A Color Balancing Algorithm for Cameras

Noy Cohen
Email: ncohen@stanford.edu
EE368 Digital Image Processing, Spring 2011 - Project Summary
Electrical Engineering Department, Stanford University

***Abstract*—Color constancy refers to the ability to recognize the true color of objects in a scene regardless of the illumination which is incident upon them. While the human visual system is color-constant to a large extent, digital cameras have to rely on fast color balancing algorithms, integrated into their image signal processing (ISP) pipeline, in order to estimate the illumination in a scene and correct for it digitally. In this paper, we discuss known methods of applying color balancing in cameras, and argue that separating the correction stage into two parts - diagonal correction before the demosaicing stage and a linear correction afterwards, can result in improved image quality. We continue to analyze the color gamut of natural illumination using a large database of natural images, coupled with ground truth illumination information, and derive a new, fast color balancing algorithm for digital cameras. Contrary to existing methods, our method uses only part of the scene gamut which may correspond to neutral gray color to estimate the illumination. We compare our method against six known fast color balancing algorithms on the database. Our experiments show that the proposed method outperforms these algorithms, while maintaining similar computational complexity.**

***Index Terms*—Color constancy, color balancing, white balancing, illuminant estimation, ISP, digital camera pipeline**

I. Introduction

Color constancy refers to the ability to recognize the true color[1] of objects in a scene regardless of the illumination which is incident upon them. Throughout the 20th century, several studies and color matching experiments have shown that the human visual system is color constant to a large extent ([6], chapter 9) - we are able to recognize colors with high fidelity under a wide range of natural illuminations. In his seminal work on color vision, including the development of the Retinex theory in the late 70s [4], Edwin Land showed that in interpreting the color of a certain object, the human visual system does not rely on the absolute composition of wavelengths emanating from it, but rather on its relative spectral composition compared with its environment.[2] He also suggested an algorithm that predicts the color of an object uniformly illuminated by a light source with arbitrary spectrum as a model for the human visual system (an analysis of Land's algorithm can be found in [7]). Land's theory and experiments led to the development of a variety of algorithms that try to estimate the illumination component from an input image and achieve color constancy in imaging systems. Many image processing and vision applications such as image segmentation, feature detection and object recognition, which are often applied to color images, stand to benefit from such algorithms.

With the emergence of digital cameras, several algorithms have been proposed for automatic in-camera color-balancing (also referred to as auto white-balancing, or AWB) in order to compensate for illumination effects and display color images as perceived by the human eye.[3] Contrary to some color constancy methods which are computationally intensive, such algorithms run in real time, as part of the image signal processing (ISP) pipeline in the camera, and therefore have to meet stringent timing constraints. In this work we will focus on simple and fast color constancy algorithms that can run in real time and can be implemented as part of a camera's ISP pipeline. A survey of various color constancy algorithm, including a family of gamut-mapping algorithms which we do not discuss herein, is given in [8], [9].

The problem of separating the illumination information from the reflectance can be formulated as follows - assume an imaging system composed of optical elements and an $M \times N$ size sensor. A pixel's R,G,B values at location (x,y) on the sensor, denoted $p_i(x,y)$, $i \in \{R,G,B\}$ depend on the object's reflectance values $r(\lambda,x,y)$ (assuming a Lambertian surface), where λ is the wavelength, on the illumination $l(\lambda)$ and on the camera sensitivity functions $c_i(\lambda)$, $i \in \{R,G,B\}$

$$p_i(x,y) = \int_\Lambda r(x,y,\lambda)\, l(x,y,\lambda)\, c_i(\lambda)\, d\lambda. \qquad (1)$$

Our formulation assumes that the camera's sensitivity functions are independent of sensor location (i.e. we're neglecting effects such as lens vignetting, chromatic aberrations and sensor nonuniformity). Color constancy algorithms try to estimate the illumination information in each pixel location $e(x,y,\lambda)$, or its projection on the camera sensitivity functions

$$l_i(x,y) = \int_\Lambda l(x,y,\lambda)\, c_i(\lambda)\, d\lambda,\ i \in \{R,G,B\}, \qquad (2)$$

[1] By using the intuitive phrase "true color", we mean the color of the object as perceived under some canonical illumination, such as white light (with flat spectrum).

[2] In fact, Land showed that even when two different color patches under different illuminations produce signals with identical composition of wavelengths, subjects are able to discern between the two colors and correctly identify them.

[3] Usually, digital cameras offer in addition to the auto white balancing mode, several preset modes such as such as "Daylight", "Cloudy", "Sunny", "Tungsten", "Fluorescent" etc., and sometimes also a manual-calibration mode, in which the user can specify which region of the image is white or gray.

from the image information $p_i(x,y)$, and correct for it. A common assumption made by many such algorithms (and by all algorithms that we review here) is that the illumination has constant hue over the entire field of view of the camera. This leads to a simplification of (2) - we are only interested in estimating and correcting for the three R,G,B components of the illumination as captured by the camera[4]

$$l_i = \int_\Lambda l(\lambda)\, c_i(\lambda)\, d\lambda,\ \ i \in \{R,G,B\}. \tag{3}$$

Throughout this paper, we will follow this simplification and concern ourselves only with estimating the three R,G,B components that result from projecting of the illumination on the camera's three sensitivity functions. We will refer to these as the illumination R,G,B components and use the vector notation $\mathbf{l} = \begin{bmatrix} l_R & l_G & l_B \end{bmatrix}$. We will further assume that the vector is normalized i.e. $\|\mathbf{l}\| = 1$ (that is, the R,G,B components of the illumination are coordinates in rgb-chromaticity space).

Even after this simplification, this problem is in general ill-posed - for a given input $p_i(x,y)$, there are many possible solutions of illumination and surface-reflectance pairs that can explain the input. Color constancy algorithms add constraints to the problem, by posing assumptions on the type of illumination (e.g. spatial smoothness) or on the captured scene.

In this work, we propose a new color balancing algorithm for cameras. Our algorithm estimates the illumination R,G,B components by calculating a luminance-weighted average of pixels with (x,y) chromaticities which lie in an ellipsoid in xy-chromaticity space, around the neutral point. This ellipsoid is learned from a large database of natural images. We compare our algorithm's performance to six other methods for color balancing which are available in the literature - gray-world, max-RGB, shades of gray [3], gray-edge, max-edge [1] and color by correlation [5]. For the purpose of performance comparison, and for studying the range of illuminations and reflectances in natural scenes, we use Ciurea and Funt's large database of 11,346 images of various scenes, collected using a video camera that has a gray sphere attached to it which appears in the bottom-right side of the field of view [2]. By measuring the RGB values of the gray sphere in each scene, it is possible to extract the illumination R,G,B components that can be used for comparison and calibration.

