UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Facoltà di Scienze Matematiche, Fisiche e Naturali

Dipartimento di Informatica, Sistemistica e Comunicazione

Dottorato di Ricerca in Informatica - XXII Ciclo

Color Correction Algorithms for Digital Cameras

Supervisor: Prof. Raimondo Schettini

Tutor: Prof. Enrico Fagiuoli

Ph.D. Coordinator: Prof.ssa Stefania Bandini

Ph.D. Dissertation of: Simone Bianco

Anno Accademico 2008-2009

To my wonderful parents, who have raised me to be the person I am today. You have been with me every step of the way, through good and bad times. Thank you for all the unconditional love, guidance, and support that you have always given me, helping me to succeed and instilling in me the confidence that I am capable of doing anything I put my mind to. Thank you for everything.

To my soul mate and confidant, for always being there for me. Thank you for your continual love, support, and patience as I went through this journey. I could not have made it through without you by my side.

Acknowledgements

I would like to formally thank Prof. Raimondo Schettini, my supervisor, for his hard work and guidance throughout these years and for believing in my abilities. I have learned so much, and without you, this would not have been possible.

I would also like to thank my friends in the IVL - Imaging and Vision Laboratory, for our exchanges of knowledge and skills, which helped to enrich the PhD experience.

I would like to thank Ing. Filippo Naccari and Ing. Arcangelo Bruna from ST Microelectronics for their help and suggestions.

Finally I would like to acknowledge ST Microelectronics for the financial founding.

Abstract

The image recorded by a digital camera mainly depends on three factors: the physical content of the scene, the illumination incident on the scene, and the characteristics of the camera. This leads to a problem for many applications where the main interest is in the color rendition accuracy of the scene acquired.

It is known that the color reproduction accuracy of a digital imaging acquisition device is a key factor to the overall perceived image quality, and that there are mainly two modules responsible for it: the former is the illuminant estimation and correction module, the latter is the color matrix transformation. These two modules together form what may be called the color correction pipeline.

This thesis has the objective to design and test new and more robust modules for the color correction pipeline, studying and exploiting the existing crosstalks in order to obtain a higher color reproduction accuracy.

The first module considered is the illuminant estimation and correction one; many illuminant estimation solutions have been proposed in the last few years, although it is known that the problem addressed is actually ill-posed as its solution lack uniqueness or stability. To cope with this problem, different solutions usually exploit some assumptions about the statistical properties of the expected illuminants and/or of the object reflectances in the scene. In the last few years two research areas that are important in the context of improving the performance of color constancy algorithms have been highlighted: making additional measurements at the time of image capture (i.e. using more color pixel information), and algorithm combining (i.e.

using two or more estimations of the illuminants). In this thesis a third hypothesis is investigated: the use of low level visual information to improve illuminant estimation.

The second module considered is the transformation of the camera-dependent RGB image data into a standard RGB color space. This transformation, usually called color correction matrix or color matrixing, is needed because the spectral sensitivity functions of the sensor color channels rarely match those of the desired output color space (usually sRGB).

The color correction matrix transformation is usually optimized assuming that the illuminant in the scene has been successfully estimated and compensated for. Both the illuminant estimation process and the color correction matrix concur in the formation of the overall perceived image quality. The two processes have always been studied separately, thus ignoring the interactions between them.

In this thesis the interactions between the illuminant estimation process and the color correction matrix in the formation of the overall color accuracy are investigated, especially when the white point estimation is imperfect. How the color correction transform amplifies the illuminant estimation errors is also investigated. Furthermore, it is shown that it is possible to incorporate knowledge about the illuminant estimation behavior in the optimization of the color correction matrix to alleviate the error amplification. It is demonstrated that a fixed device chromatic response characterization, which is often adopted, is not able to produce good color accuracy in most situations. New strategies to improve color accuracy under both perfect and imperfect white point estimation are proposed, which clearly suggest that adaptive color transformations have to be preferred in order to improve the color accuracy.

Contents

List of Figures

List of Tables

Nomenclature

AE auto exposure

AWB automatic white balance

CAC Classification-based Algorithms Combination

CART classification and regression trees

CAS Classification-based Algorithm Selection

CAT chromatic adaptation transform

CCT correlated color temperature

CDA Class-Dependent Algorithms

CDAUC Class-Dependent Algorithms with Uncertainty Class

CDP Class-Dependent Parameterization

CFA color filter array

CI Class-Independent

CSF contrast sensitivity function

DN do nothing

DoF degrees of freedom

DSC digital still camera

GE1 first-order gray edge

GE2 second-order gray edge

gGW general gray world

GW gray world

HVS human visual system

LMS least mean squares committee

MCC Macbeth ColorChecker

NNM no-N-max

OECF opto-electronic conversion function

PSF point spread function

PSM pattern search method

SG shades of gray

WP white point

WST Wilcoxon sign test

Chapter 1

Introduction

The image recorded by a digital camera mainly depends on three factors: the physical content of the scene, the illumination incident on the scene, and the characteristics of the camera. This leads to a problem for many applications where the main interest is in the color rendition accuracy of the scene acquired.

It is known that the color rendering accuracy of a digital imaging acquisition device is a key factor to the overall perceived image quality Ramanath *et al.* (2005), and that there are mainly two modules responsible for it: the former is the illuminant estimation and correction module, the latter is the color matrix transformation. These two modules together form what may be called the color correction pipeline.

This thesis has the objective to design and test new and more robust modules for the color correction pipeline, studying and exploiting the existing crosstalks in order to obtain a higher color rendering accuracy.

The first module considered is the illuminant estimation and correction one; many illuminant estimation solutions have been proposed in the last few years although it is known that the problem addressed is actually ill-posed as its solution lacks uniqueness or stability Funt *et al.* (1998). To cope with this problem different solutions usually exploit some assumptions about the statistical properties of the expected illuminants and/or of the object reflectances in the scene. Hordley in his review paper Hordley (2006) gives an excellent review of illuminant estimation algorithms and highlighted two research areas that are important in the context of improving the performance of color constancy algorithms: making

additional measurements at the time of image capture (i.e. using more color pixel information), and algorithm combining (i.e. using two or more estimations of the illuminants). In this thesis a third hypothesis is investigated: the use of low level visual information to improve illuminant estimation.

The second module considered is the transformation of the camera-dependent RGB image data into a standard RGB color space. This transformation, usually called color correction matrix or color matrixing, is needed because the spectral sensitivity functions of the sensor color channels rarely match those of the desired output color space (usually sRGB Stokes *et al.* (1996)).

The color correction matrix transformation is usually optimized assuming that the illuminant in the scene has been successfully estimated and compensated for Bianco *et al.* (2007); Hubel *et al.* (1997). Both the illuminant estimation process and the color correction matrix concur in the formation of the overall perceived image quality. The two processes have always been studied separately, thus ignoring the interactions between them.

In this thesis the interactions between the illuminant estimation process and the color correction matrix in the formation of the overall color accuracy are investigated, especially when the white point estimation is imperfect. It is also investigated how the color correction transform amplifies the illuminant estimation errors. Furthermore, it is shown that it is possible to incorporate knowledge about the illuminant estimation behavior in the optimization of the color correction matrix to alleviate the error amplification. It is demonstrated that a fixed device chromatic response characterization, which is often adopted, is not able to produce good color accuracy in most situations. New strategies to improve color accuracy under both perfect and imperfect white point estimation are proposed, which clearly suggest that adaptive color transformations have to be preferred in order to improve the color accuracy.

1.1 Thesis overview

The first part of the thesis introduces the mathematical model of the image formation process and then describes the different modules of which a typical

color image processing pipeline is composed of. A simplified color correction pipeline is introduced and described. These are arguments of chapter 2.

Chapter 3 deals with the base methods, metrics and datasets used in the next chapters of the thesis.

Chapter 4 is dedicated to the description of classification based illuminant estimation algorithms. The chapter starts with the description of the low level features, which can be divided into two main groups: general purpose features and problem-dependent features. The next topic is related to the dataset selection and to the basic and benchmark color constancy algorithms considered. Two new illuminant estimation algorithms are then proposed: the first one exploits a class-based classification, each image is first classified as indoor or outdoor and then processed with the best illuminant estimation algorithm for the predicted class. The second one exploits a feature-based classification without assigning the images to a semantic class: the classifier is trained to assign each considered image with the illuminant estimation algorithm that would give the lowest error in the illuminant estimation. Images are so no longer linked to semantic classes but directly linked to the illuminant estimation algorithms. The chapter ends with the comparison of the proposed algorithms with state of the art ones on a dataset of real images.

