TRAINING-BASED COLOR CORRECTION FOR CAMERA PHONE IMAGES

Hasib Siddiqui and Charles A. Bouman

School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285, USA

ABSTRACT

In this paper, we propose a method for improving the color rendition of low quality cell phone camera images. The proposed method is based on a multi layer stochastic framework whose parameters are learned in an offline training procedure using the well known expectation maximization (EM) algorithm. The color correction algorithm functions by first making soft assignments of images into defect classes and then processing images in each defect class with an optimized algorithm, which we refer to as resolution synthesis-based color correction (RSCC). The parameters of the color correction algorithm are trained using pairs of low quality images, obtained from real cell phone cameras, and high quality spatially registered reference images, captured with a high quality digital still camera.

We present experimental results comparing the performance of our method to some existing commercial color correction algorithms.

***Index Terms*—** Color correction, color cast, cell phone camera.

1. INTRODUCTION

Recently, cell phone cameras have become very popular because of their portability and multipurpose functionality. The images captured using cell phone cameras appear to have acceptable quality when viewed on the mobile LCD screens. However, when downloaded to a computer and viewed on a high quality CRT/LCD display, the colors in a cell phone camera image seem off from colors in the original scene. The objectionable colors may appear either as an unwanted color cast, characterized by a single color dominant in the entire image, or local hue shifts that may affect different portions of the captured image differently.

The unwanted color casts in an image can potentially be corrected using color constancy processing [1]. One important contribution in color constancy processing is color by correlation proposed by Finlayson *et al.* [2]. The method exploits the correlation between scene illuminants and global distribution of image colors to determine the optimal linear color transformation for color correcting the input image.

An important contribution in digital color enhancement is multi scale retinex with color restoration (MSRCR) [3]. The goal of the MSRCR algorithm is to improve the overall image quality by providing simultaneous contrast enhancement, dynamic range compression, and color constancy. However, in our experience, the MSRCR algorithm shows limited success in improving the color rendition of low quality cell phone camera photos.

In this paper, we propose a novel algorithm for improving the color rendition of low quality cell phone camera images. The algorithm is based on a multi layer stochastic framework that draws inspiration from color by correlation [2] and resolution synthesis (RS) [4, 5]. Similar to the concept in [2], a training procedure is used to learn the parameters for the probability distribution of colors from a set of cell phone camera images displaying a particular type of global color distortion. Next, global color attributes of the test image are used to compute the likelihood that the observed colors in the image are due to each of the global color distortions learned during training. Based on the computed likelihoods, an optimal color transformation is determined for correcting the image. Unlike in [2], the color transformation is non-linear and spatially variant. Using a scheme similar to RS [4, 5], the color transformation at a pixel location is determined by the color of neighboring pixels in the local window. We use pairs of low quality images obtained from a multitude of cell phone camera sources and spatially registered reference images to train our algorithms. The reference images are captured with a high quality camera and are spatially registered to sub-pixel accuracy. The resulting pairs of images accurately represent the real-world non-idealities typically found in real mobile camera pictures.

This research was supported by Hewlett-Packard Inc. The authors would like to thank Morgan Schramm for his valuable comments and Yeesoo Han for his image registration software.

2. MULTI LAYER FRAMEWORK FOR COLOR CORRECTION

The proposed color correction scheme uses a two layer hierarchical classifier in order to identify the color distortion at a given pixel location in the input image. Figure 1 shows how global image classification and local color classification are arranged in a hierarchical framework.

The global image classifier determines the likelihood that the input image suffers with a particular color defect commonly found in mobile camera pictures. In this paper, we use 4 defect classes to characterize global color distortions in cell phone camera images: (1) Global class 1 represents images with reddish color cast; (2) Global class 2 represents images with greenish/yellowish color cast; (3) Global class 3 represents images with bluish color cast; and (4) Global class 4 represents images with no dominant color cast. Example images for the 4 global classes are shown in figure 2. In each column, the top image is a cell phone camera image, while the bottom image is a reference image. The reference image is a picture of the same scene taken under the same lightning conditions using a high quality digital still camera. The assigned class labels are based on the observed color shift in the cell phone camera image with respect to its reference.

After global classification, images from each defect class are processed using a non-linear color correction algorithm which we call resolution synthesis-based color correction (RSCC). This algorithm is similar to the resolution synthesis predictor developed by Atkins *et al.* [4, 5], but it is adapted for color correction rather than resolution enhancement. The RSCC algorithm extracts the color feature vector from a neighborhood around the current pixel and performs its soft classification into a number of local color subclasses.

