Analysis and Improvement of Multi-Scale Retinex

Kobus Barnard and Brian Funt
Simon Fraser University
Burnaby, BC, Canada

Abstract

The main thrust of this paper is to modify the multi-scale retinex (MSR) approach to image enhancement so that the processing is more justified from a theoretical standpoint. This leads to a new algorithm with fewer arbitrary parameters that is more flexible, maintains color fidelity, and still preserves the contrast-enhancement benefits of the original MSR method. To accomplish this we identify the explicit and implicit processing goals of MSR. By decoupling the MSR operations from one another, we build an algorithm composed of independent steps that separates out the issues of gamma adjustment, color balance, dynamic range compression, and color enhancement, which are all jumbled together in the original MSR method. We then extend MSR with color constancy and chromaticity-preserving contrast enhancement.

Introduction

Recent work [1,2,3,4] advocates MSR as a method of image enhancement which provides color constancy and dynamic range compression. Nonetheless, there are a number of problems with the original MSR method. The chief conceptual problem is that a number of image-processing tasks are performed simultaneously without sufficient regard to the interactions occurring between them. The main practical consequence of this is that MSR is not appropriate for applications which are sensitive to color.

MSR serves a subset of the following five image processing goals, depending on the circumstances:

1) Compensating for uncalibrated devices (gamma correction)
2) Color constancy processing
3) Local dynamic range compression
4) Global dynamic range compression
5) Color enhancement

In the original MSR method all the processing steps are intertwined, and as a result, the colors are changed in image dependent and unpredictable ways. We will disentangle these tasks, and develop a sound theoretical basis for them. In addition, when the fifth task is appropriate, it is our view that this processing should proceed relative to a well defined color baseline, and such a baseline is provided by our approach. To do this we use an effective, neural-network-based, color constancy algorithm [5] to correct for mismatches between the illuminant and the camera balance. Then we use a modified version of MSR which does not change the color. This results in an appropriate, well defined, baseline for subsequent color enhancement if required by the application.

Overview of the original MSR method

MSR is explained easily from single-scale Retinex [2, 3, 4]. For SSR we have:

$$R_i(x,y,c) = \log\{I_i(x,y)\} - \log\{F(x,y,c) \otimes I_i(x,y)\} \quad (1)$$

where $R_i(x,y,c)$ is the output for channel "i", $I_i(x,y)$ is the image value for channel "i", $\otimes$ denotes convolution, and $F(x,y,c)$ is a Gaussian surround function explicitly given by:

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

with K selected so that:

$$\iint F(x,y,c)\,dx\,dy = 1 \quad (3)$$

In the above, the constant "c" is the scale. The MSR output is simply the weighted sum of several SSR's with different scales:

$$R_{M_i}(x,y,\mathbf{w},\mathbf{c}) = \sum_{n=1}^{N} w_n R_i(x,y,c_n) \quad (4)$$

where $R_{M_i}(x,y)$ is the MSR result for channel "i", $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ where w_n is the weight of the n'th SSR, $\mathbf{c} = (c_1, c_2, \ldots, c_N)$, where c_n is the scale of the n'th SSR, and we insist that $\sum_{n=1}^{N} w_n = 1$. In [2] the authors state that the choice of scales is application dependent, but that for most applications at least three scales are required, and that equal weighting is usually adequate. The example illustrated in Figure 1 of [2] uses scales of 15, 80, and 250 pixels, which is also the set used in [4].

The result of the above processing will have both negative and positive RGB values, and the histogram will typically have large tails. Thus a final gain-offset is applied as mentioned in [3] and discussed in more detail below.

This processing can cause image colors to go towards gray, and thus an additional processing step is proposed in [1]:

$$R'_{M_i}(x,y,w,c,C) = R_{M_i}(x,y,w,c) * I'_i(x,y,C) \quad (5)$$

where $I'_i(x,y,C)$ is given by:

$$I'_i(x,y,C) = \log\left(1 + C\frac{I_i(x,y)}{\sum_{i=1}^{3} I_i(x,y)}\right) \quad (6)$$

where we have taken the liberty to use log(1+x) in place of log(x) to ensure a positive result. In [4] a value of 125 is suggested for C; for [6] we empirically settled on a value of 100 for a specific test image. The difference between using these two values is small. In formula (5) of [4] a second constant is used which is simply a multiplier of the result: $I''_i(x,y,C,b) = \beta I'_i(x,y,C)$. However, in our implementation this constant is absorbed in the final gain-offset step.

