Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, 30-31 Aug. 2008

# A FAST MULTI-SCALE RETINEX ALGORITHM FOR COLOR IMAGE ENHANCEMENT

**WEN WANG[1], BO LI [1], JIN ZHENG[1], SHU XIAN[2], JING WANG[1]**

[1]Digital Media Laboratory, School of Computer Science and Engineering, Beihang University, Beijing 100083, China
[2]School of Computer Science and Technology, North China Electric Power University, Beijing 102206, China
E-MAIL: myheaven6410@163.com, myheaven6410@gmail.com

**Abstract:**

**This paper puts forward a Fast Multi-Scale Retinex Algorithm in order to solve the problem of color distortion and to improve the disadvantages of a slowly time-consuming arithmetic in the course of image enhancement which is based on the multi-scale retinex algorithm. Through construction of average value templates in advance, Gaussian convolution operation is simplified to be average arithmetic. A new rational color space is constructed which possesses the advantages of simple space conversion and small quantity of calculation. Meanwhile it keeps the color image in a better way. The experimental results demonstrate that the Fast Multi-Scale Retinex Algorithm can ensure the treated effect and improve the efficiency of execution; furthermore it can run in the personal computers and the embedded environment. Therefore it has much better practical value.**

**Keywords:**

**Retinex theory; color image enhancement; color space; multi-scale Retinex**

## 1. Introduction

As one of the effective methods to ameliorate the quality of image, color image enhancement can make the image more vivid and more colorful while enhancing the image details and improving the quality of image. It will not lead to the distortion of image's color so that it is better to make human observation and later-period machine treatment. At present, color image enhancement mainly has two tasks: research of gray image enhancement algorithm and color information processing of color space.

The conventional enhancement algorithm mostly includes non-linear conversion techniques, histogram equalization techniques, and homomorphic filtering techniques and so on. These algorithms have limitation in practice, whose effects are not satisfactory. In recent years, with the development of color constancy theory, enhancement algorithm based on the Retinex theory has reached a better effect in dealing with the problem arose from the uneven illumination. It has currently become the hot spot of research.

Color constancy theory refers to people's stably psychological inclination when they sense the surface of the objects under the circumstance of illumination change [2]. After development, nowadays the most applied algorithm is single-scale Retinex algorithm (SSR) and multi-scale Retinex algorithm (MSR) [4], [5], [9]. They are realized in terms of center/surround function. But the convolution algorithm wastes a lot of time, so that it is hard to apply it in the high real-time video enhancement. Moreover, these two algorithms with large calculation relatively process the three components—*R, G*, *B* in the RGB color space. They may also cause the distortion of color when the original image is not following with the "gray world assumption" [3], [9].

In the view of the above problems, this paper based on MSR algorithm proposes a fast multi-scale Retinex (FMSR) color image enhancement algorithm. Through several pre-constructed average templates, Gaussian convoluted operation is selectively simplified to be the approximate average operation which can convert floating-point multiplication operation into the operation of addition and shift. Through template remove skill to reduce the amount of repeated calculation, the look-up list is established to simplify logarithmic algorithm and floating-point division algorithm. A new rational color space is constructed which possesses the advantages of simple space conversion and small quantity of calculation. Meanwhile it keeps the color image in a better way. The experimental results demonstrate that the fast multi-scale Retinex algorithm can ensure the treated effect and improve the efficiency of implementation; furthermore it can run in the personal computers and the embedded environment.

## 2. MSR algorithm

Color constancy theory is proposed and developed into

**978-1-4244-2239-5/08/$25.00 ©2008 IEEE**

Retinex theory by E. Land according to the perception model of luminance and color constructed by human Retina and Cortex [8]. Jobson and other people bring forward a center/surround Retinex algorithm by the above theory, including single-scale Retinex algorithm (SSR) and multi-scale Retinex algorithm (MSR)[4], [5], [9].