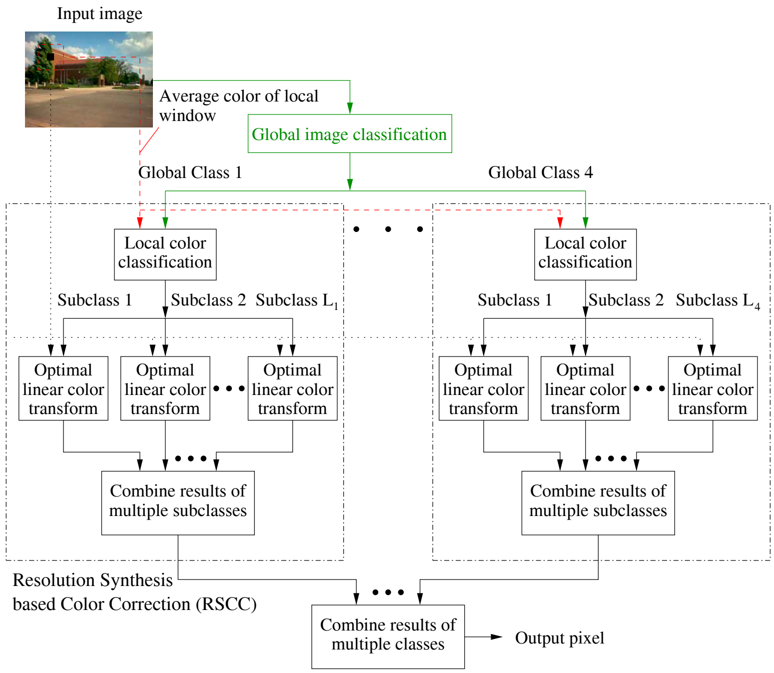

Fig. 1. Multi layer structure of color correction. The input image is first classified into multiple classes using the global color classification algorithm, then the average color in a local window is classified into multiple subclasses using a local color classification algorithm. The input pixel is processed with linear color transforms optimized for individual classes and subclasses, and the results of individual color transforms are combined to determine the color of the output pixel.

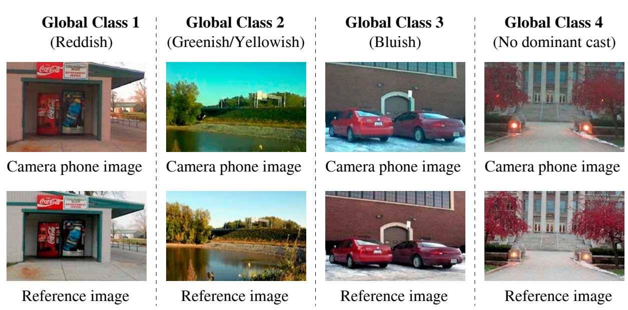

Fig. 2. Typical examples of the global classes assigned to camera phone images.

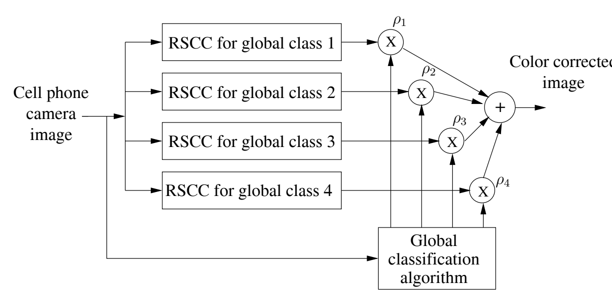

Fig. 3. Flow diagram of color correction algorithm.

An affine color transform associated with each subclass is next applied to the current pixel, and the outputs of the RSCC color transforms are combined to compute the final color corrected image.

3. COLOR CORRECTION ALGORITHM

Figure 3 shows a flow diagram of the color correction algorithm. We assume that the input cell phone camera image is in the sRGB color space with the RGB color values ranging from 0 to 255. The cell phone camera image is first input to the global classification algorithm. The global classification algorithm uses a stochastic image model to perform a soft classification of the image into four global classes. The global classes are as shown in figure 2. The output of the global classification algorithm is a set of 4 values ρ_k, where $0 \leq \rho_k \leq 1$ and $\sum_{k=1}^{4} \rho_k = 1$. Each ρ_k gives a measure of the likelihood that the input image belongs to the global class k. The cell phone camera image is next processed with each of the four RSCC predictors. Each RSCC predictor processes the input image to remove the color distortions associated with its global class. Finally, the outputs of the RSCC algorithms are combined using the global classification weights to yield the color corrected image.