The rest of the article is organized as follows - in Section II we briefly discuss the preferred way of combining a color balancing block in a camera's ISP pipeline ; in Section III we review the six reference algorithms ; in Section IV we describe the proposed method in detail ; and in Section V we present experimental results.

[4]In [19], DiCarlo and Wandell present a method for estimating higher dimensional spectral information from low-dimensional camera responses. As we are interested in estimating the illumination solely for the purpose of color-balancing an image, we settle the three-dimensional model.

II. Color Balancing as Part of the Camera's ISP Pipeline

Most color balancing algorithms used in digital cameras can be viewed as a 2-step operation - the first step is estimating the illumination in the scene, and the second step is applying the correction on the image. The estimation step has been the focus of much research. Two models have been proposed for the correction step. The first suggests applying a fixed diagonal transformation on the R,G,B values at each pixel as in (4) - this model is based on von Kries's coefficient rule [11], who proposed it as a model for the human color-adaptation mechanism. It is widely used in color-constancy algorithms. The second, does not restrict the transformation to be diagonal as shown in (5) - as shown by Finlayson et al in [12], using a linear combination of the R,G,B values can assist in dealing with non-orthogonal camera sensitivity functions (a common case for cameras based on color filter arrays).

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} \Gamma_R & 0 & 0 \\ 0 & \Gamma_G & 0 \\ 0 & 0 & \Gamma_B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} \Gamma_{RR} & \Gamma_{RG} & \Gamma_{RB} \\ \Gamma_{GR} & \Gamma_{GG} & \Gamma_{GB} \\ \Gamma_{BR} & \Gamma_{BG} & \Gamma_{BB} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5}$$

We argue that when combining a color balancing algorithm in a standard camera's ISP pipeline, which incorporates a sensor with a BAYER color filter array [15], there are advantages to overall image quality in separating the correction stage into two parts - the first is a diagonal transformation, which should be applied directly on the RAW BAYER data, before the demosaicing algorithm performs color interpolation, and the second one is a more general linear transformation, which is applied after the demosaicing stage (when all pixels have R,G,B information) in order to correct for color cross-talk and obtain better color reproduction. The estimation stage of the algorithm, in which the illumination's R,G,B components are calculated, can be carried out at any stage of the ISP, before or after[5] correction has been applied on the image, and feed the estimated illumination values to the two correction engines, to be applied on subsequent frames in a closed feedback loop manner. Figure 1 illustrates the structure of a suggested ISP pipeline. The reason for this separation is that a demosaicing algorithm can benefit greatly from a color-balanced image by correctly resolving ambiguity in interpolating fine high-frequency details. As an example, consider a monochromatic high-frequency object under white illumination, captured by the camera as part of a larger scene - Figure 2 shows the image of such an object, made of vertical stripes spread apart to match the

[5]It may be more convenient to perform the estimation of the illumination R,G,B components after the demosaicing stage. In this case, as the coefficients in (4) and (5) which are used in the correction stages are known to the estimation block and as both corrections are invertible, it is easy for the estimation stage to account for corrections applied in earlier stages.

sensor's Nyquist frequency.[6] When sampled by the BAYER sensor, each pixel samples only a portion of the spectrum, defined by the spectral response and the spatial arrangement of the color filter array.[7] Such color sampling strongly

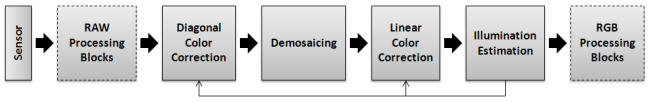

Figure 1. Implementing a color balancing algorithm as part of an ISP pipeline - proposed structure

Figure 2. Vertical stripes object under white illumination (left), and the same pattern sampled by a BAYER color filter array with the resulting 8-bit pixel values (right) .

couples luminance information with color information. A demosaicing algorithm which assumes smooth, slow spatially-varying chrominance and allows for fast-varying luminance (a common assumption of demosaicing algorithms, e.g. [16]) would be able to interpolate the pixels correctly along the stripes direction and resolve the high frequency information, by comparing the green pixels values to the values of the red/blue pixels. However, if the input image is not color-balanced (for instance the object is illuminated by blue light, which is not corrected for before the demosaicing algorithm), it may not be possible for the demosaicing algorithm to correctly recover the high-frequency information - in fact, it may favor the horizontal direction of interpolation, as shown in Figure 3, since it's locally better explained by the data. Balancing the colors before the demosaicing algorithm (by means of diagonal transformation) will result in this case in a correct interpolation of the data. We assume that the coefficients of the general linear transformation (5) can be optimized by characterizing the sensor and calibrating the matrix under different types of illuminations. Due to the lack of sufficiently large database of RAW images coupled with ground-truth illumination R, G, B components information, we did not verify our proposed correction scheme in actual experiments, however, we plan to validate it in future work. In the rest of the paper, we focus on estimating the illumination's R, G, B components and focus on diagonal correction by simply normalizing the three color channels

[6]Note that the contrast is not very high, as expected in such high frequencies when taking into account the performance of a typical optical system - the optical transfer function acts like a low-pass filter, reducing contrast at very high frequencies.

[7]For simplicity, we assume filters with similar transmission/absorption properties at each filter's transmission/absorption areas.

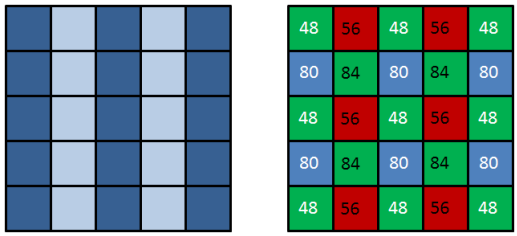

Figure 3. The same vertical stripes object of Figure 2, illuminated by blue light with relative R, G, B spectral components of $\begin{bmatrix} 0.4 & 0.6 & 1 \end{bmatrix}$ (left) and the resulting BAYER 8-bit pixel values (right). Clearly, the pixel values (wrongly) suggest that lines should be interpolated in the horizontal direction.

to the maximum channel

$$\begin{aligned} \hat{\mathbf{l}} &= \begin{bmatrix} \hat{l}_R & \hat{l}_G & \hat{l}_B \end{bmatrix}^T \\ \hat{\Gamma}_i &= \frac{max\left(\hat{\mathbf{l}}\right)}{\hat{l}_i}, \; i \in \{R, G, B\}. \end{aligned} \tag{6}$$

III. Review of Existing Fast Color Balancing Approaches

As AWB algorithms in cameras typically run in viewfinder frame rate and continuously perform the illumination estimation according to input frames (before actually pressing the shutter button to take the image), they have strict limitations on run time and complexity. In addition, they should be able to handle the wide range of natural scenes and illuminations that people regularly encounter - this is still an open problem and today cameras offer, in addition to the AWB mode, several preset white-balancing modes and also manual-calibration modes, in which the user can specify which region of the image is gray.