The form of SSR algorithm is:

$$R_i(x,y) = \log I_i(x,y) - \log(I_i(x,y) * F(x,y)) \quad (1)$$

In this formula, $R_i(x,y)$ is the Retinex output; the subscript $i$ represents the three different color channels— $R, G, B$; $I_i(x,y)$ is the $i$-th image of spectrum zone; "*"denotes the convolution operation; $F(x,y)$ refers to normalized surround Gauss function, whose form is

$$F(x,y) = Ke^{-r^2/c^2}, \quad r = \sqrt{x^2 + y^2} \quad (2)$$

in which $K$ is a normalized parameter, so that $\iint F(x,y)dxdy = 1$, $c$ is the gauss parameter which should be chosen adaptable.

MSR algorithm is counted by the sum of many different SSR algorithms, whose expression is

$$R_{MSR_i} = \sum_{n=1}^{N} \omega_n R_{n_i} \quad (3)$$

In this formula, $N$ is the number of scale; $R_{n_i}$ is result of the $i$-th spectrum zone of the $n$-th scale; $R_{MSR_i}$ is the result of MSR algorithm for the $i$-th spectrum zone; $\omega_n$ refers to the weighting factor for the $n$-th scale. The difference between $R_n(x,y)$ and $R(x,y)$ is the different surround function. The specific formula is as follows:

$$F_n(x,y) = Ke^{-r^2/c_n^2} \quad (4)$$

The difference of parameter $c_n$ selection in the Gaussian function will have a different impact on the effect. The smaller the value of $c_n$ is, the better contrast and edge is gotten; and the larger the value of $c_n$ is, the better color is gotten. The document [1] points out that the value of small scale $c_n$ occupies 1%~5% in the entire image, the value of middle scale $c_n$ occupies 10%~15% and the value of large scale $c_n$ occupies the 30%~50%.

## 3. Fast color image enhancement algorithm

### 3.1. Optimization of MSR algorithm

The most time-consuming part of MSR algorithm appears in the convolution operation, which is mentioned in the previous section— $I_i(x,y) * F_n(x,y)$. Center/Surround function means Gaussian function— $F_n(x,y) = Ke^{-r^2/c_n^2}$, that is, the closer the point away the center is, the greater the impact on the center is; vice versa.

The convolution algorithm in the spatial domain can be transformed into multiplication in frequency domain. The premise of this transformation is that the image and template must be in the same size, as well as the size of image must be a power of two in the process of fast Fourier transformation. These two premises have the problem of edge extensions. The size of the edge extension and the numerical value of extension will influence the calculation results and the efficiency of the algorithm ,and increase the amount of calculation. Therefore it is necessary to consider another point of view to make it optimization.

Convolution mentioned above corresponds to a weighted average, whose substance is a low-pass filter, and center/surround function is similar with low-pass filter template in spatial domain. After the experiments, when the parameter of the center surround function and the size of template split the difference, the numerical values in Gaussian template are all approximate equal. For example, when the size of the selected template is 129×129, c=200, then the Gaussian template is as follows:

$$\begin{bmatrix} 0.5614 & 0.5623 & \cdots & 0.5829 & \cdots & \cdots & \cdots & 0.5829 & \cdots & 0.5623 & 0.5614 \\ 0.5623 & 0.5632 & \cdots & 0.5839 & \cdots & \cdots & \cdots & 0.5839 & \cdots & 0.5632 & 0.5623 \\ \cdots & \cdots & & \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots \\ 0.5742 & 0.5751 & \cdots & 0.6093 & \cdots & \cdots & \cdots & 0.6093 & \cdots & 0.5751 & 0.5742 \\ \cdots & \cdots & & \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots \\ 0.5909 & 0.5919 & \cdots & 0.6171 & \cdots & 0.6219 & \cdots & 0.6171 & \cdots & 0.5919 & 0.5909 \\ \cdots & \cdots & & \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots \\ 0.5742 & 0.5751 & \cdots & 0.6093 & \cdots & \cdots & \cdots & 0.6093 & \cdots & 0.5751 & 0.5742 \\ \cdots & \cdots & & \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots \\ 0.5623 & 0.5632 & \cdots & 0.5839 & \cdots & \cdots & \cdots & 0.5839 & \cdots & 0.5632 & 0.5623 \\ 0.5614 & 0.5623 & \cdots & 0.5829 & \cdots & \cdots & \cdots & 0.5829 & \cdots & 0.5623 & 0.5614 \end{bmatrix} \times 10^{-4} \quad (5)$$

