Cubical Gamut Mapping Colour Constancy

Milan Mosny, Brian Funt; Simon Fraser University; Burnaby, British Columbia, Canada

Abstract

A new color constancy algorithm called Cubical Gamut Mapping (CGM) is introduced. CGM is computationally very simple, yet performs better than many currently known algorithms in terms of median illumination estimation error. Moreover, it can be tuned to minimize the maximum error. Being able to reduce the maximum error, possibly at the expense of increased median error, is an advantage over many published color constancy algorithms, which may perform quite well in terms of median illumination-estimation error, but have very poor worst-case performance. CGM is based on principles similar to existing gamut mapping algorithms; however, it represents the gamut of image chromaticities as a simple cube characterized by the image's maximum and minimum rgb chromaticities rather than their more complicated convex hull. It also uses the maximal RGBs as an additional source of information about the illuminant. The estimate of the scene illuminant is obtained by linearly mapping the chromaticity of the maximum RGB, minimum rgb and maximum rgb values. The algorithm is trained off-line on a set of synthetically generated images. Linear programming techniques for optimizing the mapping both in terms of the sum of errors and in terms of the maximum error are used. CGM uses a very simple image pre-processing stage that does not require image segmentation. For each pixel in the image, the pixels in the N-by-N surrounding block are averaged. The pixels for which at least one of the neighbouring pixels in the N-by-N surrounding block differs from the average by more than a given threshold are removed. This pre-processing not only improves CGM, but also improves the performance of other published algorithms such as max RGB and Grey World.

Introduction

An image of a scene depends on the camera characteristics, the reflectances of the surfaces in the scene and on the scene illuminant. For many computer vision tasks such as object recognition or image tracking, it is desirable to remove the effects of variations in the chromaticity of the illuminant. Illumination-estimation algorithms offer one of many ways of approaching the problem. They attempt to estimate the chromaticity of the scene illuminant, often assuming the presence of only a single source of illumination, since once the chromaticity of the illuminant is known, the chromaticity of reflectances of the surfaces in the scene can be calculated.

There are many simple algorithms, such as MaxRGB [1], Grey World [2], Shades of Grey [3] and Grey Edge [4] which rely on various assumptions about the scene. MaxRGB assumes that the chromaticity of the maximum R, G, and B values in the image correspond to the chromaticity of the illuminant. Grey World assumes the average of RGB values in the scene is achromatic. Shades of Grey assumes the pth Minkowski norm of the RGB values to be achromatic. The Grey Edge algorithm generalizes the assumption even further by assuming that p^{th} Minkowski norm of the derivatives of RGB values in a scene is achromatic. Despite their simplicity, some of these

algorithms—Shades of Grey and Grey Edge in particular—have been shown to perform quite well for real-world images.

More sophisticated algorithms use statistical knowledge about the frequency with which different reflectances and illuminants occur in the real world. Neural Networks [5], Color by Correlation [6] and various gamut mapping algorithms, such as Forsyth's original algorithm [7] and Gamut Constrained Illuminant Estimation (GCIE) [8] fall into this category. These algorithms build statistical models of the distribution of RGB and/or rgb chromaticity values (r = R/(R+G+B), g = G/(R+G+B), b = G/(R+G+B)) in the image and then evaluate the input image relative to the statistical model of the world. Of these algorithms, GCIE [8] is, to our knowledge, currently one of the best performing color constancy algorithms.

Other color constancy algorithms exploit physical principles of image formation and often use Shafer's dichromatic model of reflectance [9]. Algorithms such as [10] and others rely on the presence of specularities or interreflections. The performance of the algorithms improves when combined with additional statistical knowledge [12] or constrained to a set of known illuminants [11].