A few more words about the final gain-offset step are warranted. Figure 8 of [3] shows how clipping is needed to have good contrast, as the resultant image histogram has quite large tails. We assume that the goal of consistently removing these tails led to the pair of gain-offset constants recently published in [4][1]. However, our experience prior to that publication did not leave us confident that the best choice is particularly image independent, especially for images from various sources. Thus we settled on implementing what we believe is the intent of the adjustment, and set the clipping point based on clipping a few percent of the pixels on either side. The extrema of the clipping points from each of the three channels is used, in order to apply the same gain-offset to all three. We have also explored clipping a fixed percentage of the range, as well as looking for image independent gain offset parameters, as suggested in [4]. All methods work to some extent, but we do not have an appropriate metric for performance, and additional work is required in this area.

Unfortunately, for the purposes of comparison, the methods and constants used for the gain-offset can substantially affect image appearance. This is especially the case where the part of the range which is of most interest has been compressed with a logarithm operation. In addition, the resulting color is sensitive to the gain-offset adjustment before color correction. Formula (6) of [4] seems to imply that none is needed, but in our implementation it is necessary to get a reasonable result. Furthermore, the color is sensitive to the associated constants.

[1]We have not yet been able to obtain reasonable results using the formula and constants in [4].

MSR and Color fidelity

To preserve image chromaticity while doing dynamic range compression, we must begin with calibrated input and output devices. In particular, we require a linear relationship between scene radiance and CRT luminance of the three channels (up to a uniformly applied multiplicative nonlinear function). This means that the CRT's gamma must be taken into account. The standard power-law method of gamma correction is only an approximation, and the best value of "gamma" varies from monitor to monitor. As a result, we have found it worthwhile to calibrate our monitor with a spectra-radiometer.

There is an interesting relationship with the original formulation of MSR and gamma correction. That method uses a channel-independent logarithm, which normally would have the side effect of changing the image colors. However, the operation somewhat approximates monitor gamma correction, and thus the color shift is far less of a problem when displaying the result without gamma correction than would be expected. In fact, applying gamma correction to the result of MSR processing normally gives poor results. Specifically the images look washed out and over gamma corrected. The problem with just accepting and using this coincidence as a conveniently provided gamma correction is that device calibration (gamma correction) is meant to compensate for devices, but now one is committed to a single method, and thus the result is device dependent. Regardless, since MSR can to some extent play the role of gamma correction, it is important to ensure proper gamma correction is being applied to the original image when being compared to MSR results on a monitor.

Device calibration is also an issue on input. It is interesting to note that given a histogram or range based gain-offset adjustment as described above, the original MSR method can be quite resilient to insufficient information about whether a gamma correction has been applied to the input. The gamma becomes roughly a constant factor due to the channel independent logarithm (although not exactly, due to the convolution). This constant is then essentially absorbed by the histogram based gain-offset adjustment. For confirmation, we have verified that there is little visual difference between MSR output of gamma corrected and non gamma corrected input.

The second color problem with the original MSR approach is that the image colors tend to be desaturated grayish. This is due to the manner in which gray-world-based color constancy processing is applied to relatively small image neighbourhoods. Each pixel's color is compared to the average of the colors in a surrounding neighbourhood. For regions of constant color this means that the MSR result will tend towards gray regardless of the color of the region.

Later versions of MSR include a processing step which puts back some of the color that was removed. The intermediate image colors are modified by a non-linear function of the original image colors. This processing has the obvious problem of changing the image colors in ways that are hard to characterize and predict. A second problem with the color correction step is that it seems to defeat the color constancy processing goal of MSR. A gray wall under blue light, as seen by a camera balanced for a redder light, will be too blue. MSR without color correction will move the color of the wall towards gray, and thus achieve some degree of color constancy. However, if the color correction step is now used, the color of the wall will be moved back towards blue!

Another color problem with standard MSR processing is complement color bleeding at certain color edges due to the local contrast enhancement. Consider a white card mounted on a yellow background. For simplicity, consider that the red and the green channels of the yellow are similar to that of white, and the blue is substantially smaller. Then only the blue channel will change due to the boundary, and the blue channel of the white near the boundary will be enhanced relative to the others which represent neutral. Hence the white card will have a blue halo near the boundary.

