - c_{23})\gamma + \lambda_1 - 2\lambda v m_2 = 0 \\ \frac{\partial E}{\partial \gamma} = 2(\lambda v m_1 m_3 - c_{13})\alpha + 2(\lambda v m_2 m_3 - c_{23})\beta + 2(\lambda v m_3^2 - c_{33})\gamma + \lambda_1 - 2\lambda v m_3 = 0 \\ \alpha + \beta + \gamma - 1 = 0 \end{cases}$$

Then

$$\begin{pmatrix} 2(\lambda v m_1^2 - c_{11}) & 2(\lambda v m_1 m_2 - c_{12}) & 2(\lambda v m_1 m_3 - c_{13}) & 1 \\ 2(\lambda v m_1 m_2 - c_{12}) & 2(\lambda v m_2^2 - c_{22}) & 2(\lambda v m_2 m_3 - c_{23}) & 1 \\ 2(\lambda v m_1 m_3 - c_{13}) & 2(\lambda v m_2 m_3 - c_{23}) & 2(\lambda v m_3^2 - c_{33}) & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} 2\lambda v m_1 \\ 2\lambda v m_2 \\ 2\lambda v m_3 \\ 1 \end{pmatrix}$$

Solving the above simultaneous equations we can obtain α, β, and γ.

A2. Quadratic Programming: The optimization problem can also be solved by quadratic programming.

First of all, a quadratic programming problem has a form like the following:

$$\min_x \frac{1}{2} x'Hx + f'x \quad \text{such that} \quad \begin{aligned} & A \cdot x \le b \\ & Aeq \cdot x = beq \\ & lb \le x \le ub \end{aligned}$$

So, we change the formulation E into the quadratic programming problem:

$$\begin{aligned} E &= -Var + \lambda \cdot v \cdot Energy = -u\xi u^T + \lambda v(mu^T - 1)^2 \\ &= (\lambda v m_1^2 - c_{11})\alpha^2 + 2(\lambda v m_1 m_2 - c_{12})\alpha\beta + 2(\lambda v m_1 m_3 - c_{13})\alpha\gamma + 2(\lambda v m_2 m_3 - c_{23})\beta\gamma + (\lambda v m_2^2 - c_{22})\beta^2 \\ &+ (\lambda v m_3^2 - c_{33})\gamma^2 - 2\lambda v m_1 \alpha - 2\lambda v m_2 \beta - 2\lambda v m_3 \gamma + \lambda v \\ &= \frac{1}{2}(\alpha \quad \beta \quad \gamma) \begin{pmatrix} 2(\lambda v m_1^2 - c_{11}) & 2(\lambda v m_1 m_2 - c_{12}) & 2(\lambda v m_1 m_2 - c_{12}) \\ 2(\lambda v m_1 m_2 - c_{12}) & 2(\lambda v m_2^2 - c_{22}) & 2(\lambda v m_2 m_3 - c_{23}) \\ 2(\lambda v m_1 m_3 - c_{13}) & 2(\lambda v m_2 m_3 - c_{23}) & 2(\lambda v m_3^2 - c_{33}) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} - (2\lambda v m_1 \quad 2\lambda v m_2 \quad 2\lambda v m_3) \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} + \lambda v \\ &= \frac{1}{2} uHu^T + f^T u + \lambda v \end{aligned}$$

$$\text{such that} \quad \begin{aligned} & \alpha + \beta + \gamma = 1, \\ & 0 \le \alpha \le 1, \\ & 0 \le \beta \le 1, \\ & 0 \le \gamma \le 1. \end{aligned}$$

Then we can use the function quadprog to solve it:

$$p = quadprog(H, f, [], [], Aeq, beq, lb, ub)$$

Where $Aeq = (1 \quad 1 \quad 1)$, $beq = (1)$, $lb = (0 \quad 0 \quad 0)$, $ub = (1 \quad 1 \quad 1)$.

Then from the vector p we get is what we want.