This paper introduces a new color constancy algorithm called Cubical Gamut Mapping (CGM). CGM is computationally very simple, yet performs better than many currently known algorithms in terms of its median illumination-estimation error. It can also be tuned to minimize the maximum error. CGM combines statistical knowledge about the distribution of reflectances and illuminants in the real world with a simple assumption about the scene. In terms of its statistical model, CGM is based on principles similar to gamut mapping algorithms except that it represents the gamut of image chromaticities as a simple cube characterized by the image's maximum and minimum rgb chromaticities, rather than their more complicated convex hull. It also uses maximum RGB as an additional source of information about the illuminant. The estimate of the scene illuminant is obtained by linearly mapping the chromaticity of the maximum RGB, minimum rgb and maximum rgb values. The algorithm is trained off-line on a set of synthetically generated images. Linear programming techniques for optimizing the mapping in terms of both the sum of errors and in terms of the maximum error are used.

Color constancy algorithms for real world images rely on image pre-processing. Pre-processing is usually done in a separate step before the color constancy algorithm is invoked to reduce noise and to reduce the colour shifts created by chromatic aberration or demosaicing. It has been shown [13], that the pre-processing can significantly improve the performance of color constancy algorithms. Finlayson [19] suggested that further performance gains can be obtained by introducing new pre-processing algorithms.

CGM uses a very simple image pre-processing stage that does not require image segmentation. For each pixel in the image, the pixels in the N-by-N surrounding block are averaged, and then pixels for which at least one of the neighbouring pixels in the N-by-N surrounding block differs

from the average by more than a given threshold are removed. Dark pixels are also removed. This pre-processing not only improves CGM, but also improves the performance of other published algorithms such as max RGB and Grey World.

The paper is structured as follows. The pre-processing algorithm used by CGM is described in section 2. Section 3 introduces CGM. Section 4 discusses the results obtained by running CGM on the SFU test set of images [14]. Section 5 concludes.

Image Pre-processing

The image pre-processing algorithms try to address problems introduced by random pixel noise such as shot noise and colour shifts created by chromatic aberration and demosaicing artifacts. Generally, color constancy algorithms assume there is only a single scene illuminant. Hence secondary sources of illumination and interreflections present in the scene may affect the performance of the algorithms. We make the assumption that the darker parts of the image are more likely than the brighter parts to be illuminated by a secondary illuminant and so the pre-processing removes them from further consideration. For example, objects illuminated by the sky alone are darker than objects that are illuminated directly by the sun. Removing very dark pixels eliminates the cases where camera noise and quantization errors may be high; however, we found additional improvement by removing less dark pixels as well, and attribute the improvement to a reduction in the effects of secondary illuminants.

As Barnard showed [13], the illumination-estimation errors measured in RMS distance in rg-chromaticity space $(r = R/(R+G+B), g = G/(R+G+B))$ may vary by as much as 30% depending on the pre-processing method used. As an example, consider the well known MaxRGB color constancy algorithm which computes the rgb of the illuminant by finding the maximum R value in the whole image, maximum G value and maximum B value and then normalizing the result. The MaxRGB RMS distance in rg-chromaticity space error ranges from 0.053 to 0.79 depending on the pre-processing method [13].

Existing Pre-processing Methods

Some of the pre-processing methods reported in the literature are:

Region-growing. Region-growing has been used by Barnard [13], as well as by many other authors, to stabilize the RGB values used in subsequent colour constancy processing. The region-growing was subjected to two constraints: chromaticities in a region were within an absolute threshold of each other, and the brightness values in a region were within a relative threshold of each other. Regions that contain less than a given number of pixels are discarded. Once the regions were found, the averages of the RGBs in each region are used as input to the color constancy algorithms.

Block averaging. The RGBs of pixels within fixed-sized, non-overlapping blocks are averaged and the resulting average RGB values are used in subsequent processing [13]. A variation on the block-averaging method checks the variance and discards those image blocks having high RGB variance.

Binarized RGB [13]. This technique distributes blocked averaged results into discrete bins in RGB space, with 100 discrete values per channel resulting in a total of 1,000,000 bins. The average RGB value from each bin is used as the pre-processing output. Note, that binarized RGB technique removes the information about the number of pixels in each of the bins and simply reports the presence or absence of each of the 1,000,000 possible colors.

Gaussian smoothing. Gaussian smoothing reduces the noise and is used as a pre-processing step for the Grey Edge method [4].

Dark pixel removal. Dark pixels, those with R + G + B < Threshold are removed. Dark pixels tend to be adversely affected by camera and quantization noise. As well, brighter pixels have a higher chance of being directly illuminated by the main illuminant. Dark pixel removal was used, for example, in [15].