In this section, we review six fast color balancing algorithms that are available in literature. All of these algorithms estimate the R, G, B components of the illumination vector, projected on the camera's sensitivity functions $\hat{\mathbf{l}} = \begin{bmatrix} \hat{l}_R & \hat{l}_G & \hat{l}_B \end{bmatrix}^T$ by making additional assumptions on the scene and/or light source. The image is then corrected by calculating gain factors $\hat{\Gamma}$ for each of the three color channels according to (6).

The gray-world algorithm, which is based on a model first suggested by Buchsbaum [10] as an explanation of the human visual system's color constancy property, assumes that the average reflectance of a scene, as captrued by the camera is gray. As Buchsbaum's model required additional assumptions on the light source, camera sensitivity functions and on object reflectance,[8] we follow [3] and [1] and apply a somewhat stronger assumption on the scene - that the average reflectance over the entire camera's field of view has a flat spectrum. This assumption leads to the conclusion that the 3×1 vector of average R, G, B colors captured by the camera, corresponds to the projection of the

[8]For example, Buchsbaum's model assumes that both the light source spectrum and the object reflectance can be written as a linear combination of three basis functions of wavelength.

light source's spectrum on the camera sensitivity functions, $\mathbf{l} = \begin{bmatrix} l_R & l_G & l_B \end{bmatrix}^T$. The gray-world algorithm is given in Algorithm 1: We note that when the scene indeed satisfies

Algorithm 1 Gray World Algorithm

a) Calculate the average value of each color channel $i \in \{R, G, B\}$ according to $e_i = \frac{1}{MN} \sum_x \sum_y p_i(x, y)$.
b) Normalize the $\mathbf{e} = \begin{bmatrix} e_R & e_G & e_B \end{bmatrix}^T$ vector to obtain an estimation of the R, G, B components of the illuminant $\hat{\mathbf{l}} = \frac{\mathbf{e}}{\|\mathbf{e}\|}$.

this assumption (and when the illumination is uniform over the field of view), the gray-world estimation is unbiased.

One additional algorithm called max-RGB, is based on Land's explanation of the mechanism responsible for color constancy in the human visual system. In [4], Land proposed that our visual system achieves color constancy by detecting the area of highest reflectance in the field of view, separately for long (red), medium (green) or short (blue) wavelengths (corresponding to the three types of cones in our eyes), and normalizes the response of each cone by the highest value. Similarly, the max-RGB algorithm calculates the maximum in each color channel and normalizes the pixels in each channel according to the maximal value. Note that it is not required for the maximal values of each color to appear at the same spatial location. The algorithm is given in Algorithm 2. The max-RGB algorithm should produce accurate results

Algorithm 2 Max-RGB Algorithm

a) Calculate the maximum value of each color channel $i \in \{R, G, B\}$ according to $e_i = \max_{x,y} (p_i(x, y))$.
b) Normalize the $\mathbf{e} = \begin{bmatrix} e_R & e_G & e_B \end{bmatrix}^T$ vector to obtain an estimation of the R, G, B components of the illuminant $\hat{\mathbf{l}} = \frac{\mathbf{e}}{\|\mathbf{e}\|}$.

when the scene contains a white patch, which reflects the entire spectrum of light evenly, or when the maximal object reflectance is the same for the R, G, B color channels. When implementing max-RGB care should be taken to make sure that the chosen maximal values accurately represent reflectance information from the scene and are not clipped or saturated due to the camera's limited dynamic range. It is usually a good practice to ignore pixels above a certain threshold level (e.g. 95% of dynamic range).

The gray-world and max-RGB algorithms were generalized by Finlayson and Trezzi in [3]. They proposed a color constancy algorithm which is based on Minkowski norm - for each color channel, the Minkowski p-norm is calculated and the normalized result forms the estimated illumination vector. In their setting, the gray world algorithm is obtained by setting $p = 1$, while max-RGB is the result of $p = \infty$. Their algorithm, labeled shades of gray, is given in Algorithm 3. Finlayson and Trezzi concluded that using Minkowski norm with $p = 6$ gave the best estimation results on their data set.

Algorithm 3 Shades of gray algorithm

a) Calculate the normalized Minkowski p-norm of each color channel $i \in \{R, G, B\}$ according to

$$e_i = \left(\frac{1}{MN} \sum_x \sum_y p_i(x, y)^p \right)^{1/p}$$

.
b) Normalize the $\mathbf{e} = \begin{bmatrix} e_R & e_G & e_B \end{bmatrix}^T$ vector to obtain an estimation of the R, G, B components of the illuminant $\hat{\mathbf{l}} = \frac{\mathbf{e}}{\|\mathbf{e}\|}$.

Another method, recently proposed by Van De Weijer et al [1], further generalizes the work of Finlayson and Trezzi in that it also considers the image derivatives (first and second order) instead of the image itself. Their assumption is based on the observation that the distribution of derivatives of images forms an ellipsoid in R, G, B space, of which the long axis coincides with the illumination vector. The derivatives Two notable algorithms they propose are gray-edge, which assumes that the Minkowski 1-norm of the derivative of the image is a-chromatic, and max-edge which uses the infinity-norm instead (see Algorithm 4).

Algorithm 4 Gray-edge and max-edge algorithms

a) Convolve the image (each color channel separately) with a Gaussian filter with standard deviation σ, to produce a σ-scaled image $p_i^\sigma = p_i \otimes G^\sigma$.
b) Calculate the Minkowski p-norm of each color channel of the first order derivative of the scaled image $i \in \{R, G, B\}$ according to

$$e_i = \left(\sum_x \sum_y \left(\sqrt{\left(\frac{\partial}{\partial x} p_i^\sigma \right)^2 + \left(\frac{\partial}{\partial y} p_i^\sigma \right)^2} \right)^p \right)^{1/p}.$$

Take $p = 1$ for gray-edge and $p = \infty$ for max-edge.
c) Normalize the $\mathbf{e} = \begin{bmatrix} e_R & e_G & e_B \end{bmatrix}^T$ vector to obtain an estimation of the R, G, B components of the illuminant $\hat{\mathbf{l}} = \frac{\mathbf{e}}{\|\mathbf{e}\|}$.

