1 Introduction

The problem of separating the illumination from the reflectance information in a given image has been extensively researched in the last three decades, following Edwin Land's seminal work on color vision and his development of the Retinex theory [4]. The problem can be described as follows - given an input image \( S \), we would like to decompose it into two different images - the reflectance image \( R \) and the illumination image \( L \), such that \( S(x, y) = R(x, y) L(x, y) \). There are many benefits to such a decomposition, including the ability to correct for color-shifts due to illumination, correct for uneven illumination, introduce artificial lighting and enhancing dynamic range. It is not hard to see that in general, this problem is ill-posed - for a given input image \( L \), there are infinitely possible solutions of \( L \) and \( R \) pairs that can explain \( S \). Many works have tried to constraint the problem, by posing assumptions on the type of illumination (e.g. constant-hue illumination over the field-of-view and spatial smoothness).

With the growing popularity of digital cameras the importance of fast algorithms for color correction (also known as “auto white-balancing”, AWB in short) grew as well. Such algorithms are an integral part of the image signal processing (ISP) pipeline that is responsible for converting the RAW image captured by the sensor into the final color JPEG image that is saved on the memory card. AWB algorithms try to estimate the correct three white-balance gains (for the red, green and blue channels) that should be applied on an input image in order to correct for color shifts caused by illumination, so that white elements in the scene indeed appear white in the image - similar to the way the human visual system can compensate for different lighting conditions so that white color always seems white under different illuminations. Figure 1 shows an example of correct vs. incorrect color balancing.

Figure 1: Example of correct vs. incorrect color balancing. Left: indoor scene, captured with correct color balancing for incandescent illumination. Middle: outdoor scene, captured with the same color balancing used for the indoor image. The color balance is obviously wrong. Right: Correcting the color balancing for outdoor D65 illumination. Example taken from the BBC film “Colorful Notions” that surveys Edwin Land's work on Retinex.

As AWB algorithms typically run in viewfinder frame rate (i.e. before actually pressing the shutter button to take the image) and continuously adjust according to input frames, they have a very strict limitation 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 (such as “Daylight”, “Cloudy”, “Sunny”, “Tungsten”, “Fluorescent” etc.) and also manual-calibration modes, in which the user can specify which region of the image is white.

2 Project Description

2.1 Goals

In this project we will devise and implement and test a new algorithm for AWB that aims to estimate three gains from an input image. The algorithm will receive as an input an image that was captured under some unknown illumination and will output the color-corrected image along with its estimation of the AWB gains that were applied to each color channel. We will use the large data set of scenes and illuminations to study the impact of lighting on the color gamut of various scenes. We hope that such a study will result in a new, more robust way to estimate the AWB gains for a given input image.

In addition to developing the new algorithm, we will also implement the following fast algorithms that are often used (with some variations) for performing AWB in cameras - for comparison purposes:

a. Gray-World [3].
b. Max-RGB [3].
c. Shades-of-Gray [3].
d. Gray-Edge [1].
e. Max-Edge [1].
f. Color-by-Correlation [5].

It’s important to note that a large family of gamut-mapping algorithms for color correction is not represented here - this is mainly due to computational complexity that is involved in most of these algorithms, which forbids actual implementation in most cameras.

2.2 Comparison Method

For testing the quality of the different algorithms, we will make use a large data set, containing 11346 natural images (each is 240 x 360 pixels - viewfinder resolution) that are used for developing color balancing algorithms, courtesy of the Simon Fraser University (SFU) [2]. The camera used for capturing these images has a gray sphere attached to it in the bottom-right corner of the field-of-view, allowing measurement of the true color composition of the scene illumination. This data set therefore allows measuring the estimation error of the color-balancing algorithms by comparing each of them to the ground truth. Figure 2 shows some example images from the database.

![Figure 2: Example images from the SFU data set](image)

Our error metric will be the angular error in the rgb chromaticities space, between the vector of true illumination and vector of estimated illumination. The angular error is defined as

\[
E \triangleq \cos^{-1} \left( \frac{I \cdot \hat{I}}{\|I\| \|\hat{I}\|} \right),
\]

where \( I = [ R \ G \ B ] \) is the true illumination color vector and \( \hat{I} = [ \hat{R} \ \hat{G} \ \hat{B} ] \) is the estimated illumination color vector, and is measured in degrees.
3 Implementation Notes

Platform: The need for ground truth information to compare the algorithms against, and the fact that there’s no access to the Droid’s camera ISP pipeline, prohibit the use of Droid phone for this project. I will implement the algorithms and test functions in Matlab environment.

References