Chapter 5 is dedicated to the study of new strategies to increase the color rendition accuracy. It is demonstrated that the use of a single color transformation matrix is not the optimal solution, but different illuminants would need different transformation matrices. To further increase the color rendition accuracy it is shown that it is possible to find color transformation matrices that incorporate the knowledge of how the illuminant estimation algorithm used tends to make errors. This is the first time that the crosstalk between the illuminant estimation algorithms and the color transformation matrices is investigated: in the state of the art these two modules are always studied and optimized independently. The chapter ends with the comparison of the proposed strategies with state of the art ones on a dataset of spectral images acquired with a virtual camera simulator.

Chapter 6 couples the results obtained in chapters 4 and 5 to design and test new color correction pipelines. These are simplified versions of the full pipeline presented in chapter 2, but contain the modules that are responsible for the color

rendering accuracy: the illuminant estimation algorithm and the color correction matrix. All the other modules in the processing pipeline can be divided into two macro-categories: the former is composed of those modules which are dedicated to the correction of the artifact introduced by the non optimality of the sensor (e.g. sensor non linearity correction, noise reduction, etc.); the latter is composed of those modules which are dedicated to the enhancement of the image in order to produce a more pleasant final image when viewed by a human observer.

Finally, chapter 7 ends the thesis summarizing the results obtained, reporting the conclusions and giving the directions for future works.

Chapter 2

Color Image Processing Pipeline

In this chapter a typical digital still camera (DSC) processing pipeline is presented, from the image formation on the sensor to the display and storage of the final image. Inspired by Ramanath *et al.* (2005) and Lukac (2008), each building block is analyzed and described.

2.1 Image Formation

An image acquired by a digital camera can be represented as a function $\boldsymbol{\rho}$ mainly dependent on three physical factors: the illuminant spectral power distribution $I(\lambda)$, the surface spectral reflectance $S(\lambda)$ and the sensor spectral sensitivities $\mathbf{C}(\lambda)$. Using this notation, the sensor responses at the pixel with coordinates (x, y) can be thus described as:

$$\boldsymbol{\rho}(x, y) = \int_{\omega} I(\lambda) S(x, y, \lambda) \mathbf{C}(\lambda) \mathrm{d}\lambda, \tag{2.1}$$

where ω is the wavelength range of the visible light spectrum, $\boldsymbol{\rho}$ and $\mathbf{C}(\lambda)$ are K-component vectors, where K is the number of the spectral bands acquired by the sensor. Digital still cameras (DSC) typically use $K = 3$, although one may use even sensors with more spectral bands. A typical solution is to use an image sensor composed of three different photo-receptors acquiring red, green and blue color components, displaced following the Bayer pattern Bayer (1976). This is the reason why the three-component vector of sensor responses $\boldsymbol{\rho} = (\rho_1, \rho_2, \rho_3)$ is also referred to as the sensor or camera raw $\mathbf{RGB} = (R, G, B)$ triplet. In the

following the convention that **RGB** triplets are represented by column vectors is always adopted.

Since usually $I(\lambda)$, $S(\lambda)$ and $\mathbf{C}(\lambda)$ are sampled uniformly in a spectral range $[\lambda_{min}, \lambda_{max}]$, eq.2.1 can be replaced by its discrete version:

$$\boldsymbol{\rho}(x, y) = \mathbf{S}(x, y)\mathbf{IC}, \tag{2.2}$$

A more complete model could also include the eventual nonlinearity $\mathcal{N}$ introduced by the acquisition system and an additive noise term $\mathbf{n}$:

$$\boldsymbol{\rho}(x, y) = \mathcal{N}\left(\mathbf{S}(x, y)\mathbf{IC} + \mathbf{n}\right), \tag{2.3}$$

Let us denote by f an image with M rows, N columns and K spectral bands. The image can also be considered a two-dimensional (2-D) array with vector-valued pixels. Each vector-valued pixel is formed according to the model in eq.2.3, with values determined by the reflectance and illumination at the three-dimensional (3-D) world point indexed by 2-D camera pixel position. The image formed is then further modeled as follows:

$$\mathbf{g} = \mathcal{B}\left\{\mathbf{Hf}\right\}, \tag{2.4}$$

where $\mathcal{B}$ is a color filter array (CFA) sampling operator, $\mathbf{H}$ is the point spread function (PSF) corresponding to the optical system and $\mathbf{f}$ is a representation of the full-color image in which each pixel is formed according to eq.2.3.

2.2 Processing pipeline

The signal flowchart shown in Fig.2.1 presents a typical DSC processing pipeline. It should be noted that the sequence of operations could substantially differ from manufacturer to manufacturer. Now each block of Fig.2.1 is considered and described.

2.2.1 Sensor, aperture and lens

Although there is the need to measure three (or eventually more) bands at each pixel location, this would require the use of more than one sensor and, consequently, would increase the cost and the dimensions of the camera. As a common

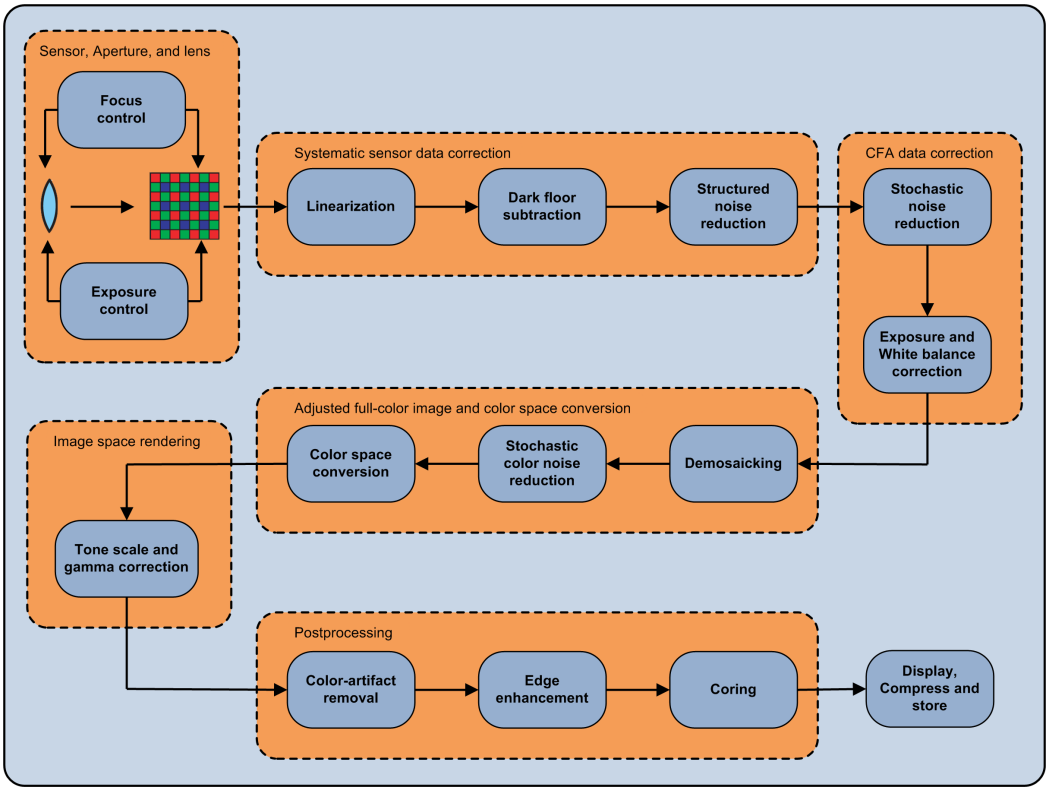


Figure 2.1: Signal flowchart of a typical DSC processing pipeline

solution, manufacturers place a CFA on the top of the sensor element. Control mechanisms interact with the sensor to determine the exposure (aperture size, shutter speed, and automatic gain control) and focal position of the lens. These parameters need to be determined dynamically based on the scene content. It is conventional to include an infrared blocking filter together with the lens system, as most of the filters that are used in CFAs are sensitive in the near-infrared part of the spectrum, as is the sensor itself.

Depending on the measured energy in the sensor, the exposure control system changes the aperture size and/or the shutter speed along with a carefully calibrated automatic gain controller to capture well-exposed images. In fact, an over- or under-exposed image will greatly affect output colors.

Focus control may be performed by using one of two types of approaches. The first category utilizes active approaches that typically use a pulsed beam of infrared light from a small source placed near the lens system to obtain an

estimate of the distance to the object of interest. The second category utilizes passive approaches that make use of the image formed in the camera to determine the best focus.

One of the major drawbacks in the use of a single sensor is that aliasing (due to the CFA sampling) causes highly objectionable artifacts in the output image. In order to reduce these artifacts, most cameras use an antialiasing filter. This helps to reduce Moiré patterns that may occur due to the sampling process involved. The antialiasing filter is important, although there is a tradeoff with image sharpness.

2.2.2 Systematic sensor data correction

The raw data obtained from the sensor need to be processed to remove noise and other artifacts before producing a full-color image. This full-color image will be later passed to a color processing pipeline to produce an accurate or, alternatively, a pleasing representation of the captured scene.