Proposed Pre-Processing—Even Blocks

We propose a pre-processing method that is similar to dark-pixel removal combined with a modified block averaging strategy. Each pixel is considered relative to its N-by-N neighbourhood of surrounding pixels. If the R, G, B and r, g, b code values of each surrounding pixel do not differ from the corresponding means of the R, G, B and r, g, b code values in the neighbourhood by more than some threshold amount then the central pixel is kept and the output (R, G, B) for the central pixel is set to be the mean code value of the pixels in the neighbourhood. If any one of the R, G, B or r, g, b values from any of the surrounding pixels differs from the R, G, B or r, g, b means by more than the given threshold, then the central pixel is flagged as unreliable in the output. A color constancy algorithm that uses the pre-processed output then simply disregards all such flagged pixels. In our experiment, R, G, B code values are first normalized to the range [0, 1]. We used N = 5 and a threshold of 0.1. The threshold for the removal of dark pixels is set to the mean image intensity, where intensity is computed as (R+G+B)/3.

The effect of the algorithm is to reduce noise by averaging pixels from within an N-by-N block. The algorithm removes pixels whose neighbourhoods vary in any of the channels R, G or B or any of the chromaticities r, g, or b. Therefore, the algorithm tends to discard pixels near sharp edges, and thus removes pixels that might be affected by chromatic aberration. The algorithm preserves more color variation in the scene than segmentation because the regions from which the average is computed are limited in size. Also note that segmentation may not remove chromatic aberrations along longer edges because the pixels along the edge may collectively form a large enough region to be retained. Gaussian smoothing and block averaging both tend to reduce the effects of chromatic aberration by averaging edge pixel values with their neighbours, but do not eliminate chromatic aberration's effects completely. The proposed algorithm has a better chance of preserving smaller patches of color and smaller specular highlights than block averaging with the variance check. This is because the proposed algorithm does not divide the image into non-overlapping blocks whose boundaries may split smaller patches of color. Rather, it centers a block on each pixel.

Results

The pre-processing algorithm has been tested with MaxRGB and Grey World on the SFU color constancy test set [14]. The set consists of 321 16-bit images of 30 scenes containing both matte and specular objects illuminated by 11 different illuminants (the test-set authors removed a few of the 11 x 30 possible scene/light combinations). The performance of

the algorithms is measured in terms for median angular error computed in the rgb sensor space of the camera. The MaxRGB algorithm combined with the new pre-processing technique returns a median angular error of 3.0 degrees. To our knowledge, the best reported result for the MaxRGB algorithm on the same test set using the same error metric so far has been 4.0 degrees [15]. Similarly, the pre-processing algorithm reduces Grey World's median angular error to 6.0 degrees from 6.8 degrees when simple Gaussian Smoothing is used [4].

Cube Gamut Mapping Algorithm

We show a computationally very simple color constancy algorithm that, when combined with the pre-processing introduced in the previous section, performs very well in comparison to some of the best color constancy algorithms.

Gamut mapping colour constancy [7] reformulated to include constraints based on considering only a fixed set of illuminants (GCIE) [8]) is one the best performing color constancy algorithms for which there are published results on the SFU test set of 321 images. For testing on this set, GCIE selects from 87 possible illuminants, 11 of which are ones used in the test set. Gamut mapping exploits the relationship between the gamut of chromaticities found in a scene and the gamut of possible chromaticities for all known reflectances under a given illuminant [7]. Gamut mapping algorithms are based on the fact that a chromaticity gamut can be modeled as a convex structure. It has been shown, that illuminating a convex set of reflectances results in a convex set of RGB responses, as well as a convex set of rgb chromaticities [7][8].

In the algorithm proposed here, the modeling of the gamut is simplified. Instead of computing the gamut as the convex hull of the observed sensor responses, the gamut is represented by the smallest cube in rgb space containing all chromaticities in the image. The bounding cube is found by simply determining the largest and smallest values of r, g and b (separately) in the image. In addition to the gamut of the observed colors, the algorithm also uses the maximum R, G, B values from the image. It is apparent from the excellent performance of the MaxRGB algorithm shown in the previous section, that the 3 maximal R, G and B values carry a lot of potentially useful information about the scene illuminant.