The formula (5) points out that it needs 16641 floating-point multiplication calculations and 16641 addition calculations to get the convolution value of every pixel point. For the video standard of 704×576 appearing frequently in video processing, it needs $6.74 \times 10^9$ floating-point multiplications. Then it is difficult to realize such a large number of algorithm in the ordinary computer's CPU. However, the values in Gaussian templates have smaller differences, in which the basic values tend to $0.6094 \times 10^{-4}$ . So if the average template replaces the Gaussian template, the amount of calculation is greatly reduced, meanwhile $0.6094 \times 10^{-4} \approx 1/(128)^2$, The division operation after summation can be simplified to be shift operation, as $(128)^2=2^{14}$, in which $1/(128)^2$ means to shift 14 bits to right. The time required in the shift operation is less than division operation, which lays the base for the further fast algorithm.

Through a lot of experiments based on this point, it is shown that when the size of the template is suitable to the selected parameter of Gaussian convolution, there are not too many differences between the results processed by average filtering and the results processed by Gaussian filtering. Moreover the processing results will not get much influence. After repeated comparison and contrast and a lot of experiments, this paper will use average filtering operation for three times and identifies the following parameters:

(1) when the selected small-sized template is $33\times33$，$c=30$, every value in Gaussian template tends to be $9.2\times10^{-4}$;

(2) when the selected middle-sized template is $65\times65$，$c=90$, every value in Gaussian template tends to be $0.2367\times10^{-4}\approx1/(64)^2$, besides while calculation it should shift 12 bits to right;

(3) when the selected large-sized template is $129\times129$，$c=200$, every value in Gaussian template tends to be $0.6094\times10^{-4}\approx1/(128)^2$, besides while calculation it should shift 14 bits to right.

After receiving the summation of template in the image, the most part of the calculation between the two adjacent points is repeated in the process of calculation. Figure 1 shows that the calculated difference between those two adjacent points is that the template moves only one row or one column. So in terms of that, the summation of every point in templates can be stored by an array, the movement of the adjacent points can be realized by addition or subtraction of the data of related lines. Furthermore, the logarithm operation related in the algorithm can be achieved by establishing look-up lists so that makes the algorithm much simpler and quicker in a large scope.

Every pixel point in the original algorithm requires 16641 floating-point multiplication calculations and 16641 addition calculations, but after the simplified calculation, every pixel only needs less than 16641 addition calculations and one shift calculation. For the D1 format (image resolution of $704\times576$) commonly used in video processing, it may reduce at least $6.74\times10^9$ floating-point multiplication calculation, which greatly decreases the amount of calculation and improves the efficiency of calculation.

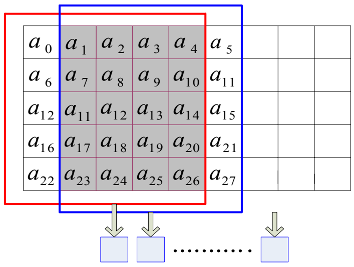



Figure 1. Computing Distinction Between templates

### 3.2. Optimization of color space conversion

RGB space is the basic color space in computer, but the three primary colors have a strong correlation. The separate disposal of the different color may result in the distortion of the entire color, so it needs relevant methods to prevent the distortion of color. MSR algorithm introduced in the second section is the method to dispose in the three color channels of RGB color space, by which the amount of calculation is huge. Moreover, Hurbert ever pointed out Retinex actually conceal a kind of assumption of “gray world”, because SSR and MSR algorithm can operate simultaneously in the three RGB channels, and only when the surrounding values in the three channels are essentially equal, the color information of the original image will not be destroyed. If there are great differences among the surrounding values of the three channels, SSR and MSR will not only eliminate the impact of illumination intensity, but also will have a color distortion [3], [9].

Besides RGB space, the color space in common use in video image is YUV space, in which Y is the luminance; U and V are the chrominance. The enhanced processing is operated for Y, but the color information is faded in the course of disposal in which the color image is not vivid.