Lastly, we review the color-by-correlation algorithm, proposed by Finlayson et al [5]. This algorithm differs from the other algorithms mentioned here in that it puts constraints on both the reflectance and the illumination. Finlayson et al adopted a probabilistic approach, by trying to maximize the likelihood of an illuminant over a set of possible illumination sources, given the observed color gamut in the input scene (assuming a-priori uniform distribution of illuminations and object reflectance). The general outline of the algorithm is given in Algorithm 5 (for a detailed description, see [5]). It is divided into two main stages - a calibration stage, in which an illumination-likelihood matrix is built and an analysis stage, in which the input image is analyzed, its color gamut is correlated with the likelihood matrix and the scene illumination is estimated. An implicit assumption

of the color-by-correlation algorithm is that the input scene contains a wide range of reflectances which corresponds to the set of reflectances used in the calibration stage.

Algorithm 5 Color-by-correlation algorithm

a) Calibration stage

1. Choose a wide set of L possible illuminations which spans natural illumination conditions. Store the illumination R, G, B values (the projection of the illumination on the camera's sensitivity functions) for each of the L illuminations.
2. Characterize the color gamut in xy-chromaticity space, observable by the camera under each of the illuminations. Typically, this stage requires imaging a set of objects with a wide range of reflectance characteristics, distributed so to match the distribution of reflectances in real world, under each type of illumination. The color gamut is represented by uniformly quantizing the xy-chromaticity space to $n \times n$ bins (we chose $n = 24$) and calculating the relative frequency with which an object in the set contributes to each chromaticity bin.
3. Build and store a likelihood matrix $\mathbf{M} \in [0 \dots 1]^{n^2 \times L}$ of the frequency with which chromaticities (coordinates in a quantized xy-chromaticity space) appear under each illumination. Each column of the matrix respresents a different illumination, and its rows are the column-stacked chromaticities frequencies, for each indices pair (x, y).

b) Analysis stage

1. Given an input image, generate $h(x, y)$ a (quantized) 2D histogram of (x, y) chromaticities. Create a column-stacked vector from h.
2. Threshold h by applying to create a column-stack binary vector $\mathbf{i} \in [0, 1]^{n^2 \times 1}$, $\mathbf{i}(j) = \begin{cases} 1 & h(j) > 0 \\ 0 & o.w. \end{cases}$ i.r. $\mathbf{i}$ contains the chromaticities values that appear in the image.
3. Correlate $\mathbf{i}$ with each column of the illumination matrix and find the column which gives the highest score. The correlation vector $\mathbf{c}$ is calculated by multiplying $\mathbf{i}$ with $\mathbf{M}$ to $\mathbf{c} = \mathbf{i}^T\mathbf{M}$, and the index in $\mathbf{c}$ for which the correlation value is the highest is the estimated type of illumination $j = \underset{j \in [1 \dots L]}{\text{argmax}}(\mathbf{c})$. The matching R, G, B values of the illumination are then read from a look-up table and the estimated illumination vector $\hat{\mathbf{l}} = \begin{bmatrix} \hat{l}_R & \hat{l}_G & \hat{l}_B \end{bmatrix}^T$ is formed.

IV. The Proposed Color Balancing Algorithm

We describe a new method for estimating the illumination's R, G, B components, that can be implemented as part of a camera ISP pipeline. Our method is data-driven, derived from our analysis of natural illuminations based on large database of images [2]. We begin by analyzing the color gamut of illuminations - Figure 4 shows a scatter plot of the color gamut in xy-chromaticity space of natural illuminations. We observe that the range of natural illuminations is compact, and that it matches a part of the color temperature curve (red line), from 2700K to 9000K. Given the color gamut of an input image, we can use this

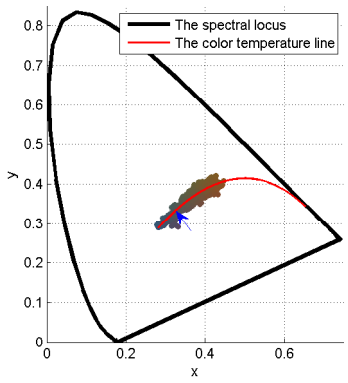

Figure 4. Color gamut of natural illuminations. The blue arrow marks the neutral point $(x, y) = (0.333, 0.333)$.

prior information to isolate the pixels that may correspond to gray in the image - these are simply the pixels whos (x, y) chromaticities values lie inside the area defined by the illuminations color gamut. We approximate this area by finding the minimum-area ellipse that covers all these point in xy space (the Löwner-John ellipse). We find this ellipse by solving an optimization problem (see [14], pp. 410-411) - we denote by $\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ the coordinate vector of the i-th illumination in xy-chromaticity space, and parametrize the ellipse as $\mathcal{E} \triangleq \left\{ \mathbf{p} \in [0 \dots 1]^2 \mid \|\mathbf{A}\mathbf{p} + \mathbf{b}\|_2 \leq 1 \right\}$, where $\mathbf{A} \in \mathbf{S}^2_{++}$, $\mathbf{b} \in \mathbf{R}^2$. The area of the ellipse is proportional to $\det\left\{\mathbf{A}^{-1}\right\}$, so we can define the following convex optimization problem

$$\begin{aligned} &\text{minimize} && \text{logdet}\left\{\mathbf{A}^{-1}\right\} \\ &\text{subject to} && \|\mathbf{A}\mathbf{p}_i + \mathbf{b}\|_2 \leq 1, \; i = 1, \dots, N. \end{aligned}$$

This problem is readily solved by using standard convex solvers (as the dimension of $\mathbf{p}_i$ is 2 and N is not very large), yielding the following parameters for $\mathcal{E}$

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 17.74 & -10.17 \\ -10.17 & 22.06 \end{bmatrix} \\ \mathbf{b} &= \begin{bmatrix} -2.86 \\ -4.05 \end{bmatrix}. \end{aligned}$$

Figure 5 shows the borders of the resulting Löwner-John ellipse.