2.2.2.1 Linearization

Some cameras require the captured data to be linearized since they resides in a nonlinear space, mostly due to the electronics involved. Most DSC sensors typically have a linear response. Cameras that include correction for nonlinear data use an opto-electronic conversion function (OECF) that relates the input nonlinear data to an output linear space. Relating to eq.2.3, the OECF is the inverse of the term denoted as $\mathcal{N}$.

2.2.2.2 Dark floor subtraction

One may expect that a pixel receiving no light will produce a code value of zero. Unfortunately, thermal noise and other nonphoton-generated noise sources will produce nonzero pixel values. The common solution is to subtract a dark floor from the original CFA image. This can be accomplished in two different ways: applying a fixed value equally to every pixel in the image or alternatively, a spatially dependent set of values can be subtracted. While the goal is to remove

unwanted biases in the pixel data, some residual bias may be left in order to avoid data clipping and other quantization errors in the shadow region.

2.2.2.3 Structured noise reduction

It is known that dark floor subtraction operation tends to address low-frequency structured noise better than high-frequency structured noise. A significant type of high-frequency noise is due to defective pixels. It is almost inevitable that a number of the individual pixel in any given sensor will be defective. In fact, whole columns and rows of pixels might be nonfunctional. One can consider defective pixels falling into two broad categories. First, there are the pixels that are completely and consistently nonfunctional, being locked at complete black or complete white regardless of what amount of light may fall upon them. This is the simpler of the two categories to address. The second category is more problematic because these pixels are only partially defective. Given the difficulty to identify this category of pixels, one may be forced to treat all unidentified defective pixels with stochastic noise cleaning methods. However, the main method of defective pixel masking is to replace defective pixel values with the average values from known working neighboring pixels.

2.2.3 CFA data correction

The next stage of the image processing pipeline is concerned with the reduction of stochastic noise and correction for exposure and white balance errors at the time of capture.

2.2.3.1 Stochastic noise reduction

It is well known that if a system consist of a number of operations that are signal amplifiers, then it is best to reduce noise contributions as early as possible in the processing pipeline. Many of the image processing operation in Fig.2.1 are signal amplifiers, e.g. color correction, tone scale and gamma correction, and edge enhancement. This would suggest to perform noise reduction before these operations. Since at this stage of the processing pipeline only one color channel at each pixel location is known, it becomes difficult to exploit the partial correlation

between the color channels in the image. Consequently, the techniques used generally employ single-channel grayscale noise reductions.

2.2.3.2 Exposure and white balance correction

Unlike the human visual system (HVS) that constantly and automatically adjusts the apparent exposure and white point of what we see, digital cameras have no such innate functionality. Therefore, this adjustment must be made algorithmically after image capture. The goal of the first of the two algorithms, namely automatic white balance (AWB), is to render neutral areas in the scene as a region of equal code values for all color channels in the final image. For this operation three independent gains are usually employed, one for each of the sensor color channels. The second one, the auto-exposure (AE), maps midlevel grays (18% scene reflectance) to mid code value range in the final image. For this operation a common gain is usually employed, which is the same for all the sensor color channels. Sometimes the process of exposure and white balance correction are referred collectively as scene balance correction. It is to notice that it is not always the case that exposure and white balance correction must be done automatically: some cameras give the possibility to the user to use preprogrammed or even manual corrections.

2.2.4 Adjusted full-color image and color space conversion

At this point of the pipeline the CFA image data have been reduced in both structured and stochastic noise, and represent an image that has been properly scene balanced. It is important to underline that most subsequent image processing operations in the pipeline tacitly assume these idealized conditions.

The next task is to create a full-color image and then convert it into a known, calibrated color space.

2.2.4.1 Demosaicking

Demosaicking is perhaps the most computationally intensive step in the processing pipeline. The techniques used are usually either proprietary or covered by patents. All demosaicking techniques make use of pixel neighborhood information

to estimate the values of the pixel colors that were not measured. As one might imagine, this process will introduce artifacts that the remainder of the pipeline needs to remove. At the end of this stage, the image is in a $K-$band form, having code values in the color space of the camera, defined by the filters used in the CFA.

2.2.4.2 Stochastic color noise reduction

Now that all the color channels are fully populated, another facet of stochastic noise emerges. A texture that might be acceptable in the context of a single-channel image is deemed not acceptable when matched with similar, but different, textures in the other color channels. Instead of producing a light-dark texture, residual stochastic noise in an RGB image produces unexpected color variations. Such color artifacts are far more visible and objectionable than the corresponding light-dark fluctuations in a single-channel image. The color aspect of stochastic noise reduction is now addressed. Since this is a color noise reduction operation, the luminance channel may be left completely untouched.

2.2.4.3 Color space conversion

Given that the spectral sensitivities of the camera are not identical to the human color matching functions, it is clear that cameras are not colorimetric. It is then necessary to transform the image into a standard calibrated color space. There are a number of possible destination color spaces. Among these, industry has standardized in sRGB Stokes *et al.* (1996). However, because sRGB is itself a color transform from the CIE 1931 XYZ space CIE (1932), the latter can be considered the first target of the color space conversion. Assuming an RGB camera color space, the operation can be done using a 3×3 matrix multiplication, whose coefficients are computed in the factory through a regression process using measured camera RGB values of color patches with known XYZ tristimulus values. Once the XYZ tristimulus values have been computed, they can be transformed to sRGB tristimulus values with a standard matrix.

2.2.5 Image space rendering

The remaining steps in the image processing pipeline are targeted at producing the best image for a given image rendering. The transformation to sRGB began this process. The process is completed by transforming the image into a nonlinear space suited for video display devices.

2.2.5.1 Tone scale and gamma correction

The HVS's ability to adapt to a wide range of scene luminances is another essential capability that the digital camera must replicate. In this case, the overall contrast of the scene must be adjusted so that the image as viewed on the display device looks similar to the original scene. Furthermore, the image data must be transformed to account for the nonlinearity of the video display. The tone and gamma correction operation is implemented as a single transform composed of these two components.

The tone scaling operation adjusts the contrast of the image. It is usually implemented as a fixed lookup table that is applied equally to red, green and blue channels. It assumes that the input data is in a linear space and that it has been properly exposure corrected.

The second operation to be applied is the gamma correction. This standard transform is also defined in the sRGB specification Stokes *et al.* (1996) and accounts for the fundamental photometric nonlinearity of the displays.

2.2.6 Postprocessing

Different camera manufacturers use different proprietary steps subsequent to all the aforementioned processing, aimed at image appearance enhancement. Postprocessing is necessary as each of the previous steps may introduce objectionable artifacts. A few of the common postprocessing steps employed are color-artifact removal, edge enhancement and coring. These techniques are mostly heuristic based and require considerable fine-tuning.

2.2.6.1 Color-artifact removal

The demosaicking step introduces objectionable artifacts referred to as zipper and confetti artifacts. It is important to keep these artifacts to a minimum while at the same time retaining image sharpness. The solution to this problem lies in a series of choices starting from the camera lens system to the size of the sensor to the demosaicking technique used. For a given set of choices, further processing is used to reduce artifacts. This is typically done by generating luminance-chrominance channels from the demosaicked image and performing spatial operations on the chrominance channels.

2.2.6.2 Edge enhancement

The human eye is known to be highly sensitive to sharp edges; we prefer sharp edges in a scene to blurred ones. Most camera manufacturers use an edge enhancement step, also called sharpening, to make the image more appealing by increasing the high-frequency content in the image. Because noise has also high-frequency characteristics, attention must be given to the question of controlling noise amplification during edge enhancement.

2.2.6.3 Coring

In order to control noise amplification during the edge enhancement process, the high-frequency record needs to be noise cleaned somehow prior to being added back to the original image. Rather than performing a spatial noise cleaning operation, an amplitude noise cleaning method is usually employed. A coring function is used to noise-clean the high-frequency record. This function is a point operation that is usually implemented as a lookup table. The shape of the coring function is heuristically determined based on the fundamental noise characteristics of the digital camera system.