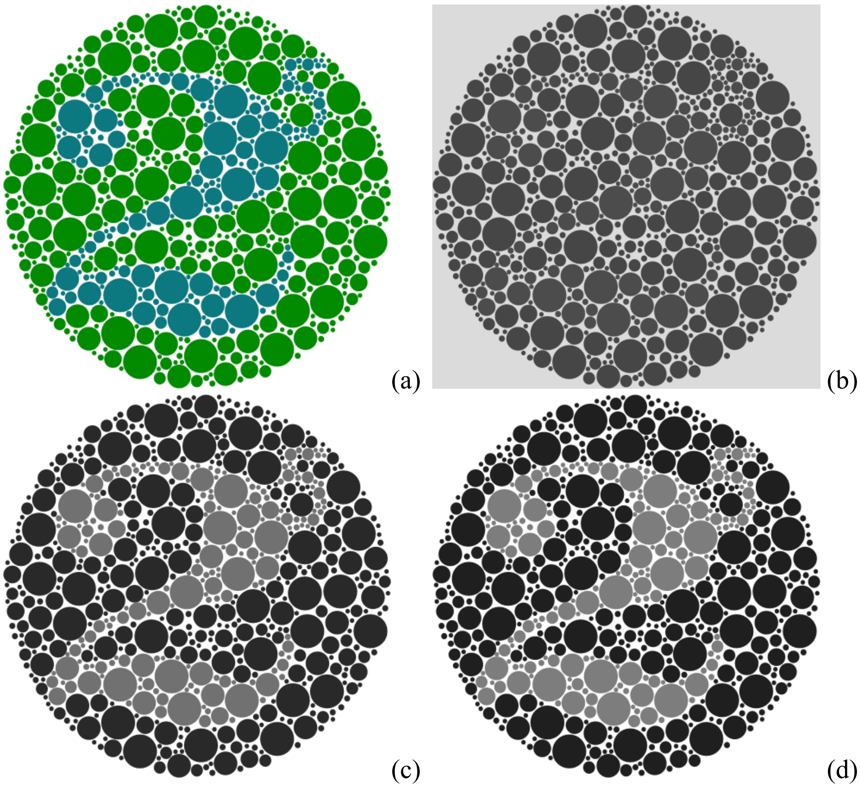

Fig. 1 (a) Original color image; (b), NTSC, (c) [4] (d) Ours

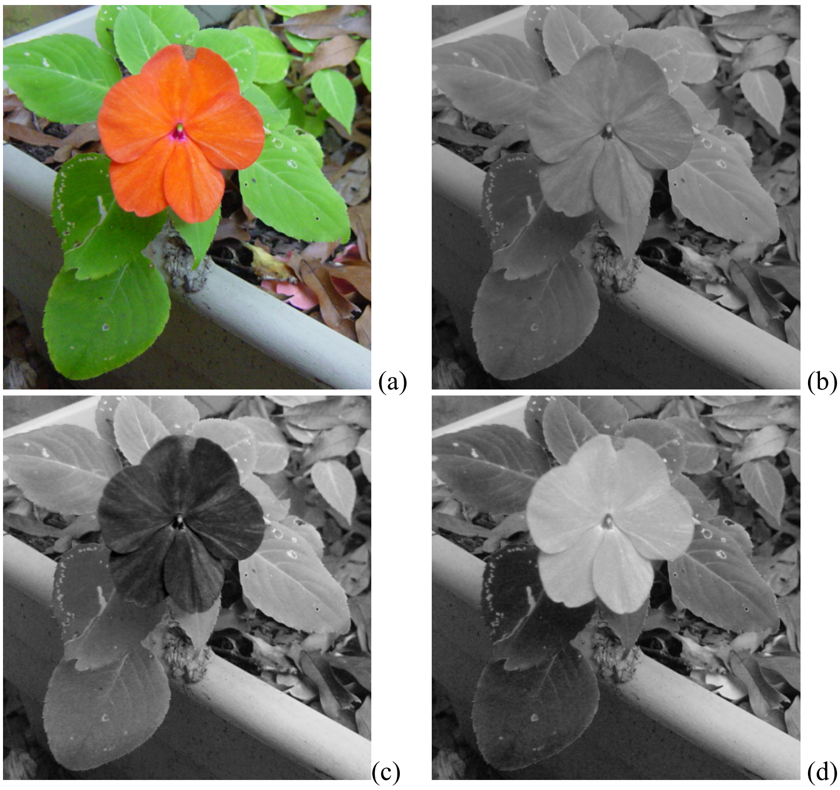

Fig. 2 (a) Original color image; (b), NTSC, (c) [4] (d) Ours

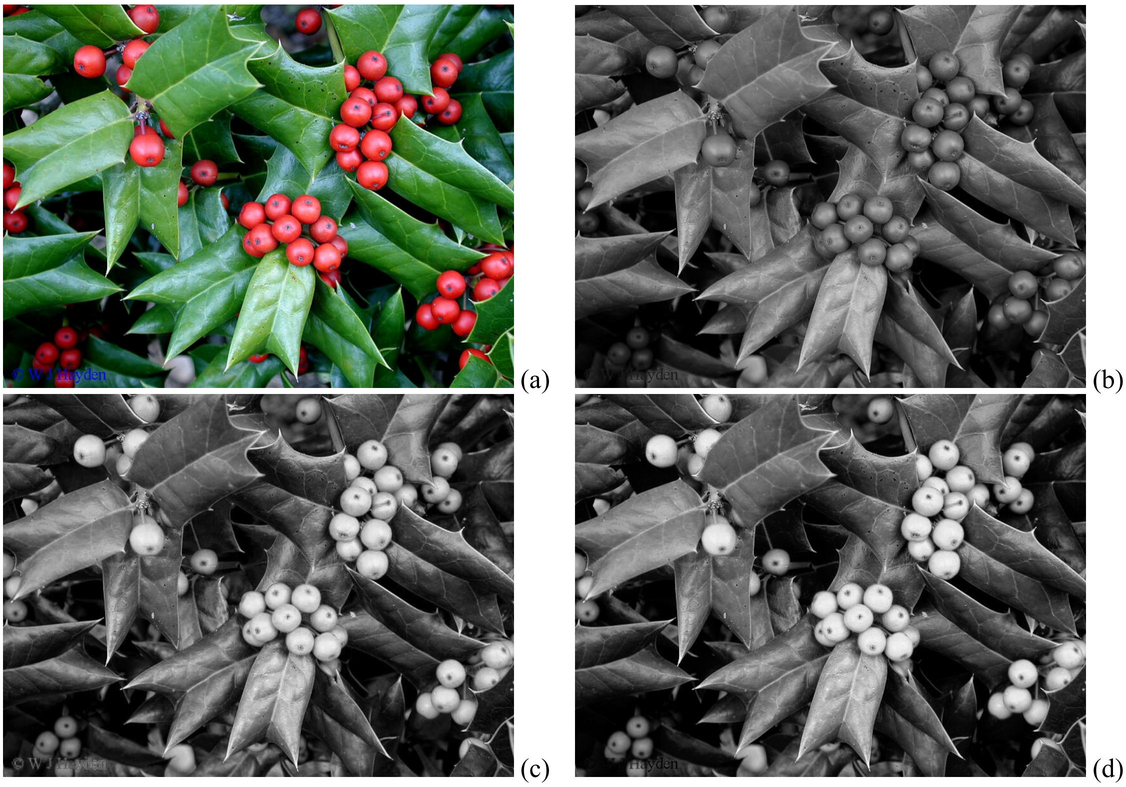

Fig. 3 (a) Original color image; (b), NTSC, (c) [4] (d) Ours

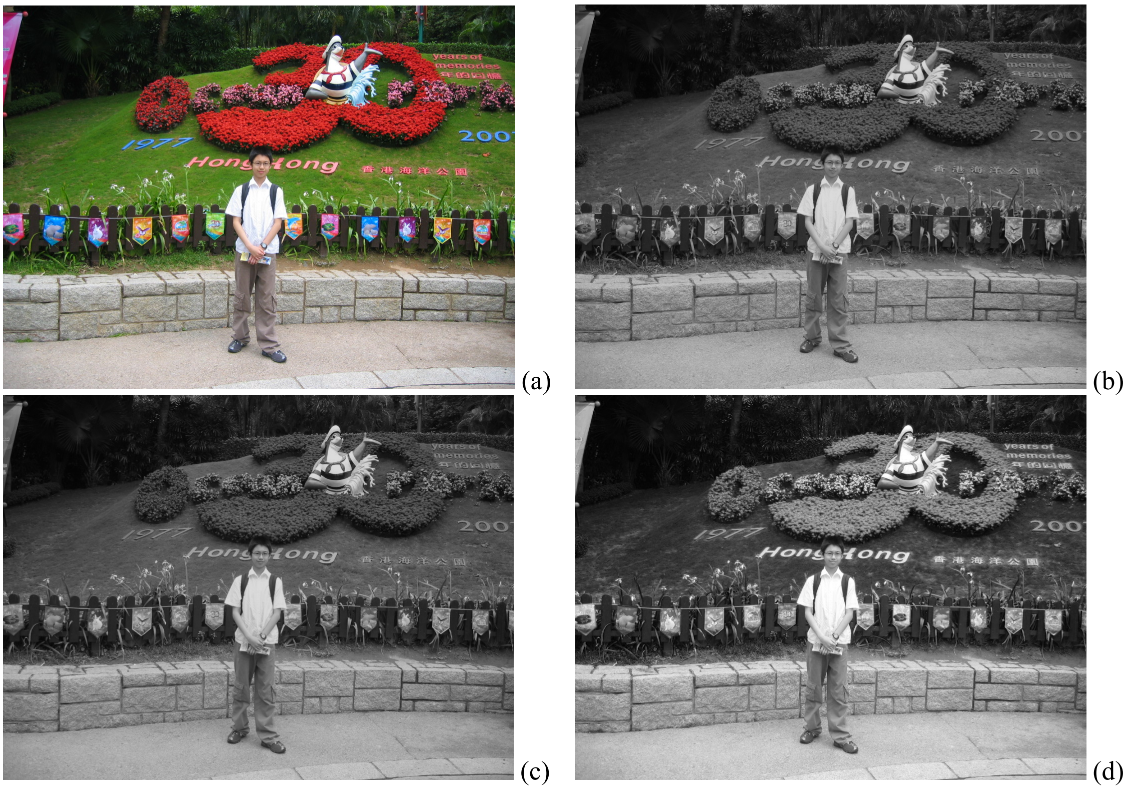

Fig. 4 (a) Original color image; (b), NTSC, (c) [4] (d) Ours

Fig. 5 (a) Original color image; (b), NTSC, (c) [4] (d) Ours. (e) a sub-image of (c), (f) a sub-image of (d)

Author Biography

Min Qiu is a Year 4 BSc student in the School of Mathematical Sciences at the South China University of Technology, Guangzhou, China. He will graduate in July 2008. Currently he is on internship working on Internet banking for HSBC Software Development (Guangdong) Ltd.

Graham D Finlayson is a Professor in the School of Computing Sciences at the University of East Anglia, Norwich, UK.

Guoping Qiu is an Associate Professor and a Reader in Visual Information Processing in the School of Computer Science at the University of Nottingham, UK.