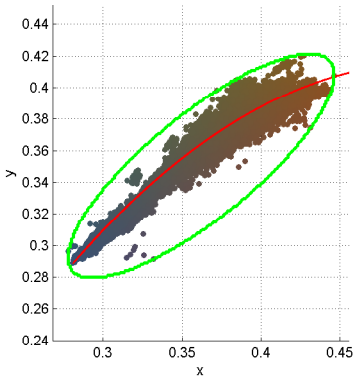

Figure 5. Borders of the minimum-area covering ellipse around the illuminations color gamut

In addition, we can use the illumination information to derive reasonable lower and upper bounds on the R, G, B components of our illumination estimator - we set those to be two standard deviations above/below the mean value of each color component,

$$\begin{aligned} \bar{R} - 2\sigma_R \leq \quad \hat{R} \quad \leq \bar{R} + 2\sigma_R \\ \bar{G} - 2\sigma_G \leq \quad \hat{G} \quad \leq \bar{G} + 2\sigma_G \\ \bar{B} - 2\sigma_B \leq \quad \hat{B} \quad \leq \bar{B} + 2\sigma_B, \end{aligned} \tag{7}$$

where

$$\begin{aligned} \bar{X} &= \frac{1}{N}\sum_{i=1}^{N} X_i \\ \sigma_X &= \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(X - \bar{X}\right)^2}. \end{aligned} \tag{8}$$

Calculating the mean and standard deviation on the illuminations' R, G, B components, we find that $\left[\begin{array}{ccc} \bar{R} & \bar{G} & \bar{B} \end{array}\right] = \left[\begin{array}{ccc} 0.363 & 0.338 & 0.299 \end{array}\right]$ and $\left[\begin{array}{ccc} \sigma_R & \sigma_G & \sigma_B \end{array}\right] = \left[\begin{array}{ccc} 0.0723 & 0.0097 & 0.0749 \end{array}\right]$. Note that g component has a mean which is the closest to the neutral value of 0.333 and the lowest standard deviation of the three chromaticities. This property, together with the higher sensitivity property of the green color channel in cameras [15], makes the green value at each pixel a good estimator of the luminance (or reflectance brightness) in the scene.

Our proposed method for estimating the illumination is composed of three steps. It is summarized in Algorithm 6. Firstly, we extract the pixels that may correspond to gray from the image. We build a 3D color histogram of unique pixel values in the input image, quantized into 64 bins to avoid small fluctuations in color due to noise. Only unique pixel values are considered - disregarding repetitions avoids one of the drawbacks of gray-world type of algorithms, which may be biased by a presence of large color surfaces in the image. Next, from the set of unique pixels we extract only those who lie within the ellipse $\mathcal{E}$ in xy-chromaticity space. These pixels are possible candidates for gray pixels (their hue may have changed by the illumination). Second, we calculate a sum of the R, G, B components of the pixels, each wighted by an estimator for the luminance level of the pixel, which is the pixel's G component. Such weighting which emphasizes pixels with higher luminance values, represents a compromise between gray-world (uniform weighting) and max-RGB (considering only maximum values) approaches, and is more robust than either. After normalizing the weighted-sum vector we get an initial estimation for the R, G, B components of the illumination. Finally, we make sure each of the initial illumination estimator components are within a plausible range, defined in (7) - if needed, the values are clipped to the limits and the illumination vector is normalized again.

V. Experimental Results

In this section we present experimental results, obtained by testing the algorithms described in Section III and the proposed method on a large database of 11346 images (each is 240×360 pixels), captured by Ciurea and Funt [2], using a Sony VX2000 3-CCD video camera. A gray sphere was mounted on the camera, which appears on the bottom right side of the field of view. By measuring the mean R, G, B values on the gray sphere in every image, the vector $\mathbf{l}$ of the illumination's R, G, B components was extracted and normalized to obtain ground truth r, g, b chromaticities values. The camera's AWB algorithm was disabled (an "outdoors" preset was chosen as a white balancing scheme). Figure 6 shows a few samples from the database.

Figure 6. Sample images from the database. The gray sphere is visible on the bottom right part of the image

We implemented the gray-world, max-RGB, shades of gray, gray-edge, max-edge and color by correlation algorithms and the proposed method in Matlab environment. In all algorithms, we excluded the gray sphere area from the image data available for estimation. In max-RGB, we discarded pixels above a threshold of 242 (95% of dynamic range), to avoid non-linearity due to signal clipping. In shades of gray, we used a Minkowski norm value of $p = 6$.

Algorithm 6 Proposed illumination estimation algorithm

a) Extract candidates for gray
   1. Build a quantized R,G,B histogram of unique pixels in the input image $\mathbf{F}$. Each pixel's 8-bit color components are quantized according to $\tilde{p}_i = 4\,[p_i/4]$ to create $\tilde{\mathbf{F}}$. Each bin of the histogram gets a value of 1 if there exist pixels in $\tilde{\mathbf{F}}$ that maps to that bin, and 0 otherwise. Extreme values (below 2% or above 98% in $\mathbf{F}$) are ignored. The unique R,G,B values are then extracted from the histogram to form a $K \times 3$ matrix, $\mathbf{G}$.
   2. Calculate the (x,y) chromaticities of pixels in $\mathbf{G}$. We right-multiply $\mathbf{G}$ with the standard $RGB \rightarrow XYZ$ transform matrix
   $$\mathbf{T} \triangleq \begin{bmatrix} 0.49 & 0.31 & 0.2 \\ 0.17697 & 0.8124 & 0.01063 \\ 0 & 0.01 & 0.99 \end{bmatrix}^T,$$
   and normalize to obtain the xyz coordinates matrix $\mathbf{J}$ $\mathbf{J}_n = \left(\sum \mathbf{G}_n\mathbf{T}\right)^{-1}\mathbf{G}_n\mathbf{T}$. We then extract only the (x,y) chromaticities from $\mathbf{J}$ by calculating $\mathbf{J}_{xy} = \mathbf{J}\mathbf{Q}$, where
   $$\mathbf{Q} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T.$$
   3. Extract all the pixels in $\mathbf{G}$ with (x,y) chromaticities that lie inside the ellipse $\mathcal{E}$. This is done simply by finding all the rows in $\mathbf{J}_{xy}$ for which $\left\|\mathbf{A}\left(\mathbf{J}_{xy}\right)_i^T + \mathbf{b}\right\|_2 \leq 1$ and extracting the corresponding rows in $\mathbf{G}$. Denote the resulting $\tilde{K} \times 3$ matrix $\tilde{\mathbf{G}}$. If there are too few such pixels (below 512), all pixels in $\mathbf{G}$ are used.

b) Calculate a weighted sum of the R,G,B values in $\hat{\mathbf{G}}$, weighting each of them with the G value as an estimate for the luminance $\hat{\mathbf{e}} = \begin{bmatrix} \sum_{i=1}^{\tilde{K}} G_i R_i & \sum_{i=1}^{\tilde{K}} G_i^2 & \sum_{i=1}^{\tilde{K}} G_i B_i \end{bmatrix}$, and normalize the resulting vector to obtain an estimation for the illumination R,G,B components $\hat{\mathbf{l}} = \frac{\hat{\mathbf{l}}}{\|\hat{\mathbf{l}}\|}$.

c) Limit the illumination components according to the limits defined in (7). Normalize again after limiting.