The chromaticity of the scene illuminant is estimated by a linear mapping transforming a column vector P of 9 input values P_i to a column vector O of 3 output values O_j corresponding to the rgb of the illuminant. The 9 input P_i values are the chromaticity of the maximum RGB values, the maximum rgb values, and the minimum rgb values. The mapping is represented by a 9-by-3 matrix W. The output rgb O is computed as

$$O = WP$$

Training Phase

The coefficients W_{ij} of the matrix W are computed off-line using a training set of 6,650 synthetic images. Reflectances and illuminants for the synthetic images were sourced from an existing database of 1995 reflectance and 87 illuminant spectra [14]. The training images were generated using the POWRay ray tracing graphics package [16]. Examples of the generated images are shown in Figure 1. Each image contains one of two different background planes, and between 0 and 32 spheres of random size. Each surface is painted with a random choice of one of the 1995 reflectances. The degree of specularity is varied

randomly according to a Gaussian distribution with zero mean and standard deviation of 0.2, clipped to the interval [0, 1]. The roughness parameter was chosen randomly from the interval [0, 0.25]. The diffuse parameter was set to 1.0. No gamma correction was done, thereby preserving the linearity of the generated images.
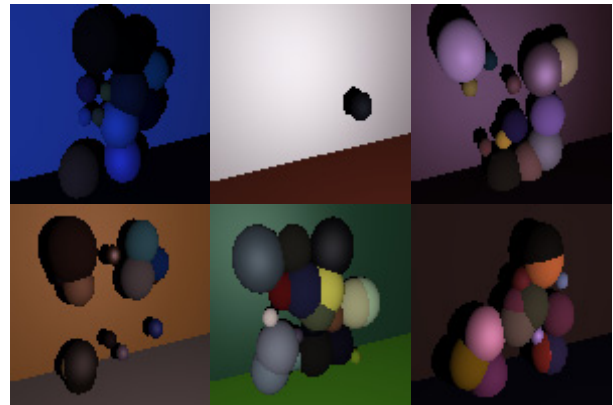

Figure 1. Five examples of synthetic training images generated using POWRay.

The training data was collected from the training set by computing the chromaticity of max RGB, max rgb and min rgb and pairing these with the rgb of the synthetic scene illuminant. However, it is also be possible to train the algorithm using a large set of real images for which the rgb of the illuminant has been measured. Knowledge of the camera sensitivities is not required because only max RGB, max rgb, min rgb from each training image along with the rgb of the corresponding illuminant are required for training. When the camera sensitivities are known, training with synthetic images is perhaps easiest since it is simpler to generate a large number of images on a computer than to acquire them in the real world.

A set of linear equations describes the training set of images. Let P_k be a column vector consisting of chromaticity of the maximum RGB values, the maximum rgb values, and the minimum rgb values obtained from the training image k. Let L_k be the true scene-illuminant chromaticity for training image k. The estimated illuminant chromaticity of the kth image is given by $O_k = WP_k$. We define column vector E_k as the channel-by-channel difference between the estimated and true chromaticities of the illuminant for the training image k,

$$E_k = L_k - WP_k$$

For each chromaticity channel j from the set of channels {r, g, b}, the equation becomes

$$E_{jk} = L_{jk} - \sum_i W_{ij} P_{ik}$$

The task of training the system given a set of pre-processed training images is then formulated as a problem of finding the matrix of weights, W, minimizing the sum, E, of the absolute values of errors E_{jk}. In other words, W must minimize

$$E = \sum_{jk} \left| E_{jk} \right|$$

The minimization is accomplished using the curve-fitting-with-linear-programming technique explained and proven by Swanson et al. [17] in which the absolute value operation is handled by rewriting the minimization in terms of an equivalent

Table 1 Median angular error, mean angular error, RMS angular error and maximum angular error for selected algorithms. (Results for region growing pre-processor are sourced from [8].)

