Term Paper
Hue-preserving Color Image Enhancement Without Gamut Problem

Raunaq Malhotra(Y5370) and Utkarsh Upadhyay(Y5488)

November 10, 2008

Abstract

This paper discusses a general technique for extending the various grey-scale transform for enhancement to color images while preserving the Hue of the image [1]. The author do not talk about any particular transform, but instead provide an innovative system using which we can extend almost any grey-scale image enhancing transform to color images. Using their new algorithm, one need not even change color spaces while trying to work on an image, thus saving an enormous amount of computation. However, they also agree that they talk in length only about one way of generalizing those transforms (and generalize only Histogram equalization) while in reality there exist many such extensions. The work on finding which of these transforms would be best for which kind of image is left as an future research exercise. Also, they subsume a very important paper [2] which forms a very integral part of the conclusions that they draw. Hence, while trying to present their views on the subject, it would benefit us greatly if we dig a deeper into their assumptions and see why they are justified. A light introduction to what their assumption was is also included in this term paper.

1 Introduction

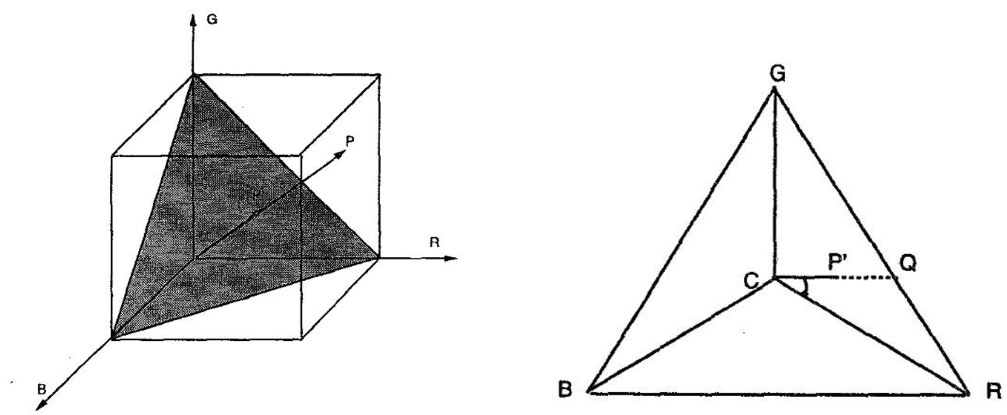
Image enhancement is used to improve the quality of an image for visual perception of human beings or for a machine for easier analysis. The set of pixel values of one image is transformed to a new set of pixel values so that the new image formed is visually pleasing and is also more suitable for analysis. We are well acquainted with the main techniques for image enhancement for grey scale images:

- Contrast stretching,
- Slicing,
- Histogram equalization, . . .

The generalization of these techniques to color images is not straight forward. Unlike grey scale images, in color images we have hue; a very important factor, which determines how we perceive colors in images. The slightest change in the hue of a point can make the picture not only from desirable to undesirable, but also from recognizable to unrecognizable. Enormous amounts of work has gone into determining hue preserving transforms for such images and most important contributions have come from surprisingly old studies. The paper which forms the base for this paper had been published in 1995 and wasn't entirely related to Hue preserving transforms [2]. Instead, it concentrated on attempting to provide changes in Luminosity without undergoing computationally heavy transforms. The interpretation that they provided happened to be such that it provided a general form of transformations which will preserve Hue. Though the original paper refrains from discussing the old paper and its transformations, in our opinion, it would be much better to touch base with the other paper and revisit some of those proofs.

2 Scaling and Shifting Operations

These operations are assumed to be hue preserving, and this section is dedicated to realising why this is so.



(a) The RGB cube with the Maxwell Triangle

(b) The Hue and Saturation

Figure 1: The Definition of Hue and Saturation

Let P be a generic color in the RGB cube as shown in 1(a). The dark triangle made with the face diagonals is known as the *Maxwell's triangle*. The frontal view of this triangle is drawn in 1(b). The point P' is the point where the $\vec{OP}$ intersects the Maxwell's plane. C is the projection of the origin on the RGB triangle.

With reference to the 1(b), we can define hue and saturation:

- **Hue**: The angle between $\vec{CP'}$ and $\vec{CR}$ is the Hue of the color.

- **Saturation**: The ratio of magnitude of $\vec{CP'}$ to $\vec{CQ}$.

Now consider the two operations separately:

- **Scaling**: If all the colours are scaled by the same value, then clearly, neither the saturation, nor the hue value of the color changes. Because the color vector $\vec{OP}$ only grows further away from the origin at exactly the same angle, it will keep intersecting the Maxwell plane at exactly the same point.

- **Shifting**: If the same value is added to all the three components, then the resultant vector will always lies in the same plane as $\vec{OP'}$ and $\vec{OC}$. This is also easy to see if we agree to the fact that upon additions on two vectors, the resultant always likes on the plane made by the two vectors. Here, one of the two vectors is parallel to $\vec{OP'}$ and the vector being added is parallel to $\vec{OC}$, since all its components have the same value. Hence, again. the value of hue will not change after this operation.

So the general form of transforms that would take one image to another image while still preserving Hue will be of the form:

$$x_k^{'} \quad = \alpha(\tilde{x}).x_k + \beta(\tilde{x}) \quad k = 1, 2, 3 \tag{1}$$

3 Linear Transform

The paper discusses a linear transform that is very simple to realize and can be easily generalised to a RGB space. For a linear transform, we would have $\alpha(\tilde{x}) = \alpha_1$ and $\beta(\tilde{x}) = \beta_1$. As for grey scale images we would like the RGB pixels to occupy the whole range available to it. Hence, as we would have done for grey scale images, we choose:

$$\alpha_1 \quad = \quad \frac{1}{\underset{k,x\in I}{max} x_k} \tag{2}$$

$$\beta_1 \quad = \quad -\underset{k,x\in I}{min} x_k \tag{3}$$

where I is our source image. This would take at least one pixel to 1 and one to 0, and to see that it is the best linear stretching possible is trivial to see.