2.3 Simplified color correction pipeline

Given the importance of the color rendering accuracy, a simplified processing pipeline is usually studied which only consists of the modules more responsible

for the final color rendering accuracy. The automatic white balance and the color space conversion form what may be called the color correction pipeline. One of the main advantages of this simplified pipeline is that it involves only basic algebraic operations, and thus can be easily analyzed. Formally it can be described as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{out} = \left(\alpha \begin{bmatrix} a_{11}\, a_{12}\, a_{13} \\ a_{21}\, a_{22}\, a_{23} \\ a_{31}\, a_{32}\, a_{33} \end{bmatrix} \begin{bmatrix} r_{awb} & 0 & 0 \\ 0 & g_{awb} & 0 \\ 0 & 0 & b_{awb} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{in} \right)^{\gamma} \tag{2.5}$$

where $\mathbf{RGB}_{in}$ are the camera raw **RGB** values, α is an exposure compensation common gain, the diagonal matrix $diag(r_{awb}, g_{awb}, b_{awb})$ is the channel-independent gain compensation of the estimated illuminant in the scene, the full 3-by-3 matrix $a_{(i,j)}$, $(i, j) = \{1, 2, 3\}^2$ is the color space conversion transform from the white balanced device-dependent RGB to the sRGB color space and it is usually a-priori estimated assuming a perfect illuminant estimation and correction; finally, γ is the gamma correction defined for the sRGB color space and $\mathbf{RGB}_{out}$ are the output sRGB values.

This representation makes a tacit assumption clear on which the general pipeline is based: it is assumed that the data given as input to the color correction matrix have been correctly white balanced. This is not always the case, as the illuminant estimation is an under-determined problem, and so for certain images it may give a poor estimate. This would greatly affect the final color rendering accuracy, as these errors in the illuminant estimation may be even amplified by the color correction matrix. It is thus necessary to develop more robust illuminant estimation algorithms and color correction matrices which are more tolerant to illuminant estimation errors. The former will be the topic of Chapter 4, the latter the topic of Chapter 5.

Chapter 3

Base tools, methods, error metrics and datasets

In this chapter the base methods, metrics and datasets used in the following chapters will be described. For each of them their use is first described, followed by a brief description. More detailed descriptions can be found in the referenced papers.

3.1 Image System Evaluation Toolkit (ISET)

For the evaluation of the color correction matrices a common dataset of real images does not exist. To overcome this problem and permit in depth tests and analysis of the color correction matrices, synthetic datasets are often used. These are based on the creation of images composed of a variable number of patches with real reflectance data illuminated by real illuminants. The images are thus synthetic multispectral data. In order to simulate how a real DSC would response to such stimuli, a virtual camera simulator is needed.

The Image System Evaluation Toolkit (ISET) Farrell *et al.* (2003) is a virtual camera simulator developed at Stanford University. This virtual camera simulator was designed to help users evaluate how image capture components and algorithms influence image quality. This system makes it possible to simulate the entire image processing pipeline of a digital camera, combining both optical mod-

eling and sensor technology simulation. Moreover, the ISET is able to emulate different kinds of noise sources involved in the image acquisition process.

The simulation pipeline starts with the definition of the scene to be acquired. Digital camera simulation requires a physically accurate description of the light incident on the imaging sensor. ISET represents image scenes as a multidimensional array describing the spectral radiance (photons/sec/nm/m2) at each pixel in the sampled scene. The spectral radiance image data are assumed to arise from a single image plane at a specified distance from the optics. Multispectral images (both real or synthetic) are thus the input to the simulation pipeline.

The second step in the simulation pipeline is the optical modeling. The imaging optics are modeled using a wave-optics approach which takes into account the finite resolution obtained with finite size optics. The user can vary the size of the aperture of the imaging optics by changing the f-number, which will automatically result in an adjustment of the image irradiance and resolution. Finite resolution is calculated using an optical transfer function (OTF) approach, which is based on the finite aperture as determined by the f-number. Image irradiance is determined using radiometric concepts and includes the effect of the off-axis cos-4th effect, which results in a darkening of the corners with respect to the center of the image when a uniform object is imaged.

The third step is the sensor modeling. The function of the sensor model is to simulate the transformation of irradiance (photons/nm/m2) into an electrical signal. The image sensor model includes a great many design parameters. Among the various factors accounted for are the spatial sampling of the optical image by the image sensor, the size and position of pixels and their fill-factor. The wavelength selectivity of color filters, intervening filters (such as an infrared filter), and the photodetector spectral quantum efficiency are also included in the simulation. The sensor current (electrons/s) is converted into a voltage (V) using a user-specified conversion gain (V/electron). Various sources of noise (read noise, dark current, DSNU, PRNU, photon noise) are all included in the simulation. Finally, to complete the physical signal pipeline, the analog voltage (V) is converted into a digital signal (DN) according to the specifications of the user (analog-to-digital step size in V/DN).

The last step in the simulation pipeline is the processor modeling by means of the image processing module. The image processing module transforms the linear sensor values into an image for display. The image processing module includes several standard algorithms for basic camera operation. These include algorithms for interpolating missing RGB sensor values (demosaicing) and transforming sensor RGB values into an internal color space for encoding and display. Because of the simulator's extensible and open-source architecture (Matlab) the user can also insert proprietary image-processing algorithms in the pipeline and evaluate how these algorithms will perform under various imaging conditions or with various types of sensors.

3.2 Pattern Search Method (PSM)

The first tool to be introduced is an optimization method. This will be used in the following chapters to optimize the parameters of the white balance algorithms (section 4.5), the thresholds of the proposed strategies (section 4.6.2), and to find the color correction matrices (chapter 5). The selected optimization tool is the Pattern Search Method.

Pattern Search Methods (PSM) are a class of direct search methods for nonlinear optimization Lewis & Torczon (1999, 2000). Their popularity is given by their simplicity and by the fact that they work very well in practice on a variety of problems. Furthermore, global convergence can be established under certain regularity assumptions of the function to minimize Lewis & Torczon (1997). PSMs are also extremely simple to implement and do not require any explicit estimate of derivatives.

The form of a general pattern search algorithm can be described in the following way. At each step k, we have the current iterate $\mathbf{x}_k$, a set D_k of vectors which identify the search directions, and a step-length parameter Δ_k. Usually the set D_k is the same for all iterations. For each direction $\mathbf{d}_k \in D_k$, we set $\mathbf{x}^+ = \mathbf{x}_k + \Delta_k \mathbf{d}_k$ (the "pattern") and we examine $f(\mathbf{x}^+)$ where f is the function to be minimized. If $\exists \mathbf{d}_k \in D_k : f(\mathbf{x}^+) < f(\mathbf{x}_k)$, we set $\mathbf{x}_{k+1} = \mathbf{x}^+$ and $\Delta_{k+1} = \alpha_k \Delta_k$ with $\alpha_k > 1$; otherwise, we set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and $\Delta_{k+1} = \beta_k \Delta_k$ with $\beta_k < 1$. The algorithm stops

when the step Δ_k is smaller than a fixed threshold or when the maximum number of iterations has been reached.

3.3 Classification and Regression Trees (CART)

The second tool to be introduced is a classification tool. This will be necessary for all the white balance strategies proposed in chapter 4 as both the class-based and feature-based illuminant estimation algorithms exploit a classification step. The chosen classification method is the Classification and Regression Trees one.

The Classification and Regression Trees (CART) Breiman *et al.* (1984) is a non-parametric and non-metric method that does not need neither a-priori knowledge about the distribution of the values of the features nor feature normalization. CART methodology has proven to be effective for image classification tasks Schettini *et al.* (2002a, 2004). CART trees are produced by recursively partitioning a set of feature vectors $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ labeled with the corresponding correct class $\{y_1, \ldots, y_N\}, y_j \in \{1, \ldots, K\}$. The partition is driven by an impurity function which measures the diversity of the classes associated to a set of feature vectors on the basis of the estimated distribution of the classes in that set. The Gini diversity index can be used as the impurity function: $i_{Gini}(P_1, \ldots, P_K) = 1 - \sum_{j=1}^{k} P_j^2$. The process starts by considering the whole training set T. For each feature j and for each value of the threshold τ, the subsets $T_L = \{\mathbf{x} \in T | \mathbf{x}_j \leq \tau\}$, $T_R = T \setminus T_L$ are defined and the decrease in impurity is computed as:

$$\Delta I(j, \tau) = i(T) - i(T_L)P_L - i(T_R)P_R, \tag{3.1}$$

where P_L and P_R are the resubstitution estimates of the probabilities that an element of T falls into the subsets T_L and T_R, respectively; $i(T)$ represents the application of an impurity function i to the resubstitution estimates of the distribution $P(y = 1|T), \ldots, P(y = K|T)$ of the classes of the elements of the set T. Among all possible splits (j, τ) the one which maximizes the decrease in impurity is selected. In tree terminology T represents the parent node of the nodes T_L and T_R. The process is recursively repeated for the new nodes T_L and T_R until no further decrease in impurity is possible.