HSV color space is based on the psychological feeling of human being to the color; it expresses the color through H (Hue), S (Saturation), and V (Value). It is a good choice for color image because of the less correlation among the three components [5]. In the situation of not adjustment of saturation, HSV color space only has to deal with one component. Compared with RGB color space which deals with three components, it can reduce the amount of calculation and prevent effectively the color distortion. Then compared with YUV color space, it keeps a good color of image. Therefore, HSV is an ideal color space in the realization of enhanced processing, but at the same time there are also many limitations. For instance, there is no color space in video image for HSV space, so the format conversion is required here.

The conversion between RGB space and HSV space requires larger calculation and a great number of decimal fraction, which reduces the operation efficiency of procedures and leads to the difficult transplantation of algorithm and weak generalization. So it needs to optimize the algorithm.

The conversion formula (only luminance in HSV) from RGB to HSV is:

$$V = \max(R, G, B) \tag{6}$$

The processing to $V$ is the processing to the largest value of the three components of $R$, $G$ and $B$.

The conversion formula from HSV to RGB is as follows (in which $H$, $S$ and $V$ represent the three components of HSV color space and $R$, $G$ and $B$ represent the three components of RGB color space):

$$\begin{cases} R=V,\ G=c,\ B=a\ ; & if\ \ i\text{=0} \\ R=b,\ G=V,\ B=a\ ; & if\ \ i\text{=1} \\ R=a,\ G=V,\ B=c\ ; & if\ \ i\text{=2} \\ R=a,\ G=b,\ B=V\ ; & if\ \ i\text{=3} \\ R=c,\ G=a,\ B=V\ ; & if\ \ i\text{=4} \\ R=V,\ G=a,\ B=b\ ; & if\ \ i\text{=5} \end{cases} \tag{7}$$

in which：

$$\begin{cases} i = get\ \ the\ \ integer\ \ of\ \ (H/60) \\ f=(H/60)-i \\ a=V*(1-S) \\ b=V*(1-S*f) \\ c=V*(1-S*(1-f)) \end{cases} \tag{8}$$

From the above formula, it can be seen that in the conversion between HSV and RGB a large number of floating-point multiplication and division calculations are required. Therefore it is hard to accomplish the conversion from RGB to HSV then back to RGB in a short time.

But at the same time, from the formula (7), $R$, $G$ and $B$ are all composed of four components "$a$, $b$, $c$, and $V$", with different order under different circumstances in the course of the conversion from HSV to RGB. From the formula (8), "$a$, $b$, $c$, and $V$" these four components are all multiple of component $V$, that is, the processing results only for the component $V$ in HSV space are equal to the results which can be achieved by every component multiplies the same multiple value in RGB space. Moreover, the multiple value is the value of $V$ that after processing divides the one before processing. Therefore, one new rational color space $J$ is constructed in this paper. In RGB space, the maximum in every pixel of RGB components is selected to get the quotient of the results after processing and before processing as an enlarged parameter. Then every component of RGB multiplies the enlarged parameter to get the result which is equivalent to the result of $V$ in HSV space.

Now this color space conversion is applied to MSR algorithm. In RGB color space,

$$J = \max(R, G, B) \tag{9}$$

The component $J$ uses MSR algorithm to make enhance processing to get the final result $J'$. Suppose multiple parameter matrix is $r$ , so

$$r = J^{'} / J \tag{10}$$

The three components in RGB space are expanded with the relative multiple parameter to get the results of these three components $R$, $G$ and $B$, see (11)

$$\begin{cases} R^{'} = R * r \\ G^{'} = G * r \\ B^{'} = B * r \end{cases} \tag{11}$$

Concerning the division operation of parameter $r$, the look-up table can be made up to solve the problem. The values in the table are those with whose denominator is 1~255, that is, $\{ \frac{1}{1}, \frac{1}{2}, \frac{1}{3}, ..., \frac{1}{254}, \frac{1}{255} \}$. In this way, the floating-point division can be modified as checking-list operation and floating-point multiplication operation.