4 Non-linear Transform

There exist a wide and useful array of non-linear transforms for grey scale image enhancement, like:

- S-type enhancement,
- Piecewise linear stretching,
- Clipping, ...

We would like to use all of these for colored images too, but this extension is non-trivial. The problem comes when we attempt to apply the transform while attempting to preserve the hue. To simplify our search space, we set $\beta(\tilde{x}) = 0$. Even then there will be an uncountable number of Hue preserving transforms that will perform the desired transform to the image. However, we restrict ourselves further and look only at functions of the form which perform linear operation of the form:

$$x_k^{'} = \alpha(\tilde{x}).x_k \quad k = 1, 2, 3 \tag{4}$$

Even now the possibilities of the functions are not tractable. Hence, now we make another restricting assumption, which is that $\alpha(\tilde{x})$ is a function of $l_x = x_1 + x_2 + x_3$.

Now the authors of the paper present a novel algorithm using which we will be able to perform the desired transform on the Image, avoid clipping of any dimension, and preserve Hue of the image too. Their algorithm is presented without any change:

4.1 The Algorithm

Initially, we define $\alpha(l_x) = \frac{f(l_x)}{l_x}$, where f is a nonlinear transformation used in contrast enhancement for grey scale images. For example, S-type transformation is listed earlier in this section. $\alpha(l_x)$, hence, may exceed 1. Now there will be such cases that value of $x_k^{'}$ may exceed 1 and thus resulting in gamut problem. A possible solution to this is to transform the color vector to CMY space and process it there. This will be dealt with in two separate cases.

$$\begin{aligned}
&\textit{Case I)} & \alpha(l_x) &\leq 1 & & (5)\\
& & x_k^{'} &= \alpha(l_x).x_k & \forall k &\in [1, 2, 3] \quad (6)\\
&\textit{Case II)} & \alpha(l_x) &\geq 1 & & (7)\\
&1) & (y_1, y_2, y_3) &= (1 - x_1, 1 - x_2, 1 - x_3) & & (8)\\
&2) & l_y &= y_1 + y_2 + y_3 & &= 3 - l_x \quad (9)\\
&3) & g(l_y) &= 3 - f(l_x) & & (10)\\
& & \alpha(y_l) &= \frac{g(l_y)}{l_y} & & (11)\\
&4) & y_k^{'} &= \alpha(l_y).y_k & \forall k &\in [1, 2, 3] \quad (12)\\
&5) & (x_1^{'}, x_2^{'}, x_3^{'}) &= (1 - y_1^{'}, 1 - y_2^{'}, 1 - y_3^{'}) & & (13)
\end{aligned}$$

The steps in case two are:

1. Transform the RGB color vector to CMY
2. Perform the transform in the CMY space
3. Revert back to the RGB space.

4.2 Histogram Equalization

Histogram equalization adjusts the histogram of the image to make it resemble to a uniform distribution. It works extremely well for contrast enhancement of gray scale images. But, performing the same operation on the RGB plane changes the hue of the pixel values of the image. So what is sometimes done is to use histogram equalisation only on the luminance/saturation plane of the image in the HSV color space. This enhances the image without affecting the hue of the pixels.

But, over enhancement of certain pixels leads to gamut problem when the image is transformed back to RGB plane. So by using the above method, we can avoid this problem.

5 Results and Conclusions

The results produced by this seemingly simple but extremely useful technique are indeed commendable.

Three images 2(a), 2(c) and 2(f) have been considered here. 2(f) is a very bright image originally. The results of our method is shown in 2(b), 2(d) and 2(g) have been shown here. The results show visible improvement in the contrast of the image. It should be noted that we have used only one of the many choices of $f(l_x)$ for the nonlinear transformation. There can be many other choices of $f(l_x)$. Sometimes, the function used by us causes over-enhancement, due to which yellow color in 2(c) is almost converted to white. In 2(b) on close examination, we see that the background has become grainy with false contours. Also, the proposed algorithm would always decrease the saturation whenever $\alpha(l_x) \geq 1$, which may not be desirable. One of the methods to avoid the above problems can be to use different function for l_x. By changing l_x to $\sqrt{\frac{x_1^2+x_2^2+x_3^2}{3}}$ from $x_1+x_2+x_3$, the result obtained for 2(c) is shown in 2(e). We can see a slight improvement in the results from 2(d). The sky color has not been completely washed out as was the case in 2(d).

We now have a general hue preserving transformation for image enhancement, but with a few constrains. For example, we have till now only used a zero value for β. The main problem in generalising this definition would be the choice of $\alpha(l_x)$ and $\beta(l_x)$. This is something that is still a matter of future research.

References

[1] Sarif Kumar Naik and C. A. Murthy. Hue-preserving color image enhancement without gamut problem. *IEEE Transactions on Image Processing*, 12(12):1591–1598, 2003.

[2] Christopher C. Yang and Jeffrey J. Rodr'iguez. Efficient luminance and saturation processing techniques for bypassing color coordinate transformations. In *Journal of Visual Communication and Image Representation*, pages 56–67, 1995.

(a) Original image (b) S-type transformation

(c) Original image (d) S-type transformation (e) Different l_x chosen

(f) Original image (g) S-type transformation

Figure 2: Images used for testing