At this point, each terminal node L of the tree is labeled with the class which minimizes the misclassification error:

$$\hat{y} = \underset{j \in \{1,\dots,K\}}{\arg\max}\, P(y = j|L). \tag{3.2}$$

New cases are classified by choosing the classes associated to the terminal nodes in which they fall on the basis of the values of the features. Since the trees almost certainly overfit the training data, generalization accuracy is expected to be low. To avoid overfitting, a pruning procedure is often applied. Instead of minimizing the misclassification error on the training set, a cost-complexity criterion may be considered: given a tree T, its performance is measured by:

$$R_\alpha(T) = R(T) + \alpha|T| \ , \tag{3.3}$$

where $R(T)$ is the probability of misclassification estimated on the training set, $|T|$ is the size of the tree (the number of terminal nodes), and α is a parameter which weights prediction errors and complexity of the tree. Starting from an initial tree T_0 (corresponding to $\alpha = 0$) built as described above and increasing the value of α, the sequence of best subtrees $T_0, T_1, \dots, T_M$ is considered. Each subtree corresponds to the optimal subtree of T_0 for a range of values of the parameter α. The subtree of the sequence which minimizes the misclassification error on an independent validation set is finally selected.

The pruning process improves the generalization accuracy of the trees. However a low misclassification error cannot be ensured due to the instability of the training procedure (slightly different training sets could produce completely different trees). To avoid instability, a multiple classifier approach can be adopted. Several CART trees are built to form a decision forest Breiman (1996). The trees of the forest are first built using different bootstrap replicates of the training set. The complement of each replicate is then used as a validation set to prune the corresponding tree. New cases are classified by majority vote on the output of the trees of the forest.

3.4 Angular color error

In order to evaluate the performance of the illuminant estimation algorithms, we have to define an error measure. Since in estimating the scene illuminant it is more important to estimate its color than its overall intensity, the error measure has to be intensity-independent.

There are different measures to asses the performance of color constancy algorithms. Two metrics are used frequently, angular error and root mean squared error(RMSE). These two measures are independent of the brightness of the illuminant and simply compare colors. Angular error is defined as the angle between the RGB triplets of the ground truth illuminant color ($\boldsymbol{\rho}_w = (R_w, G_w, B_w)$) and the algorithm's estimate of it ($\hat{\boldsymbol{\rho}_w} = (\hat{R_w}, \hat{G_w}, \hat{B_w})$):

$$e_{ANG} = \arccos\left(\frac{\boldsymbol{\rho}_w^T \hat{\boldsymbol{\rho}_w}}{\|\boldsymbol{\rho}_w\| \|\hat{\boldsymbol{\rho}_w}\|}\right). \tag{3.4}$$

RMSE is computed between the chromaticities of the illuminant ($r_w = R_w/(R_w + G_w + B_w)$; $g_w = G_w/(R_w + G_w + B_w)$) and the ones of the ground truth illuminant.

As suggested by Hordley and Finlayson Hordley & Finlayson (2004), in this thesis the error metric chosen is the angular error. This choice is motivated by the fact that it is the most widely used and comparison in terms of it can be easily done with algorithms in the state of the art. Hordely and Finlayson Hordley & Finlayson (2004) also showed that a good descriptor for the angular error distribution is the median error.

3.5 Wilcoxon Sign Test

In order to compare the whole error distributions of the color constancy algorithms in addition to the median angular error, a statistic test is needed. Since standard probability models cannot represent underlying errors well, a test that does not make any a-priori assumptions about the underlying error distributions is needed. As suggested in Hordley & Finlayson (2004) the Wilcoxon Sign Test (WST) Wilcoxon (1945) is used.

The WST is a non-parametric test used to test the null hypothesis that two error distributions have the same median, against the alternative hypothesis that they do not have it.

Let X and Y be random variables representing the error distributions that we want to compare; let μ_X and μ_Y be the median values of such random variables. The Wilcoxon signed-rank test can be used to test the null hypothesis $H_0 : \mu_X = \mu_Y$. To test H_0, we consider the differences of independent error pairs $(X_1 - Y_1), \cdots, (X_N - Y_N)$ for N different samples. We rank these error pairs according to their absolute differences. Ranks are signed considering whether the corresponding error pair difference is positive or negative. If H_0 is correct, the sum of the signed ranks W will approximate zero. If W is much larger or smaller than zero, the alternative hypothesis $H_1 : \mu_X \neq \mu_Y$ is true. We can test H_0 against H_1 at a given significance level α. We reject H_0 and accept H_1 if the probability of observing the error differences we obtained is less than or equal to α.

3.6 CIELAB color difference formula

In order to quantify the color errors made in a processing pipeline (in particular the ones introduced by the color transformation matrix, chapter 5), a standard method to specify the color is needed.

The International Commission on Illumination (Commission International de l'Eclairagem CIE) standardized a method for specifying the color of illuminants and materials by tristimulus values X, Y and Z:

$$X = k \int \Phi(\lambda)\overline{x}(\lambda)\mathrm{d}\lambda \tag{3.5}$$

$$Y = k \int \Phi(\lambda)\overline{y}(\lambda)\mathrm{d}\lambda \tag{3.6}$$

$$Z = k \int \Phi(\lambda)\overline{z}(\lambda)\mathrm{d}\lambda \tag{3.7}$$

where $\Phi(\lambda)$ describes the spectral power of the stimulus and $\overline{x}(\lambda)$, $\overline{y}(\lambda)$ and $\overline{z}(\lambda)$ are the color-matching functions of the 1931 CIE standard observer. For an illuminant, $\Phi(\lambda) = S(\lambda)$, which is the relative spectral power of the illuminant. For a reflecting material, $\Phi(\lambda) = S(\lambda)R(\lambda)$ where $S(\lambda)$ is the relative power of

an illuminant and $R(\lambda)$ is the reflection factor of the material. For a transmissive material, $\Phi(\lambda) = S(\lambda)T(\lambda)$ where $T(\lambda)$ is the transmission factor of the material. By convention, Y is assigned the value 100 for a perfect reflecting diffuser.

The spacing of the colors in the XYZ space is not perceptually uniform. This means that at equal distances in the XYZ space do not correspond equal perceived distances. The XYZ space can be transformed to a more nearly uniform CIE 1976 $L^*a^*b^*$ (CIELAB) color space. The transformation equations for CIELAB are:

$$L^* = 116\left[f\left(\frac{Y}{Y_n}\right) - \frac{16}{116}\right] \tag{3.8}$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - \left(\frac{Y}{Y_n}\right)\right] \tag{3.9}$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - \left(\frac{Z}{Z_n}\right)\right] \tag{3.10}$$

$$f(w) = \begin{cases} w^{1/3} & \text{if } w > 0.008856 \\ 7.787w + \frac{16}{116} & \text{otherwise} \end{cases} \tag{3.11}$$

$$C^*_{ab} = \left(a^{*2} + b^{*2}\right)^{1/2} \tag{3.12}$$

$$h_{ab} = \tan^{-1}\left(\frac{b^*}{a^*}\right) \tag{3.13}$$

where X_n, Y_n and Z_n are the tristimulus values of the reference white, L^* denotes lightness, a^* and b^* denote chromaticity, C^*_{ab} denotes chroma, and h_{ab} denotes hue. The CIELAB equations were derived so that the illuminant is always at $L^* = 100$, $a^* = 0$, $b^* = 0$. Thus, the illuminant is the reference white.

Total color difference ΔE_{ab} from a reference color (L^*_0, a^*_0, b^*_0) to a target color (L^*_1, a^*_1, b^*_1) in the CIELAB space is given by:

$$\Delta E_{ab} = \left[(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2\right]^{1/2} \tag{3.14}$$

where

$$\Delta L^* = L^*_1 - L^*_0 \tag{3.15}$$

$$\Delta a^* = a^*_1 - a^*_0 \tag{3.16}$$

$$\Delta b^* = b^*_1 - b^*_0 \tag{3.17}$$

The CIE 1994 color difference ΔE_{94} was introduced with the aim of being more perceptually uniform than the ΔE_{ab}. Its calculation is performed as:

$$\Delta E_{94} = \left[\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2\right]^{1/2} \tag{3.18}$$

$$\Delta H = \left[(\Delta a^*)^2 + (\Delta b^*)^2 + (\Delta C^*)^2\right]^{1/2} \tag{3.19}$$

$$S_L = 1 \tag{3.20}$$

$$S_C = 1 + K_1 C_1 \tag{3.21}$$

$$S_H = 1 + K_2 C_1 \tag{3.22}$$

$$k_L = k_C = k_H = 1 \text{ for reference conditions} \tag{3.23}$$

$$K_1 = 0.045 \tag{3.24}$$

$$K_2 = 0.015 \tag{3.25}$$

3.7 Spatial extension of the CIELAB color difference formula: sCIELAB

In order to compare not only isolated colors, but whole images in a perceptual way that accords with what the HVS would perceive, an extension of the CIELAB color metric is needed.

The sCIELAB Johnson & Fairchild (2003); Zhang & Wandell (1997a) is a spatial extension of the CIELAB color metric which is useful for measuring color reproduction errors of digital images.