Color preserving MSR

We now outline an alternative approach to MSR. As mentioned earlier, the main idea is to separate the processing goals/effects of MSR so that each one can be done more optimally. First we ensure that the input is linear. Then we optionally apply color constancy processing followed by MSR style processing to an appropriately defined image luminance. The processing here can take many forms, of which two are discussed in detail below. The RGB of the output image pixels are then set so that their chromaticity is the same as in the (possibly color corrected) linear input image, but their luminances are the result of the previous processing step. At this point color enhancement, such as increasing the color saturation, can be applied. Finally, the image is mapped into the appropriate space to give linear output on the target device. In the case of a CRT monitor, this can be approximated by a gamma correction. We now provide some additional details.

As discussed above, color fidelity is best achieved if the input is proportional to scene radiance. Thus we attempt to linearize the input if this is not the case. We have experimented with input from a Sony CCD camera as well as Kodak photo CD images. In the case of the camera, we have verified that it is linear. We linearize photo CD images by inverting the algebra described in [7]. It is not known how well this corresponds to the radiance in the original scenes, but for the purposes of experimentation we assume it is linear.

If color constancy is an issue for the application, it is dealt with next. For the purpose of this paper, we define color constancy processing as a correction for a mismatch between the illuminant chromatically for which the imaging system is calibrated and the actual illuminant chromaticity of the scene. Color correction so defined is different than simply determining an illuminant independent description of the scene. Most methods available to do this correction implicitly assume that the input is linear, and thus a good result is dependent on the linearity considerations discussed above. In fact, using the above definition for color constancy processing almost demands reference to a linear space.

Standard MSR has its roots in the latest color constancy work by Land [8,9,10], and color constancy processing is one of the purported goals of MSR processing. However, the color constancy processing inherent in standard MSR processing has several weaknesses. First, it attempts to do color correction in a non-linear space. Second, it essentially is based on the gray world assumption, which is not a major problem, except that there are better algorithms available (see, for example, [5,11,12,13]). A more serious problem is that the implementation of the gray world algorithm is not optimal. Color constancy algorithms generally make some assumption about how the illuminant chromaticity varies spatially (the most common assumption being that it is uniform), and then exploit that assumption. In the case of MSR, the use of a large scale implies some confidence that the illumination uniformity is wide, but the use of smaller scales yields poor color constancy results due to local violations of the gray world assumption, and leads to a grayed out image. Averaging the results mitigates the errors, but also reduces the chances for good performance, and thus is unsatisfactory. We posit that if illumination uniformity is an issue, it should be dealt with explicitly in the algorithm (as is done in [14]). Otherwise, the illumination chromaticity should be assumed constant, as this gives the most effective color constancy processing.

The color constancy algorithm used for our experiments is a neural network trained to predict the chromaticity of the scene illuminant [5]. This is then used to compute an estimate of what the scene would look like, had it been illuminated by an appropriate illuminant for the imaging system. The performance of this algorithm is significantly better than gray world based methods.

The next step is to apply MSR style processing on an appropriately defined expression of the image luminance. We offer two methods to do this. The first method is simpler and changes the image less, and may be preferable for images from sources known to have small dynamic range. The second method is designed to approximate the dynamic range compression of the original MSR method. The significance of the second method is that it is more appropriate on images with high dynamic range. In order to investigate the relationship of the various methods and input dynamic range we created some images with extended dynamic range by either combining a number of images taken at different apertures, or averaging a large number of images.

For the first method we apply MSR style processing *without* taking logarithms on the image luminance defined by $I_I = \sum I_i$ (in the case of three-channels $I_I = I_{red} + I_{green} + I_{blue}$) as follows. For each scale we map the input intensity to the output intensity, $R_I = \sum R_i$, using formula (1) where without logarithms the subtraction becomes a division:

$$R_I(x, y, c) = I_I(x, y)/F(x, y, c) \otimes I_I(x, y) \tag{7}$$

with $F(x, y, c)$ given by (2) above. To get a luminance version of MSR, we simply use formula (4) with the arbitrary channel "i" being replaced by the single intensity result. This method has the appeal that the luminance is in a space which is locally approximately linear, and thus the image which require little or no change should look more natural.

With an appropriate choice of scales, the above method can give an arbitrary amount of dynamic range compression. This is the case because a very small scale will remove all intensity differences, and reduce the image to a chromaticity image. Nonetheless, applying the above method to images with large dynamic range often gives a poor result at sharp shadow edges. The region in shadow is typically brightened significantly, but the edge itself becomes a dark area between two light areas, and thus looks unnatural.

Standard MSR typically does not brighten the shadow as much, but has much less of this edge effect, and the shadow simply looks like a less dark shadow. The reason for this difference is that a large part of the dynamic range compression of standard MSR is due to the logarithm operation. This can be verified by applying the processing without any ratios. The observation that the logarithm operation has a definite benefit leads us to the second method for luminance based MSR style processing.