Algorithm	Preprocessor	Median Error	Mean Error	RMS Error	Max Error
Grey World	Gaussian sigma=6	6.82	9.74	13.45	36.72
Grey World	even blocks	6.00	9.81	14.03	38.16
Shades of Grey, Minkowski norm = 7	none	3.72	6.40	9.31	29.60
Grey Edge, 2nd derivative, norm = 7	Gaussian sigma=5	2.74	5.26	7.82	27.17
MaxRGB	region growing	4.02	6.30	8.77	26.19
MaxRGB	even blocks	3.02	5.23	8.16	25.29
Color by Correlation	region growing	3.19	6.56	10.09	28.78
GCIE v3, 11 lights	region growing	*1.31*	4.18	6.88	27.64
GCIE v3, 87 lights	region growing	2.60	4.75	7.11	19.43
Gamut Mapping	region growing	2.92	4.17	5.60	23.19
CGM	even blocks	**2.46**	**3.77**	**5.22**	19.34
CGM Emax,j = 0.21	even blocks	3.65	4.50	5.52	**16.19**

linear problem. Each error E_{jk} is represented as a difference $E_{jk} = E^+_{jk} - E^-_{jk}$ of a nonnegative "underestimation" error E^+_{jk} and nonnegative "overestimation" error E^-_{jk}. If E_{jk} is > 0, then $E^-_{jk} = 0$ and $E^+_{jk} = E_{jk}$. If $E_{jk} < 0$, then $E^-_{jk} = \bullet E_{jk}$ and $E^+_{jk} = 0$. Absolute value $|E_{jk}|$ can be then computed as $|E_{jk}| = E^+_{jk} + E^-_{jk}$. The system of equations becomes

$$E = \sum_{jk} E^+_{jk} + E^-_{jk}$$

$$E^+_{jk} - E^-_{jk} = L_{jk} - \sum_i W_{ij} P_{ik}$$

subject to the constraints

$$E^+_{jk} \geq 0, \; E^-_{jk} \geq 0$$

for each channel j from the set of channels {r, g, b} and for each training image k.

This system has the same solution, *W*, as the original formulation, with the advantage that it can be solved via linear programming.

Maximum Error Optimization

We also introduce a modification to the CGM algorithm in order to minimize the maximum error based on the approach described in [18]. Minimization of the maximum error is achieved by introducing additional constraints that limit each training image error E_k to a predefined maximum value E_{max}. The predefined maximum value is then progressively lowered until it is no longer possible to find a solution to the linear programming problem. The constraints take form

$$\sum_j E^+_{jk} + E^-_{jk} \leq E_{max}$$

for each training image k.

Although this actually minimizes the sum of chromaticity errors and not the angular error, the sum of chromaticity errors imposes an upper boundary on the corresponding angular error.

Another option is to minimize the maximum errors $E_{max,j}$ on a per-channel basis and require

$$E^+_{jk} + E^-_{jk} \leq E_{max,j}$$

for each channel j from the set of channels {r, g, b} and for each training image k.

This formulation makes it possible to set different upper limits on the errors from different channels.

Results

The algorithms were tested on the 321 images of the SFU color constancy test image set. The images in this set are 16-bit linear. The median angular error, mean angular error, RMS angular error and maximum angular error are reported. All errors are calculated in the RGB sensor space so as to make them readily comparable to the results published for other algorithms. Results for CGM and CGM with maximum error optimization are compared to the current state-of-the-art algorithms, which can be roughly divided into two groups. The first group contains simple, computationally inexpensive algorithms including Shades of Grey, standard Grey World, 2nd degree Grey Edge, and MaxRGB. While the performance of these algorithms in terms of median error is comparable to more complex algorithms, their RMS errors tend to be higher, which may indicate the presence of a larger number of poor illumination estimates. The second group contains more sophisticated, usually more computationally expensive, algorithms that have a lower RMS error. The results are summarized in Table 1.