Gray-edge and max-edge were implemented using Sobel masks for gradient estimation

$$\begin{aligned} f_x &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \\ f_y &= f_x^T, \end{aligned} \tag{9}$$

and parameters $p = 1$ and $s = 6$ (following notation in [1]). Of the six algorithms, only Color-by-correlation required additional reflectance and camera sensitivity data for calibration. We used calibration data from Simon Fraser University [13] for that purpose, despite the fact that the camera sensitivity functions are given for a different type of Sony camera (the DXC-930 model). We justify the use of this data for our purposes by noting that both cameras use a 3-CCD mechanism to capture color - a dichroic prism splits incoming light into three primary R,G,B wavelengths, and each is sampled independently by a dedicated CCD sensor (hence the 3-CCD). The prism's red, green and blue dichroic coatings have very sharp spectral responses, as shown in Figure 7, compared to those of a standard BAYER color filter array [18].[9] Therefore 3-CCD cameras enjoy significant reduction in color-crosstalk and a more accurate color reproduction than cameras with one BAYER CCD. As the center of each spectral curve of color coatings in

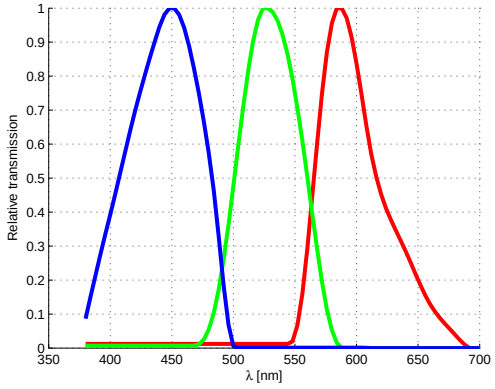

Figure 7. Spectral response of the dichroic coatings in the Sony DXC-930 camera

cameras is relatively fixed (around $450\,[nm]$, $530\,[nm]$ and $600\,[nm]$ for B, G, and R respectively) to match that of the human visual system [17], and due to their narrow shape, the sensitivity functions of the DXC-930 should provide a reasonable approximation of the actual sensitivity functions of the VX2000 model.

Initially, we calculated the camera's response to 1995 objects with different reflectance characteristics, under 598 types of illuminationwhich represent a very wide range (most of them are very different from natural illuminations). From this database, we created the likelihood matrix $\mathbf{M}$ as described in Algorithm 5, and in addition we calculated the camera response for each type of illumination (to serve as the estimate for the r,g,b chromaticities vector of the illuminant). However, using such a wide range of illumination produced poor estimation results.[10] We then discarded all illuminations with (x,y) chromaticities that fall outside the ellipse described in Section IV (i.e. artificial illuminations) and repeated the estimation process with the constrained illumination probability matrix (results are described below).

For each of the algorithms, we calculated the estimated illumination vector $\hat{\mathbf{l}}$ of R,G,B components, normalized it

[9] The BAYER color filter array is more common in cameras mainly due to cost and form-factor considerations.

[10] A mean error of 18.27 degrees, median error of 16.08 degrees, maximal error of 39.47 degrees and a standard deviation of 9.34 degrees.

and compared it to the ground truth r, g, b chromaticites values for every image. We used the common angular error metric in the rgb-chromaticity space, to measure how accurate the estimation is. The angular error, measured in degrees, is defined as

$$E \triangleq cos^{-1}\left(\frac{\mathbf{l}\cdot\hat{\mathbf{l}}}{\|\mathbf{l}\|\left\|\hat{\mathbf{l}}\right\|}\right), \quad (10)$$

so a value of 0 degrees means perfect reconstruction. We measured the angular error on the entire database of 11346 images and calculated the mean, median, and maximal errors, as well as the standard deviation. We also tested the algorithms of Section III when applying the maximum and minimum illumination constraints (defined in (7)) on their outputs[11]). The results are summarized in Table I. Our proposed method achieved lower mean and median errors, and also the lowest standard deviation of the various algorithms. It came out behind max-RGB in terms of maximal error. Applying the maximum and minimum constraints on the estimation improved the results in all cases. Figure 8 shows the mean and standard deviation of the various algorithms (constrained version), and in Figure 9 we present a histogram of the errors for each algorithm (again, constrained version) - each histogram shows the distribution of errors between $0°$ and $40°$, with a bin width of 0.5 degrees. The error histograms reveal interesting properties

	Mean error	Std	Median error	Max Error
Gray-world	7.3°	5.13°	6.28°	42.63°
Constrained gray-world	6.71°	4.68°	6.28°	42.63°
Max-RGB	7.86°	6.69°	6.22°	27.41°
Constrained max-RGB	7.84°	6.69°	6.22°	**27.41°**
Shades of gray	6.67°	4.53°	5.83°	35.66°
Constrained shades of gray	6.55°	4.52°	5.83°	35.66°
Gray-edge	6.27°	4.51°	5.32°	34.02°
Constrained gray-edge	6.13°	4.44°	5.32°	34.02°
Max-edge	8.39°	6.24°	6.72°	36.32°
Constrained max-edge	8.12°	6.24°	6.72°	36.32°
Color by correlation	10.24°	9.03°	7.86°	38.87°
Proposed	**5.5°**	**4.02°**	**4.47°**	30.51°

Table I
COMPARISON OF DIFFERENT COLOR BALANCING ALGORITHMS. VALUES IN DEGREES.

of the different algorithms. For gray-world, shades of gray, gray-edge, max-edge and the proposed method, the error distribution has a similar shape, with a significant part of the errors in the range of 0.5-5 degrees, and a long tail,

[11]We do not give the constrained version of color by correlation as these constraints are already fulfilled when building the probability matrix by including only plausible illuminations.