When the color image needs to be dealt with but the saturation needs not to be adjusted in HSV space, it is better to employ the new set of color space $J$ constructed in this paper. The processing effects are the same to those in HSV space. And the space $J$ has the advantages of simple conversion, less calculation and high efficiency.

In summary, this paper offers the flow chart of fast multi-scale Retinex for color image enhancement algorithm, just as shown in figure 2:

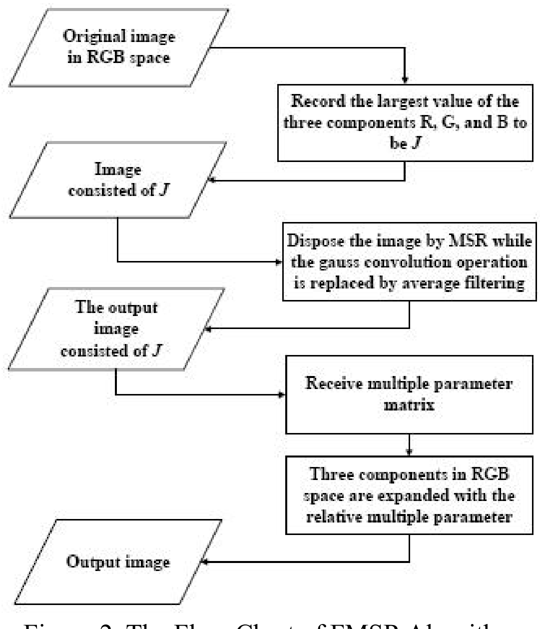
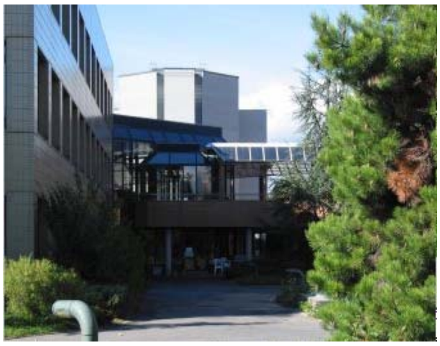


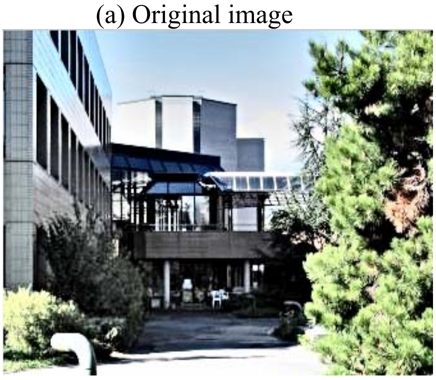

Figure 2. The Flow Chart of FMSR Algorithm

## 4. Experimental results and analysis

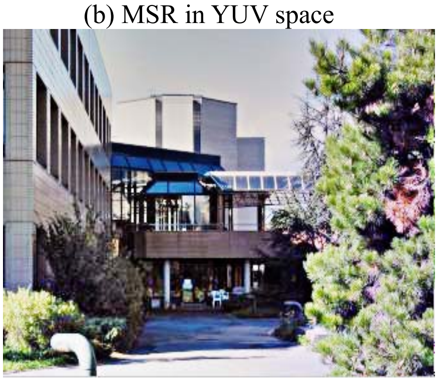
In order to prove the efficiency of the algorithm in this paper, the experiment on a larger number of pictures is made in terms of MSR in YUV space, MSR in RGB space, as well as MSR in HSV space, and FMSR algorithm in this paper. The experimental results are shown in figure 3.