The image data are transformed into an opponent-color space. Each opponent-color image is convolved with a kernel whose shape is determined by the visual spatial sensitivity to that color dimension. The calculation is pattern-color separable because the color transformation does not depend on the image's spatial pattern, and the spatial convolution does not depend on the image's color. The filtered representation is transformed to a CIE-XYZ representation, and this representation is transformed using the CIELAB formula. The resulting S-CIELAB representation includes both the spatial filtering and the CIELAB processing.

Differences between the sCIELAB representation of an original image and its reproduction measure the reproduction error. These differences are summarized by a quantity, ΔE_s, which is computed precisely as ΔE in the conventional CIELAB. The sCIELAB difference measure reflects both spatial and color sensitivity, and it equals the conventional CIELAB difference measure over uniform regions of the image.

Inspired by the excellent description of the sCIELAB given in Johnson & Fairchild (2003), more details about each step of the sCIELAB can be found here. The first step consists of a transformation into an opponent color space, containing one luminance and two chrominance channels. These channels were determined through a series of psychophysical experiments testing for pattern color separability. The opponent channels, AC1C2, are a linear transform from CIE 1931 XYZ:

$$\begin{bmatrix} A \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0.297 & 0.720 & -0.107 \\ -0.449 & 0.290 & -0.077 \\ 0.086 & -0.590 & 0.501 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.26}$$

After the transformation of both the images into the opponent color space, the channels are independently spatially filtered, using filters that approximate the contrast sensitivity functions (CSF) of the HVS. The traditional sCIELAB model used two-dimensional separable convolution kernels. These kernels are unit sum kernels in the form of a series of Gaussian functions. The unit sum was designed such that for large uniform areas sCIELAB predictions are identical to the corresponding CIELAB predictions. Equation 3.27 and 3.28 show the spatial form of the convolution kernels:

$$\text{filter} = k \sum_i w_i E_i \tag{3.27}$$

where

$$E_i = k_i \exp -(x^2 + y^2)/\sigma_i^2 \tag{3.28}$$

The parameters k and k_i normalize the filters so that they sum to one, thus preserving the mean color value for uniform areas. The parameters w_i and σ_i represent the weight and the spread of the Gaussian functions. The values used in the sCIELAB are reported in Table 3.1.

Table 3.1: Weight and spread of Gaussian convolution kernels used in the sCIELAB

Filter	Index (i)	Weight (w_i)	Spread (σ_i)
Achromatic	1	1.00327	0.0500
	2	0.11442	0.2250
	3	-0.11769	7.0000
Red-Green	1	0.61673	0.0685
	2	0.38328	0.8260
Blue-Yellow	1	0.56789	0.0920
	2	0.43212	0.6451

The filtered opponent channels are then transformed back into the CIE XYZ space using the inverse of equation 3.26. The filtered XYZ pixel values for both the original and the test image are then transformed into the CIELAB space using the formulae introduced in section 3.6. Once the CIELAB coordinates are calculated for the filtered images, color differences can be determined on a pixel-by-pixel basis using the CIE 1994 color difference ΔE_{94} (see section 3.6).

3.8 Standard color datasets

3.8.1 Grayball dataset

In order to asses the performance of the illuminant estimation algorithms, a dataset of real images with ground truth illuminant measurement is needed.

In Ciurea & Funt (2003) Ciurea and Funt presented an image database to be used as a common data set in the evaluation of color constancy algorithms. In this database, 15 digital video clips were recorded (at 15 frames per second) in different settings such as indoor, outdoor, desert, markets, cityscape, etc. . . for a total of two hours of videos. From each clip, a set of images was extracted, resulting in a database of more than 11,000 images. In each image, a gray sphere appears

in the bottom right corner of the images. This sphere was used to estimate the color of the scene illuminant. The database is thus composed of images taken at different locations, each one coupled with the measured illuminant.

3.8.2 Spectral dataset

In order to generate synthetic scenes with real reflectance and illuminant data (which are then given as input to the ISET) to asses the performance of the color space transformation strategies, a real dataset of both is required.

The reflectance dataset selected is the ISO reflectance database ISO (2003). This is composed of real reflectance data for a total of more than 50,000 items. The illuminant dataset selected is the one used in Barnard *et al.* (2002), which is composed of more than 200 illuminant spectral power distributions. These are both measured and linear interpolated in order to simulate mixed illuminants.

3.8.3 Cambridge dataset

In order to asses the performance of the color correction pipelines in real-world situations, a dataset of real, unprocessed images is needed. In order to estimate the color rendition accuracy of the pipelines, it is required that a known color target is placed inside each photographed scene.

In Gehler *et al.* (2008) a dataset RAW camera images is presented. This dataset is captured using a high-quality digital SLR camera in RAW format, that is therefore free of any color correction. The dataset consists of a total of 568 images, both indoor (246) and outdoor (322). The Macbeth ColorChecker (MCC) chart is included in every scene acquired, and this allows to estimate accurately the actual illuminant within each acquired image and to measure the final color rendition accuracy of the investigated color correction pipelines.

Chapter 4

Classification-based illuminant estimation

As already pointed out in Sec.2.2.3.2 the goal of the AWB is to render neutral areas in the scene as region of equal code values for all color channels in the final image. The AWB is in fact a two stage procedure: the scene illuminant is estimated from the image data, and the image colors are then corrected on the basis of this estimate to generate a new image of the scene as if it were taken under a known, canonical illuminant.

With reference to eq.2.1 and assuming that the color $\mathbf{L}$ of the scene illuminant as seen by the camera only depends on the illuminant spectral power distribution $I(\lambda)$ and on the sensor spectral sensitivities $\mathbf{C}(\lambda)$, then the illuminant estimation problem is equivalent to the estimation of $\mathbf{I}$ by:

$$\mathbf{I} = \int_{\omega} I(\lambda)\mathbf{C}(\lambda)\mathrm{d}\lambda, \tag{4.1}$$

given only the sensor responses $\boldsymbol{\rho}(x, y)$ across the image.

Many illuminant estimation solutions have been proposed in the last few years although it is known that the problem addressed is actually ill-posed as its solution lack uniqueness or stabilityFunt *et al.* (1998). To cope with this problem different solutions usually exploit some assumptions about the statistical properties of the expected illuminants and/or of the object reflectances in the scene. Recently, considering some well known and widely used color constancy algorithms that are based on color image statistics, it has been shown that on large datasets of

both synthetic and real images the best and the worst algorithms do not exist at all Bianco *et al.* (2008c).

Hordley in his review paper Hordley (2006) gives an excellent review of illuminant estimation algorithms and highlighted two research areas that are important in the context of improving the performance of color constancy algorithms: making additional measurements at the time of image capture (i.e. using more color pixel information), and algorithm combining (i.e. using two or more estimations of the illuminants). In this chapter a third hypothesis is investigated: the use of low level visual information to improve illuminant estimation.

The use of content-based image analysis for automatic color correction has been originally proposed by Schröder and MoserSchröder & Moser (2001). They classify the images into several signal-oriented generic classes (e.g. scene with high color complexity) and, after the class-specific application of a set of color correction algorithms (White Patch and Gray World), they combine the results in such a way as to take into account the class-specific reliabilities of each algorithm. Their proposal is based on a hierarchical Bayesian image content analysis consisting of feature extraction and unsupervised clustering. They also suggest that semantic classes (e.g. indoor, outdoor, vegetation scene, mountain scene,...) and specific image degradation classes (e.g. underexposure, strong color cast, ...) could be used in a similar way. Gasparini and Schettini Gasparini & Schettini (2004) applied an adaptive mixture of the white balance and gray world procedures. In order to avoid the mistaken removal of an intrinsic color, regions identified as probably corresponding to skin, sky, sea or vegetation, are temporarily removed from the analyzed image. Van de Weijer *et al.* van de Weijer *et al.* (2007b) proposed high-level visual information to improve illuminant estimation. They modeled the image as a mixture of semantic classes such as grass, skin, road and building and exploited this information to select the best illuminant out of a set of possible ones. They applied several illuminant estimation approaches to compute a set of possible illuminants. For each of them an illuminant color corrected image is evaluated on the likelihood of its semantic content and the illuminant resulting as the most likely semantic composition of the image is selected as the illuminant color. They tested their method on a small subset of the Ciurea and Funt database Ciurea & Funt (2003) that is composed of a variety of both indoor

and outdoor scenes and has shown that their top-down approach on outdoor images works better than any other tested algorithms. Gijsenij and Gevers Gijsenij & Gevers (2007) used natural image statistics to identify the most important characteristics of color images and achieve selection and/or combining of color constancy algorithms. To this end, they used the Weibull parameterization to capture the image characteristics, applied a $k-$means algorithm to cluster the parameters in a predefined set of clusters, and then associated the best-suited color constancy algorithm to each cluster. Unseen images are assigned to the computed clusters, and the best color constancy algorithm for that image is chosen. Gijsenij and Gevers Gijsenij & Gevers (2007) have also suggested that for certain scene categories, such as forest, open country and streets, a specific color constancy algorithm can be used.