3.1. Resolution Synthesis based Color Correction (RSCC)

Figure 4 shows the structure of the non-linear RSCC predictor. The prediction parameters comprise the classification parameters $\boldsymbol{\Theta}_k$ and color transform parameters $\boldsymbol{\Psi}_k$, where $1 \leq k \leq 4$, and are estimated in an offline training process.

For each RSCC algorithm, the training procedure is run independently using a separate set of training data. The training data comprises pairs of low quality cell phone camera images, belonging to a specific class of global color distortions, and their corresponding high quality reference images. The reference images are spatially

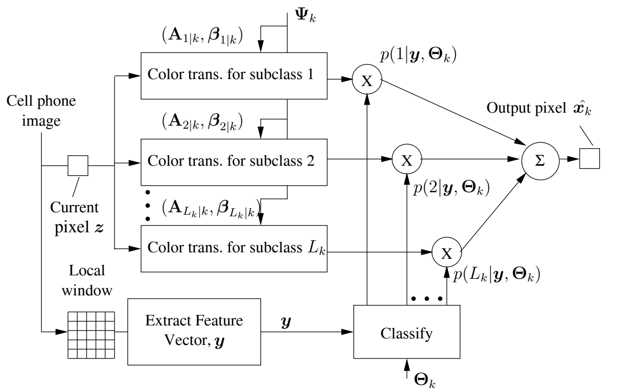

Fig. 4. Illustration of the RSCC predictor. For each global class k there is an optimized RSCC algorithm with classification parameters $\boldsymbol{\Theta}_k$ and color transform parameters $\boldsymbol{\Psi}_k$.

registered to the cell phone camera images to sub-pixel accuracy. For image registration, we use the sub-pixel registration software developed by Han [6].

To explain the working of the RSCC predictor, we shall adopt the following notation. We will use $\boldsymbol{z}$ to denote the RGB color value of the current pixel in the cell phone image, $\boldsymbol{x}$ to denote the RGB color value of the current pixel in the reference image, and $\hat{\boldsymbol{x}}_k$ to denote the value of $\boldsymbol{x}$ as estimated by the RSCC algorithm. Further, we shall use $\boldsymbol{y}$ to denote the color feature vector that is used for classification of the local window in the input image.

The RSCC algorithm assumes that the color feature vector is conditionally distributed as a mixture distribution with parameters $\boldsymbol{\Theta}_k$ given the global defect class k. The algorithm works by first computing the probability that $\boldsymbol{y}$ belongs to a particular subclass j in the mixture. The input pixel $\boldsymbol{z}$ is then processed with the optimal affine color transforms associated with the individual subclasses. Finally, the output of the RSCC predictor $\hat{\boldsymbol{x}}_k$ is computed as a linear combination of the outputs of all L_k color transforms with the weighting function for the j-th color transform corresponding to the probability that the image data is in subclass j.

3.1.1. *Optimal Prediction*

In this subsection, the output of RSCC is derived as an MMSE predictor, assuming that the prediction parameters $\boldsymbol{\Theta}_k$ and $\boldsymbol{\Psi}_k$ are known. The training process for computing the estimates of $\boldsymbol{\Theta}_k$ and $\boldsymbol{\Psi}_k$ is similar to that in [5]. We shall follow the convention of using uppercase letters for random variables and lower-case letters for their realizations.

The following three assumptions are made about the image data.

Assumption 1: The local color feature vector $\boldsymbol{Y}$ is conditionally distributed as a multivariate Gaussian mixture distribution given the class membership K of the input image. The conditional density shall be denoted as $p\left(\boldsymbol{y}|\boldsymbol{\Theta}_k\right)$, where $\boldsymbol{\Theta}_k$ represents the Gaussian mixture model (GMM) distribution parameters that comprise the cluster means $\boldsymbol{\mu}_{j|k}$, the cluster covariances ${\mathbf{\Lambda}_{j|k}}^2$, the cluster probabilities $\pi_{j|k}$, and the number of clusters in the mixture L_k. We write the GMM distribution parameters as $\boldsymbol{\Theta}_k = \left\{\pi_{j|k}, \boldsymbol{\mu}_{j|k}, \mathbf{\Lambda}^2_{j|k}\right\}_{j=1}^{L_k}$.