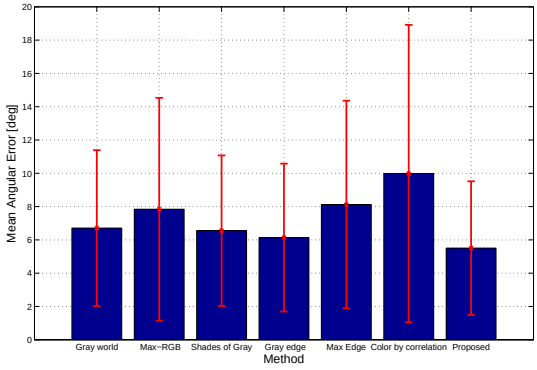

Figure 8. Mean value and error (2 standard deviations) of the various algorithms

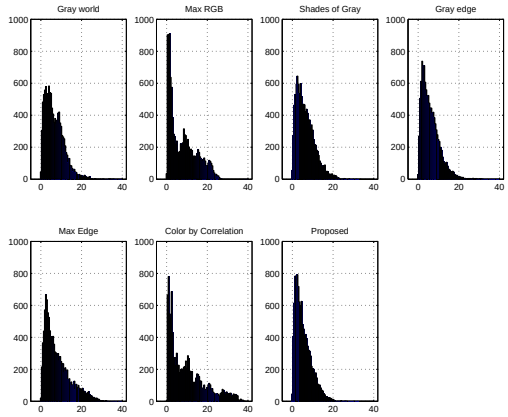

Figure 9. Histogram of errors for the various algorithms

up to an error of about 30 degrees with varying weight. The max-RGB algorithm shows very good performance on a fairly large amount of scenes (more that 2750 images with angular error below 1.75 degrees), but it also suffers from larger errors in many images (over 3800 images with error above 10 degrees). It also boasts the smallest maximal error, although its standard deviation is the second largest. Color by correlation shows interesting properties as well - for many scenes the illumination estimation is relatively accurate (2268 scenes have an error that is less than 2 degrees), however for a many scenes the error is very large (490 images with error above 30 degrees). Analyzing the color gamut of the illuminations that led to high errors (30 degrees and above) in color by correlation, shown in Figure 10, reveals that these errors correspond to illuminations with low color temperature, between 2750K and 3500K. Analyzing the proposed method in a similar manner (Figure 11) shows that while small errors (below 3 degrees) are distributed quite evenly on the range of natural illuminations, high errors (above 20 degrees) are more characteristic of illuminations with chromaticities relatively far from the neutral point. Selected examples of images after color balancing with each

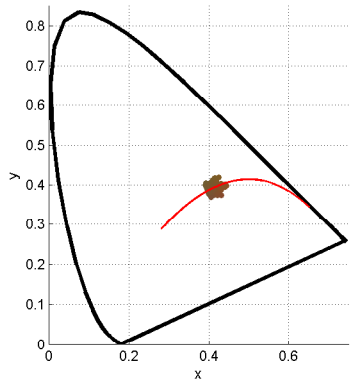

Figure 10. Color by correlation - gamut of illuminations that resulted in errors of 30 degrees and above

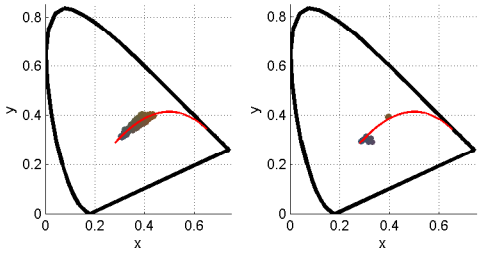

Figure 11. Proposed algorithm - gamut of illuminations that resulted in errors of 3 degrees and below (left) and of 20 degrees and above (right)

of the algorithms discussed herein are given in Figure 12.[12]

The average runtime of the algorithms is given in Table II for a standard 1.67GHz quad-core machine - all meet the timing constraints for real time implementation.

Method	Run time [sec]
Gray-world	$1.2 \cdot 10^{-4}$
Max-RGB	$1.4 \cdot 10^{-4}$
Shades of gray	$4 \cdot 10^{-4}$
Gray-edge	$9.9 \cdot 10^{-4}$
Max-edge	$9.3 \cdot 10^{-4}$
Color by correlation	$4.1 \cdot 10^{-4}$
Proposed	$5.8 \cdot 10^{-4}$

Table II
AVERAGE RUNTIME OF COLOR CORRECTION ALGORITHMS [SEC]

VI. CONCLUSION

We presented a fast, simple algorithm for color balancing, that can be implemented as part of a camera ISP pipeline. It is derived from analyzing the color gamut of natural illuminations and isolating only pixels which may correspond to gray to take part in the averaging process. Experimental results on a very large database of images show that our method improves on the results of existing fast and simple color constancy algorithms. In addition, we proposed a new way of combining color balancing algorithms in cameras ISPs, arguing that there are benefits to separating the correction stage to a diagonal and a general linear correction to be applied before and after the demosaicing block respectively. In future work, we intend to test this proposal on real world data.

An important observation which follows from Figure 12 is that the use of angular error in rgb-chromaticity space, as a metric for evaluating the quality of color balancing algorithms is insufficient. Relatively large values of angular error may correspond to an image which appears very unnatural to the human observer (as in the case of max-edge in the first row) in one case, and to an image which is qualitatively acceptable (max-edge, fourth row) in another case. Moreover, an image with a certain angular error may appear to be more off-balance than an image with a larger error (e.g. compare max-edge and the proposed method in the last row). Our qualitative evaluation of the quality of color-balancing depends not only on the angle between estimated and ground truth illumination vectors, but also on the direction in which the estimated illumination vector points and on the color gamut of objects in the scene. An error metric which penalizes estimations that result in unnatural colors will correspond better to our qualitative evaluation of color balancing, and may also assist in developing better color balancing algorithms for cameras.

[12]Color balancing should preferably be applied on linear R, G, B values, before tone-mapping and other non-linear operations in an ISP pipeline.

REFERENCES

[1] J. Van de Weijer, T. Gevers and A. Gijsenji, "Edge-Based Color Constancy," IEEE Transactions on Image Processing, vol. 16, no. 9, September 2007.

[2] F. Ciurea, and B. Funt, "A Large Image Database for Color Constancy Research," Proceedings of the Imaging Science and Technology Eleventh Color Imaging Conference, pp. 160-164, Scottsdale, November 2003.

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

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

[5] G. Finlayson, S. Hordley, and P. Hubel, "Color by correlation: A simple, unifying framework for color constancy," IEEE Trans. Pattern Anal. Machine Intell., vol. 23, pp. 1209–1221, 2001.