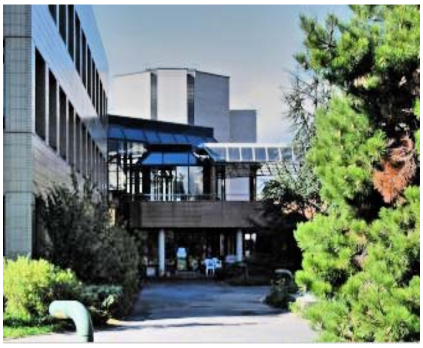
(a) Original image



(b) MSR in YUV space



(c) MSR in RGB space



(d) MSR in HSV space

(e) FMSR in this paper

Figure 3. Experimental Result

When MSR algorithm is used in YUV color space (b), the effect of image is clear after processing, but the color is faded. The color information of trees and roof in the original picture is lost so that the whole image presents gray tone. Employing the same method in RGB color space (c), the image becomes very clear and gets a better visual effect. But because the original picture does not fully conform to the "gray world assumption", the color is distorted, which can be seen from the images of trees and sky. However in the HSV color space (d), the image becomes clear, shining and bright-colored, which is suitable for people to observe. Furthermore, the FMSR algorithm (e) preserves the processing effect of the original algorithm and makes the picture clear and colorful.

In order to verify the executable efficiency of algorithm, the test is based on the image resolution of 704 × 576, takes the personal computer as experimental platform whose CPU is INTEL 2.4 G and memory is 512 M, and takes DM642 DSP-based embedded platform, with the operation time is as follows:

Table1. Comparison of Executing Time

| | MSR on PC | FMSR on PC | MSR on DM642 | FMSR on DM642 |
|---|---|---|---|---|
| TIME(ms) | 896 | 38 | 726 | 26 |

The program on computer after optimization needs 38 ms, and the processing effect is increased by 24 times; however, the optimization of DM642 on DSP only needs 26 ms, the processing effect is increased by 28 times. Using FMSR algorithm in this paper not only gets a better image enhanced effect, but also greatly reduces the executing time. The processing of image can apply to real-time video disposal system with fine practical meaning.

## 5. Conclusion

Because MSR color image enhancement algorithm may cause large calculation and color distortion in the course of its realization, this paper proposes a kind of fast multi-scale Retinex algorithm (FMSR) for color image enhancement. It improves the problem of large calculation in the convolution part of MSR algorithm. Through the construction of several average value templates in advance, Gaussian convolution operation is simplified to be average arithmetic, and floating-point multiplication operation is converted to addition and shift operation. Then through template shift skill, the repeated calculation is reduced. A new rational color space is constructed which possesses the advantages of simple space conversion and small quantity of calculation; meanwhile it keeps the color image in a better way. The experimental results demonstrate that the fast multi-scale Retinex algorithm in this paper can ensure the treated effect and improve the efficiency of execution; furthermore it can run in the personal computers and the embedded environment, thereby expanding the applying scope of algorithm.

## Acknowledgements

Supported by the NSFC (60775018), the Program for New Century Excellent Talents in University and 863 Program. The research was made in the State Key Laboratory of Virtual Reality Technologies.

## References

[1] Li Tao and Vijayan Asari. Modified Luminance Based MSR for Fast and Efficient Image Enhancement. In: Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop, 2003
[2] Andrew Moore, John Allman, and Rodney M.Goodman. A real-time neural system for color constancy. IEEE Transactions on Neural Networks, 1991, V2(2):237-247
[3] D. J. Jobson, Z. Rahman, and G. A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Transaction on Image Processing, vol. 6, no. 7, pp. 965-976, July 1997.
[4] Kasson J M, Plouffe W. An analysis of selected computer interchange color space. Association for Computing Machinery Transactions on Graphics, 1992, 11(2): 33~34
[5] Land, E. H. An Alternative Technique for the Computation for the Designator in the Retinex Theory of Color Vision. National Academy of Science, vol.83, pp.3078-3080, 1986.
[6] T.J.Brown. An Adaptive Strategy for Wavelet Based Image Enhancement. Proceeding of Irish Machine Vision and Image Proceeding Conference, Belfast, 2000.
[7] Zong X, Laine AF, Geiser EA, et al. Denoising and contrast enhancement via wavelet shrinkage and non2linear adaptive gain. Wavelet A pplications 3: Proceedings of SPIE. 1996: 566 -574
[8] Edwin H Land. The Retinex Theory of Color Vision. Scientific American, 1977, 237:1082 128.
[9] D. J. Jobson, Z. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. IEEE Transaction on Image Processing, vol. 6, no. 3, pp. 451-462, Mar. 1997.