Assumption 2: The conditional distribution of $\boldsymbol{X}$ given $\boldsymbol{Z}$, J, and K is multivariate Gaussian, with mean $\mathbf{A}_{J|K}\boldsymbol{z} + \boldsymbol{\beta}_{J|K}$. The RSCC prediction parameters, represented by $\boldsymbol{\Psi}_k$, are obtained directly from the parameters for these distributions. So we write $\boldsymbol{\Psi}_k = \left\{\mathbf{A}_{j|k}, \boldsymbol{\beta}_{j|k}\right\}_{j=1}^{L_k}$.

Assumption 3: The subclass J is conditionally independent of the vectors $\boldsymbol{X}$ and $\boldsymbol{Z}$, given the color feature vector $\boldsymbol{Y}$ and the global class K. Formally, this means that

$$p\left(j|\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\Theta}_k\right) = p\left(j|\boldsymbol{y}, \boldsymbol{\Theta}_k\right). \qquad (1)$$

Using these assumptions, for each RSCC predictor, we may compute the MMSE estimate as

$$\begin{aligned} \hat{\boldsymbol{X}}_k &= E\left[\boldsymbol{X}|\boldsymbol{Z}, K=k\right] &(2)\\ &= \sum_{j=1}^{L_k} E\left[\boldsymbol{X}|\boldsymbol{Z}, K=k, J=j\right] p\left(j|\boldsymbol{Z}, \boldsymbol{\Theta}_k\right) &(3)\\ &= \sum_{j=1}^{L_k} \left(\mathbf{A}_{j|k}\boldsymbol{Z} + \boldsymbol{\beta}_{j|k}\right) p\left(j|\boldsymbol{Y}, \boldsymbol{\Theta}_k\right). &(4) \end{aligned}$$

Equation (4) is obtained by invoking assumptions 2 and 3, and using the fact that $p\left(j|\boldsymbol{Z}, \boldsymbol{\Theta}_k\right) = \int p\left(j|\boldsymbol{Z}, \boldsymbol{x}, \boldsymbol{\Theta}_k\right) p\left(\boldsymbol{x}|\boldsymbol{Z}\right) d\boldsymbol{x}$. The distribution $p\left(j|\boldsymbol{y}, \boldsymbol{\Theta}_k\right)$ can be computed using Bayes' Rule as follows

$$p\left(j|\boldsymbol{y}, \boldsymbol{\Theta}_k\right) = \frac{\pi_{j|k}|\mathbf{\Lambda}_{j|k}|^{-1} \exp\left\{-\frac{1}{2}\left\|{\mathbf{\Lambda}_{j|k}}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_{j|k})\right\|_2^2\right\}}{\sum_{m=1}^{L_k} \pi_{m|k}|\mathbf{\Lambda}_{m|k}|^{-1} \exp\left\{-\frac{1}{2}\left\|\mathbf{\Lambda}_{m|k}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_{m|k})\right\|_2^2\right\}}, \qquad (5)$$

where $\|.\|_2$ denotes the L_2 norm. The final estimate, $\hat{\boldsymbol{X}}$, for the color corrected pixel is computed using the relation

$$\hat{\boldsymbol{X}} = \sum_{k=1}^{4} \rho_k \hat{\boldsymbol{X}}_k. \qquad (6)$$

3.1.2. *Extraction of Color Feature Vector*

The color feature vector $\boldsymbol{y}$ is 3-dimensional and contains color information extracted from a 9×9 local window in the input image. To obtain the feature vector, we first compute an average of the RGB color values, denoted by R_a, G_a, and B_a, of similar pixels in the local window. To make the notion of similar pixels more concrete, we perform a color segmentation of the input image using the algorithm described in [7]. The RGB color averages are then computed using only those pixels in the local window that belong to the same color segment as the center pixel. Next, the average chromaticity values are computed as $r_a = R_a/(R_a + G_a + B_a)$, $g_a = G_a/(R_a + G_a + B_a)$, and $b_a = 1 - r_a + g_a$. Finally, the feature vector $\boldsymbol{y}$ is formed using the following three uncorrelated descriptors of local color: $Y_a = (0.30R_a + 0.59G_a + 0.11B_a)/255$, $rg_a = r_a - g_a$, and $yb_a = (r_a + g_a)/2 - b_a$.