This method is designed to provide the same dynamic range compression as original MSR. Here we define the image luminance by the geometric mean of the channels : $I_I = \left(\prod_i^N I_i\right)^{1/N}$. Although it is possible to use the arithmetic mean (as was done in [6]), the geometric mean is intuitively superior, as it gives a cleaner correspondence between the luminance of standard MSR and the luminance based alternative. Having computed the luminance, standard MSR processing is now applied to it, this time including the logarithm operation. In order to obtain an output luminance comparable to standard MSR, an additional step is needed. This is due to the observation above that MSR output should not be gamma corrected. Since we wish to gamma correct the output of the modified algorithm, we apply a reverse gamma correction to the MSR luminance result. Again the correspondence between the effect and the desired result is better served by the use of the geometric mean in place of the arithmetic mean. It should be noted that since we are only dealing with luminance, the reverse gamma correction need not be exact, and is adequately implemented with a power function. Specifically we raise the luminance to the 2.2 power, with any power in the 1.8 to 2.8 range being reasonable, depending on the monitor. If even more dynamic range compression is required, it can be obtained by simply omitting the reverse gamma correction step, but images processed in this manner to tend to look unnatural.

The next step is to apply the histogram based (or other) gain-offset method described above to the luminance. Thus having determined the desired relative intensify, we set each channel to the same chromaticity as in the input by:

$$R_i(x, y) = R_I(x, y)\frac{I_i(x, y)}{I_I(x, y)} \tag{8}$$

The processing so far has been designed to maintain color fidelity. However, this is not the same as producing the most pleasing color. If color enhancement is desired, then it is best added at this stage. For example, for some applications, increasing color saturation may be desired.

Next we map the pixels into the output range, typically [0, 255], recalling that the zero point is already set by the bottom clipping of the intensity. One possible solution is to simply scale the range to fit. However, often a better result is obtained by allowing some clipping of the upper range. The chromaticities of the pixels that are clipped will be a slightly incorrect, but this is not normally noticeable. It is not recommended, however, to do the same with the bottom of the range, as this can affect the chromaticities of all the pixels. Instead it is generally better to increase the amount of clipping on the bottom by doing so when the luminance range is adjusted.

The final step of the algorithm is to map the output into a space which produces linear output on the target device. In the case of a CRT monitor, this may be approximated by gamma correction.

In summary, we have an algorithm which maintains the dynamic range compression benefits of standard MSR, but is precise with respect to color. In addition, the algorithm requires less processing because we only need to perform convolutions on the luminance. Even if convolutions are performed using Fourier transforms, this is a non-negligible saving.

Results

We have tested the modified method of MSR processing on a number of images. Rather than attempt to portray color results in black and white, we have made some of the results available on the internet [15].

We first verified that for standard images, the first form of the dynamic range compression usually gives reasonable results. These images included ones from Kodak photo CD and ones taken with a three chip, 8 bit, Sony CCD camera. However, even some of these images had sufficiently strong shadow boundaries that the edge effect described above is noticeable. For these images, the second method gave better results. This is even more the case for images with extended dynamic range. Thus we conclude that overall the second method is a better choice when the dynamic range compression required is significant.

Next we explored the inter-play of the various methods and color constancy. We took images of the same scene with a shadow of varying strengths using two very differently colored lights. The first was a regular incandescent bulb which is a good illuminant for the indoor setting on our Sony CCD camera. The second illuminant was a cool white fluorescent together with a blue filter which creates an illuminant similar in chromaticity to that of deep blue sky. The same camera color temperature setting was always used, creating a color constancy problem. In one image the incandescent light source was near the camera resulting in an image which was both well color balanced and devoid of shadows. This was used as a reference. Then shadows of increasing strengths were put across the images. In order to explore the method fully, for each illuminant an image with a extraordinarily dark shadow was taken by combining several images taken at different apertures.

In general, the original MSR method without color correction grayed out the images. We used the color correction scheme to correct the color in the case of the reference image using a value of 125 for C in equation (6). This value gave reasonable color, but it is hard to verify that it is optimum without introducing a metric. There is no value of C which gives exactly the original color. Since we are comparing standard MSR to a method that has no such parameter, we feel it is fair to leave the value of C at the specified value. Since the modified algorithm was designed to preserve color, the results with that method did not gray out the image, and thus did not require color correction.