Once the scene illuminant is estimated from the image data, the image colors are corrected on the basis of this estimate to generate a new image of the scene as if it were taken under a known, canonical illuminant. This is usually accomplished using a diagonal model of illumination change. This model is derived from the Von Kries hypothesis that color constancy is an independent gain regulation of the three cone signals, through three different gain coefficients von Kries (1902). This correction can be easily implemented on digital devices as a diagonal matrix multiplication.

This chapter starts with the description of the low-level features used to improve the illuminant estimation. Then follows the description of the selection of training and test images in the dataset considered; next there is the description and optimization of the basic color constancy algorithm considered, followed by the selected performance measure metric. The two different ways in which the idea of using low-level features is realized are then described: in the first realization the low level features considered are used to predict if the considered image is an indoor or outdoor scene. This class-based algorithm Bianco *et al.* (2008a,b) is the topic of section 4.6. In the second realization the low level features are directly linked to the color constancy algorithms, without exploiting any semantic class. This feature-based algorithm Bianco *et al.* (2009b,c) is the topic of section 4.7. The chapter concludes with the comparison of the performance of the proposed algorithms and the benchmarking algorithms' ones.

4.1 Image features

Since an image conveys information at different levels, the use of different features at the same time is a necessary requisite if we want to capture most of the image information. There is no single "best" representation of the content of an image, but only multiple representations which characterize the content from different perspectives. In the literature there are many features used to describe the image content Antani *et al.* (2002); Eakins (2002); Schettini *et al.* (2002b); Sikora (2001).

Two groups of low level features have been considered: general purpose features and problem-dependent features. The general purpose features are features that can be used on a large range of applications since they do not capture the characteristic of the images that are problem specific. The features selected in this category are: color histogram, edge direction histogram, statistics on the wavelet coefficients, and color moments.

Problem-dependent features try to capture properties of the images that can be useful in improving the performance of the task being considered. Since several algorithms of illuminant estimation rely on some image characteristics or make certain assumptions about the color of the images, some of the problem dependent features have been chosen to exploit these properties. For example, the extent of a color cast in an image is a feature that may be useful in the color balancing problem. A strong cast may be an indication that a particular illuminant is present. The number of different colors is an indication of color variability, and thus that the Gray World assumption can be justified for the image being considered. Conversely, an image with very few colors may not be reliably processed for balancing, since it may lack a sufficient amount of information to estimate the white point. Several algorithms rely on the edges found in the image, so features that extract edge information at different perspectives have been included.

Summarizing, the problem-dependent features chosen are: the number of different colors, the percentage of color components that are clipped to the highest and lowest value that can be represented in the image color space, a cast index representing the extent of the presence of a color cast in the image and the magnitudes of the edges.

Note that all these features have been chosen independently from their usefulness in the image classification process. They have been chosen uniquely for their ability to describe the content of an image. Some of them are not strictly independent in the sense that similar properties of the images are evaluated using different features. The aim of the classifier is to choose the features as well as which specific components in a feature are more relevant to discriminate between the classes selected for the problem under analysis. Moreover, while all the features must be computed for the images in the training sets, only the features actually chosen and used by the classifier need to be computed for the images in the test sets and for new images to be processed. This approach is made possible by the use of CART trees as classifiers. Other classification methodologies (such as support vector machines and neural networks) would have required a complex feature selection (and normalization) step.

4.1.1 General purpose features

4.1.1.1 Edge Direction Histogram

Edges are a clue about the subject depicted in an image. Strong edges can be found in buildings, roads, and other man-made structures. These edges usually have directions in a definite pattern (see Figure 4.1). On the other hand, pictures of natural scenes usually do not show strong edges and since the subject has no clear structure they do not show a specific pattern (see Figure 4.2). Edge direction histogram can be used to determine the edge structures within an image and thus allow us to distinguish between different image classes. Edges are computed applying a Derivate of the Gaussian filter with $\sigma = 1$ on the luminance image in both the x and y directions (G_x, G_y) and then the edge orientation at edge position (x, y) are computed as follows:

$$\theta(x, y) = \arctan\left(\frac{G_y(x, y)}{G_x(x, y)}\right). \tag{4.2}$$

The orientations are quantized into 18 bins each corresponding to angles of intervals of 10 degrees. The quantized orientations are then used to compute an edge direction histogram of 18 components. Only the orientations belonging to

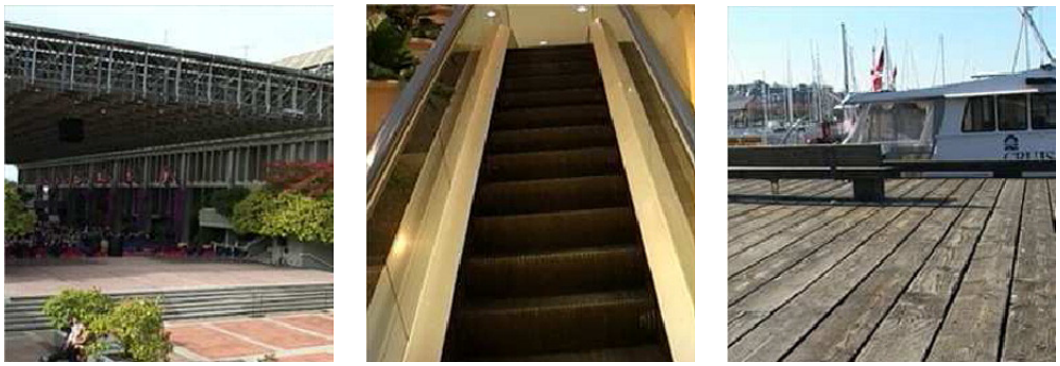


Figure 4.1: Examples of images of man-made structures.

Figure 4.2: Examples of images of natural scenes.

edges whose gradient's magnitude is above a given threshold (0.50 in our case) are taken into account. This ensures that only edges with sufficient strength are used in computing the direction histogram.

4.1.1.2 Edge Strengths

In order to capture the relevance of the edges a histogram of edge magnitudes is computed. The edges are detected as in the case of the edge direction histogram. The magnitude is quantized into 5 bins corresponding to the following intervals: $[0, 0.25)$, $[0.25, 0.50)$, $[0.50, 0.75)$, $[0.75, 1.0)$, $[1.0, \infty)$. Two examples of images with different edge strengths are shown in Figure 4.3.

Figure 4.3: Examples of images with weak (left) and strong (right) edge magnitudes.

4.1.1.3 Color Histogram

The color histogram is one of the most widely used image descriptors Gong *et al.* (1996); Swain & Ballard (1991) and represents the color distribution of the image. It possesses several useful properties that make it a robust visual feature such as compactness, invariance and robustness with respect to the geometric transformation of the original image like rotation and scale. In order to compute the histogram the RGB color space is quantized by uniformly dividing each color axis into 3 intervals. The RGB cube is thus subdivided into 27 smaller cubes and each of the original colors is mapped to the cube which it falls into.

4.1.1.4 Wavelet Statistics

Information about the textures and structures within the image can be obtained using a wavelet decomposition. This technique is often used in content-based retrieval for similarity retrieval, target search, compression, texture analysis, biometrics, etc... Idris & Panchanathan (1997); Scheunders *et al.* (1997). In multiresolution wavelet analysis, at each level of resolution (i.e. at each application of the wavelet decomposition) four bands containing different information are obtained by applying a combination of a low pass filter (L) and a high-pass filter (H). Specifically, the information correspond to a low-pass filtered version of the processed image (LL band), and three bands of details that roughly correspond to the horizontal edges (LH band) of the original images, the vertical edges (HL

band) and the diagonal edges (HH band). Each band is a matrix of values, one fourth the size of the original image.

Wavelet decomposition is applied recursively to the LL band. The resultant decomposition will contain information, i.e. details, at the lower resolution. The process can be repeated until the LL sub-band cannot be further processed or until a given number of wavelet decomposition applications is reached. Different filters can be used to produce the bands of the wavelet analysis Mojsilovic *et al.* (2000) e.g. Harr, Daubechies, Symlet, Biort, etc... For our purposes the wavelet statistic features are extracted from the luminance image using a three-iteration Daubechies wavelet decomposition, producing a total of ten bands as shown in Figure 4.4. The mean and variance of the absolute values in each band are then computed as band statistics. These feature values represent the energy i.e. the amount of information within each band and provide a concise description of the image's contents. This feature is thus composed by 20 (2 energy values for each of the 10 bands) components.