[6] B. A. Wandell, "Foundations of Vision", Stanford: University of Stanford Press, 1995.

[7] D. H. Brainard and B. A. Wandell, "Analysis of the retinex theory of color vision," Journal of Optical Society of America, vol. 3, no. 10, October 1986.

[8] K. Barnard, V. Cardei, and B. Funt, "A Comparison of Computational Color Constancy Algorithms; Part One: Methodology and Experiments with Synthesized Data," IEEE Transactions on Image Processing, vol. 11, no. 9, September 2002.

[9] 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, vol. 11, no. 9, September 2002.

[10] G. Buchsbaum, "A spatial processor model for object colour perception," Journal of the Franklin Institute, vol. 310, July 1980.

[11] J. von Kries, "Beitrag zur Physiologie der Gesichtsempfinding," Arch. Anat. Physiol., 2, pp. 5050-5524, 1878.
[12] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Spectral Sharpening: Sensor Transformations for Improved Color Constancy," Journal of the Optical Society of America A, 11, pp. 1553- 1563, 1994.
[13] K. Barnard, L. Martin, B. Funt and A. Coath, "A Data Set for Colour Research," Color Research and Application, vol. 27, no. 3, pp. 147-151, 2002.
[14] S. Boyd and L. Vandenberghe, "Convex Optimization," Cambridge University Press, 2004.
[15] B. Bayer, "Color imaging array," U.S. Patent 3971065.
[16] De Lavarene, B.C., Alleysson, D. and Herault, J., "Practical implementation of LMMSE demosaicing using luminance and chrominance spaces," CVIU, No. 1-2, pp. 3-13, July 2007.
[17] G. Wyszecki and W.S. Stiles, "Color Science: Concepts and Methods, Quantitative Data and Formulae," Second Ed., Appendix of Extended Tables and Illustrations, John Wiley & Sons, New York, 1982.
[18] P. L. Vora, J. E. Farrell, J. D. Tietz, D. H. Brainard, "Digital color cameras—2—Spectral response," HP Technical Report, March 1997.
[19] J. M. DiCarlo and B. A. Wandell, "Spectral estimation theory: beyond linear but before Bayesian," Journal of Optical Society of America, vol. 20, no. 7, July 2003.

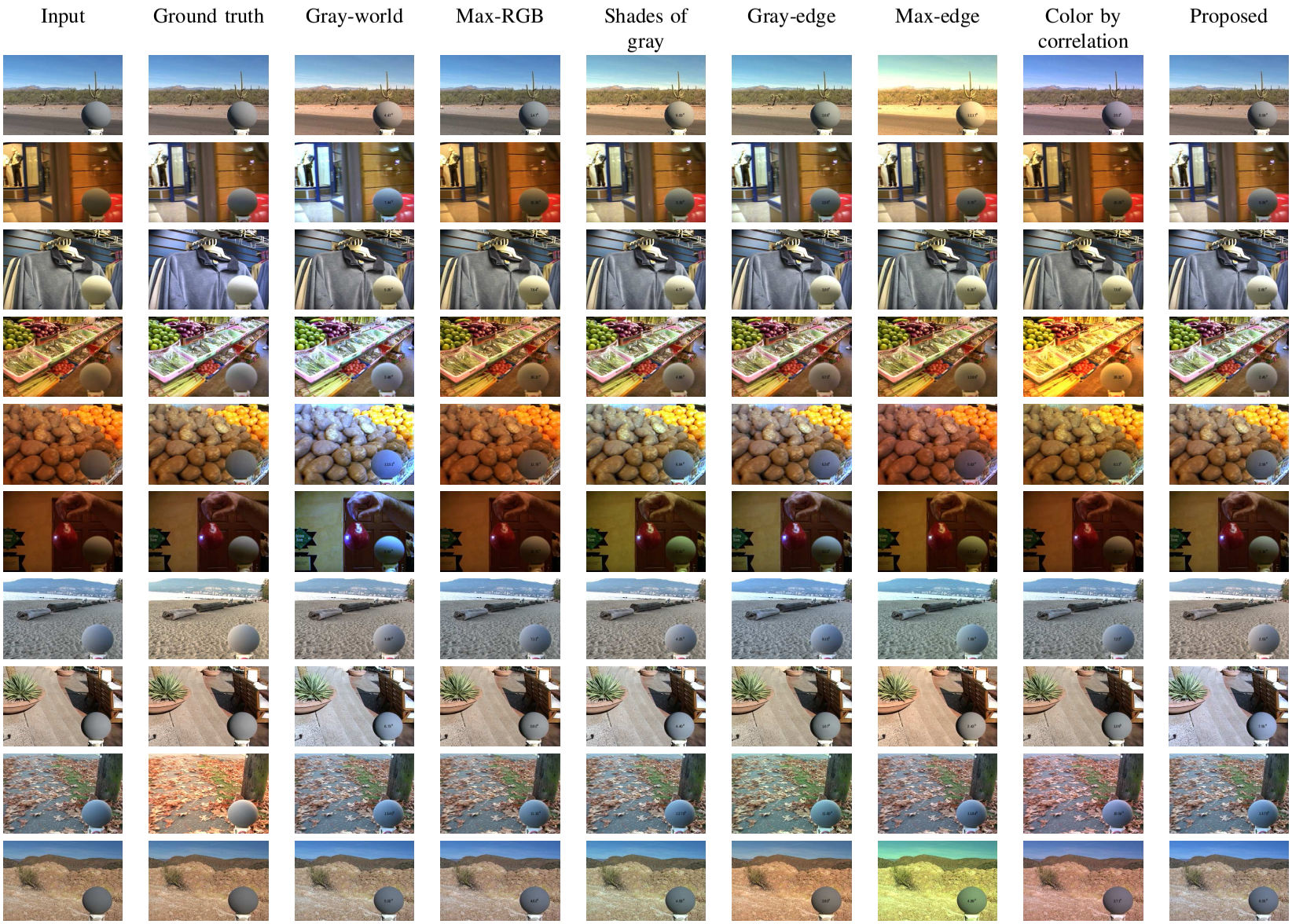

Figure 12. Results of the color balancing algorithms - examples. From left column to right column - input, ground truth, gray-world, max-RGB, shades of gray, gray-edge, max-edge, color by correlation and the proposed method. The angular error in each image is noted on the gray sphere (zoom-in in the electronic version). The first seven rows show a wide range of scenes in which the proposed method improves over the other methods. The last three rows show failures of the proposed method compared to the other methods.