3.2. Global Classification Algorithm

To compute the global classification weights ρ_k, the global classification algorithm computes the local color feature vector $\boldsymbol{y}_{mn}$ at each pixel (m,n) in the input image. Assuming that the conditional density of Y_{mn} given the global class membership K is a multivariate Gaussian mixture distribution with parameter $\boldsymbol{\Theta}_K$, the global classification algorithm computes the posterior class distribution for each pixel (m,n) in the image as

$$p(k|\boldsymbol{y}_{mn}, \boldsymbol{\Theta}_k) = \frac{p(\boldsymbol{y}_{mn}|k, \boldsymbol{\Theta}_k)p(K=k)}{\sum_{k=1}^{M} p(\boldsymbol{y}_{mn}|k, \boldsymbol{\Theta}_k)p(K=k)}. \qquad (7)$$

ρ_1	ρ_2	ρ_3	ρ_4	Dominant class
0.099	0.721	0.072	0.108	2

Table 1. Results of global image classification.

Assuming a uniform prior class distribution, $P(K = k) = 1/4$, the posterior class distribution can then be written as

$$p(k|\boldsymbol{y}_{mn}, \boldsymbol{\Theta}_k) = \frac{p(\boldsymbol{y}_{mn}|k, \boldsymbol{\Theta}_k)}{\sum_{k=1}^{4} p(\boldsymbol{y}_{mn}|k, \boldsymbol{\Theta}_k)}, \tag{8}$$

and the global classification coefficients, then computed as

$$\rho_k = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} p(k|\boldsymbol{y}_{mn}, \boldsymbol{\Theta}_k), \tag{9}$$

where M and N are the height and width of the input image.

4. EXPERIMENTAL RESULTS

The training data for optimizing the global classification and RSCC prediction parameters was obtained using one high quality digital still camera, a Nikon D-100, and two low quality camera cell phones, a Nokia 3650 and a Sony Ericsson P900. The performance of the color correction algorithm was evaluated on a set of 220 test images obtained from 6 different camera cell phones: Motorola V600, Sony Ericsson P900, Sony Ericsson T300, Samsung E715, Nokia 3650, and JPhone Sharp J-SH52. None of the test images was used for training the algorithm.

In figure 5, we compare the performance of the proposed method to other commercial color algorithms on an example cell phone image. Figure 5(a) shows the low quality cell phone camera image captured using a Nokia 3650. Figure 5(b) shows the high quality reference image obtained using a Nikon D-100. Figures 5(c) - (e) show processed cell phone images with 3 commercial color correction algorithms: "Auto Color" algorithm in Adobe® PhotoShop® 7.0, "Smart White Balance" algorithm in Corel® Paint Shop Pro® X, and "Retinex" algorithm in TruView PhotoFlair®. Figure 5(f) shows the result from the proposed training-based color correction algorithm. The results of global classification are listed in table I.

The test result shown in figure 5(f) is typical of the performance that we have achieved with our training-based color correction algorithm over more than 220 test images. From the observed performance, we feel that the proposed method is better suited for improving the quality of cell phone camera images than the existing methods.

5. CONCLUSION

The experimental results show that the proposed algorithm performs better on cell phone camera images than other commercial color correction algorithms. The training based approach helps the algorithm to learn parameters of defect classes specific to cell phone camera images. The color correction is achieved by applying non-linear color transformations that have been optimized beforehand for removing color artifacts associated with specific defect classes learned during training.

(a) (b) (c) (d) (e) (f)

Fig. 5. Outdoor scene classified as predominantly yellowish/greenish. (a) Cell phone camera image. (b) Reference image. Processed images using (c) "Auto Color" in Adobe PhotoShop® 7.0, (d) "Smart White Balance" in Corel Paint Shop Pro® X, (e) "Retinex" in TruView PhotoFlair®, and (f) Training-based Color Correction Algorithm.

6. REFERENCES

[1] Kobus Barnard, Lindsay Martin, Adam Coath, and Brian Funt, "A comparison of computational color constancy algorithms–part II: Experiments with image data," *IEEE Trans. on Image Processing*, vol. 11, no. 9, pp. 985–996, September 2002.

[2] G. D. Finlayson, S. D. Hordley, and P. Hubel, "Colour by correlation: a simple, unifying framework for colour constancy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209–1221, November 2001.

[3] 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 Trans. on Image Processing: Special Issue on Color Processing*, vol. 6, no. 7, pp. 965–976, 1997.

[4] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using pixel classification," in *ICIP*, Thessaloniki, Greece, 2001, vol. 3, pp. 864–867.

[5] C. B. Atkins, *Classification-Based Methods in Optimal Image Interpolation*, Ph.D. thesis, Purdue University, 1998.

[6] Yeesoo Han, *Sub-Pixel Registration and Image Analysis with Application to Image Comparison*, Ph.D. thesis, Purdue University, 2005.

[7] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Proc. of IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, June 1997, pp. 750–755.