We turn now to the images taken under an illumination which is too blue due to the incorrect camera balance. Here the original MSR without color correction moves the image towards gray, and somewhat towards the appropriate color, achieving some degree of color constancy. The color is still far from the standard. When the color correction was applied, using the same constant as above, the image colors moved back towards the original, incorrect color. In fact, it is hard to see how to fix this problem with the original color correction method, even if one is allowed to change the parameter manually.

In the case of the modified algorithm, the color constancy processing using the method describe in [5] works well, producing an image close to the desired color, as set by the standard image. The subsequent MSR processing preserves this color, producing an image which has the benefits of the MSR dynamic range compression, and is the desired color.

Conclusion

Standard Multi-scale retinex processing works quite well as a method of compressing an image's dynamic range so that the image contrast looks better. Standard MSR performs a mixture of local (via ratios) and global (via logarithms) contrast adjustment. Unfortunately, standard MSR has the drawback that it perturbs the image colors in quite unpredictable ways. We have analyzed the fundamental steps of MSR and disentangled the various operations so that their effects can be handled separately, which also makes it possible to add in true color constancy processing as one of the steps. The resulting algorithm provides better color fidelity, has fewer parameters to specify. In addition, it is less computationally expensive.

Acknowledgments

This work was made possible in part by support from the Hewlett-Packard Corporation and the Natural Sciences and Engineering Council of Canada (NSERC).

References

1. D. J. Jobson, Z. Rahman, and G. A. Woodell, "Retinex Image Processing: Improved Fidelity To Direct Visual Observation", *Proceedings of the IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications, Scottsdale, Arizona*, November, pp. 124-126, 1996.
2. Z. Rahman, D. J. Jobson, and G. A. Woodell, "A Multiscale Retinex for Color Rendition and Dynamic Range Compression", *SPIE International Symposium on Optical Science, Engineering and Instrumentation, Applications of Digital Image Processing XIX*, Proceedings SPIE 2825, Andrew G. Tescher, ed., 1996.
3. D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and Performance of a Center/Surround Retinex," IEEE Transactions on Image Processing, March 1997.
4. D. J. Jobson, Z. Rahman, and G. A. Woodell, "A Multi-Scale Retinex For Bridging the Gap Between Color Images and the Human Observation of Scenes," IEEE Transactions on Image Processing: Special Issue on Color Processing, July 1997.

5. B. Funt, V. Cardei, and K. Barnard, “Learning Color Constancy,” Proc. Fourth IS&T/SID Color Imaging Conf., pp. 58-60, Scottsdale, Nov. 19-22, 1996.
6. B. V. Funt, K. Barnard, M. Brockington, and V. Cardei, "Luminance-based multi-scale Retinex," *Proceedings AIC Color 97*, Kyoto, Japan, May 25-30 (1997).
7. Eastman Kodak Company, "Fully Utilizing Photo CD Images: Article No. 4—PhotoYCC Color Encoding and Compression Schemes," 1994, available from 15://15.Kodak.com/pub/photo-cd/general/pcd045.txt.
8. E. H. Land, "Recent advances in Retinex theory and some implications for cortical computations: Color vision and the natural image", *Proc. Natl. Acad. Sci.*, 80, pp. 5163-5169, 1983.
9. E. H. Land, “Recent advances in Retinex theory”, Vision Res., 26, pp. 7-21, 1986.
10. E. H. Land, "An alternative technique for the computation of the designator in the Retinex theory of color vision". *Proc. Natl. Acad. Sci.* USA, Vol. 83, pp. 3078-3080, 1986.
11. D. Forsyth, "A novel algorithm for color constancy," *International Journal of Computer Vision*, 5, pp. 5-36, 1990.
12. G. D. Finlayson, “Color Constancy in Diagonal Chromaticity Space," In *Proceedings: Fifth International Conference on Computer Vision*, pp 218-223, (IEEE Computer Society Press, 1995).
13. W. Freeman and D. Brainard, “Bayesian Decision Theory, the Maximum Local Mass Estimate, and Color Constancy," in *Proceedings: Fifth International Conference on Computer Vision*, pp 210-217, (IEEE Computer Society Press, 1995).
14. K. Barnard, G. Finlayson, and B. Funt, "Color constancy for scenes with varying illumination," In *Proceedings of the 4th European Conference on Computer Vision*, pp. II:1-15, Bernard Buxton and Roberto Cipolla, eds., Springer, 1996.
15. ftp.cs.sfu.ca://pub/cs/color/IST-97