CGM is computationally very simple. It outperforms many other computationally inexpensive algorithms in terms of median error, mean error, RMS error and max error. It also outperforms all the more complex algorithms on all statistics shown here with a single exception. The only algorithm that outperforms CGM in terms of median error is GCIE, which is given prior knowledge about the specific 11 lights used in the test set. However, CGM outperforms (in terms of median angular error) GCIE when it uses the more general set of 87 lights. In terms of worst-case performance, CGM with maximum-error optimization clearly outperforms all the other algorithms with respect to the maximum angular and RMS errors, while still producing very good results in terms of mean and median errors.

Conclusion

A simplified gamut-mapping colour constancy algorithm called Cubical Gamut Mapping (CGM) is introduced along with an improved pre-processing method for noise and colour

artifact removal. In CGM, the colour gamut is represented by a cube rather than a convex hull. The training phase depends on an optimization which is solved by linear programming. Experimental results on a standard test set show that CGM outperforms many other similar algorithms in terms of median angular error, mean angular error and RMS angular error. As well, it outperforms or ties the other algorithms in terms of maximum angular error.

Another feature of CGM is that it can be tuned to minimize the maximum error. CGM with maximum error optimization clearly outperforms many other algorithms with respect to the maximum angular and RMS errors, while still producing very good results in terms of mean and median errors.

References

[1]   E. H. Land, "The retinex theory of color vision," Scientific American, 237(6), 108 (1977).

[2]   G. Buchsbaum, "A spatial processor model for object colour perception," Journal of the Franklin Institute, 310, (1980).

[3]   G. Finlayson and E. Trezzi, Shades of gray and colour constancy, Proc. IS&T/SID Twelfth Color Imaging Conference, pg. 37. (2004).

[4]   J. van de Weijer, T. Gevers and A Gijsenij, "Edge-Based Color Constancy," IEEE Transactions on Image Processing, 16(10), 2207 (2007)

[5]   V. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity using a neural network," Journal of the Optical Society of America A, 19(12), (2002).

[6]   G. Finlayson, S. Hordley, and P. Hubel, "Color by correlation: A simple, unifying framework for color constancy," IEEE Trans. on Pattern analysis and Machine Intelligence, 23(11), 1209 (2001).

[7]   D. Forsyth, "A novel algorithm for color constancy," International Journal of Computer Vision, 5(1), 5 (1990).

[8]   G. Finlayson and S. Hordley, "Gamut constrained illumination estimation," International Journal of Computer Vision, 67(1), 93 (2006).

[9]   G. Shafer, "Using color to separate reflection components," Color Research and Application, 10(4), 210 (1985).

[10]   H. C. Lee, "Method for computing the scene-illuminant chromaticity from specular highlights," Journal of the Optical Society of America A, 3(10), 1694 (1986).

[11]   J. Toro, "Dichromatic illumination estimation without pre-segmentation," Pattern Recognition Letters, 29, 871 (2008).

[12]   G. Schaefer, S. Hordley and G. Finlayson, Combining physical and statistical evidence for computational colour constancy, Proc. 10th Congress of the International Colour Association, (2005).

[13]   K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms-part ii: Experiments with image data," IEEE Transactions on Image Processing, 11(9), 985 (2002).

[14]   K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for colour research," Color Research and Application, 27(3) 147 (2002).

[15]   S. Hordley and G. Finlayson, "Reevaluation of color constancy algorithm performance," The Journal of the Optical Society of America A, 23(5), 1008 (2006).

[16]   Persistence of Vision Pty. Ltd., "Persistence of Vision Raytracer (Version 3.6)," Computer software, http://www.povray.org/download/, (2004).

[17]   H. Swanson, R. E. D. Woolsey, "Curve fitting with linear programming," ACM SIGMAP Bulletin, 19, 22 (1975).

[18]   S.C. Narula, J.F. Wellington, "Multiple criteria linear regression," European Journal of Operational Research, 181, 767 (2007).

[19]   G. Finlayson, personal communication.

Author Biography

Milan Mosny earned an Ing. degree in Electrical Engineering at Slovak Technical University and an M.Sc. in Computing Science at Simon Fraser University. After spending a number of years in software industry, he is currently a Ph.D. candidate at Simon Fraser University.

Brian Funt obtained his Ph.D. in Computer Science from the University of British Columbia in 1976. He has been a Professor of Computing Science at Simon Fraser University since 1980.