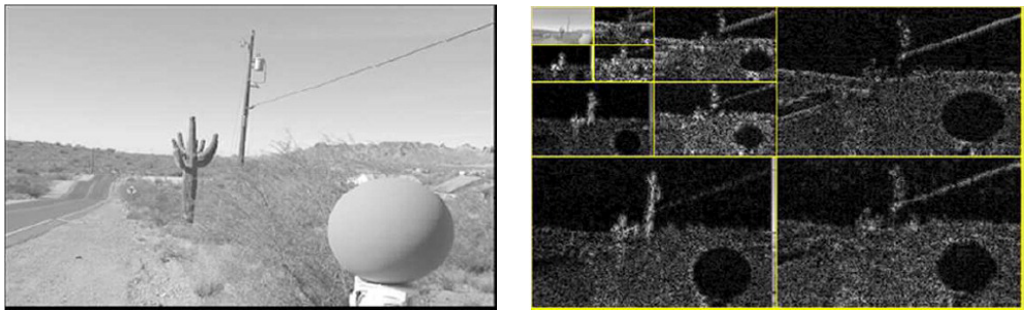


Figure 4.4: A three-iteration Daubechies wavelet decomposition.

4.1.1.5 YCbCr Color Moments

To describe the color distribution of an image, the first two central moments, mean, and standard deviation of each color channel of the YCbCr color space are computed, derived by transforming the R, G, and B color coordinates. The color distribution of an image can, in fact, be considered a probability distribution and can therefore be characterized uniquely by its central moments alone, as

can any probability distribution Stricker & Orengo (1995). The choice of the YCbCr color space allows the separation of the luminance component from the chrominance components in a simple way using a linear transformation. The color transformation used is that defined in the ITU-R Recommendation BT.601 ITU-R (1995):

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} + \begin{bmatrix} 65.74 & 129.06 & 25.06 \\ -37.95 & -74.50 & 112.44 \\ 112.44 & -94.15 & -18.29 \end{bmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}. \tag{4.3}$$

where r,g,b are the **RGB** coordinates normalized in the range $[0, 1]$. This feature is composed of 6 values (two statistics for each of the three color channels).

4.1.2 Problem-dependent features

4.1.2.1 Number of Colors

The number of distinct colors is related to the color range of the image. Since several illuminant estimation algorithms are based on the Gray World assumption, the color range is an indication of whether this assumption holds true for the given image or not. The actual values of the pixels colors may impede the occurrence of the assumption, but if an image contains many different colors then the average color is likely to be a gray value. Two examples can be seen in Figures 4.5 and 4.6. To remove small variations in the color appearance and thus limit the influence of noise in the computation of the feature, the RGB color channels are quantized by considering only the six most significant bits. Thus, the maximum number of different colors that can be discriminated is $2^6 \times 2^6 \times 2^6 = 262\ 144$.

4.1.2.2 Clipped color components

To take into account the extent of highly saturated color pixels, the percentage of pixels whose color components are clipped to the maximum value that can be represented are computed. For digital images with eight-bit color channel representation, the maximum value is 255. A discrimination is used between pixels with zero, one, two or all three color components clipped to the maximum value. In total the histogram of clipped color components is composed of eight

Figure 4.5: Examples of images with many different colors. Left image: 10 782 colors with average color (122, 123, 121) . Right image: 13 882 colors with average color (107, 106, 110).

Figure 4.6: Examples of images with few different colors. Left image: 5 380 colors with average color (150, 97, 47) . Right image: 7 538 colors with average color (96, 126, 150).

bins: 0 clipped components, 1 clipped component (either R, or G or B), 2 clipped components (either R and G, or R and B, or G and B) and 3 clipped components (R and G and B).

The values in the histogram bins are normalized with respect to the total number of pixels in the image, so that the histogram represents a probability density distribution.

4.1.2.3 Cast indexes

The cast index aims to identify the presence of a relevant cast within the image; and it is inspired by the work done in Gasparini & Schettini (2004), where the cast is detected and classified into several classes according to its relevance. The basic idea of the cast detection is that the color distribution of an image can be analyzed by converting it into a suitable color space and using statistical tools to characterize the presence of the cast. In Gasparini & Schettini (2004) the presence of a color cast is used to process images producing more pleasing images; that is, images that users perceive as more natural than the original one. In this thesis, small modifications of the original formulation are made, since the problem faced here is different from the one in Gasparini & Schettini (2004). The color space representation is changed from the CIELAB to YCbCr, since the former depends on the knowledge of the white point of the scene. Moreover, only the cast indexes disregarding the final cast classification are considered.

An image with a very strong cast will show one definite peak within the CbCr plane, far away from the neutral axis corresponding to the color cast.

The means and variances of the Cb and Cr components (μ_{Cb} and μ_{Cr}, σ^2_{Cb} and σ^2_{Cr}) are used to compute the color Equivalence Circle centre $C = (\mu_{Cb}, \mu_{Cr})$ and its radius $r = (\sigma^2_{Cb} + \sigma^2_{Cr})^{\frac{1}{2}}$, as well as the two cast indexes $D = \mu - \sigma$ and $D_\sigma = D/\sigma$ where $\mu = (\mu^2_{Cb} + \mu^2_{Cr})^{\frac{1}{2}}$. D is a measure of how far the color distribution is from the neutral axis (i.e from $(0, 0)$ in the CbCr coordinates), thus indicating the presence of a cast. D_σ quantifies the strength of the cast. An example of an image showing a strong cast is depicted in Figure 4.7. Figure 4.8 shows an example of an image without cast.

4.2 Illuminant dataset selection

In Ciurea & Funt (2003) Ciurea and Funt presented an image database to be used as a common data set in the evaluation of color constancy algorithms. In this database, 15 digital video clips were recorded (at 15 frames per second) in different settings such as indoor, outdoor, desert, markets, cityscape, etc. . . for a total of two hours of videos. From each clip, a set of images was extracted resulting in

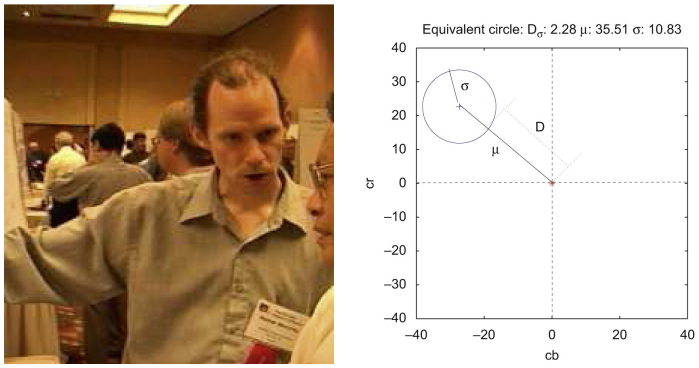


Figure 4.7: Example of an image with a strong color cast. The equivalence circle is compact and far from the neutral axis.

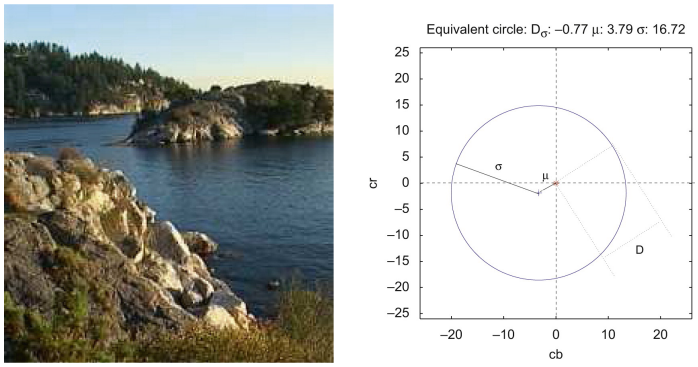


Figure 4.8: Example of an image without color cast. The equivalence circle is large and close to the neutral axis.

a database of more than 11,000 images. In each image, a gray sphere appears in the bottom right corner of the images. This sphere was used to estimate the color of the scene illuminant. The database is thus composed of images taken at different locations, each one coupled with the measured illuminant.

Since the database sources were video clips, the images extracted show high correlation. To remove this correlation only a subset of images should be used from each set. Taking into account that the image sets came from video clips, a video-based analysis is applied to select the image to be included in the final illuminant dataset. The frames which show redundancy in terms of visual content are removed and only the most representative ones are retained (see Ciocca & Schettini (2006a) and Ciocca & Schettini (2006b)). Applying this procedure the

original dataset was reduced to 1,135 images.

Note that the aim of the employed procedure is to decorrelate the *pictorial content* of the images. Decorrelation of the illuminant is not guaranteed. This fact, does not invalidate the evaluation strategy since the performance of illuminant estimation algorithms depends more on the content of the image than on the illuminant itself.

More details about this procedure can be found in Appendix A, while typical examples of the images belonging to the grayball dataset used are reported in figure 4.9.

Figure 4.9: Typical examples of the images belonging to the grayball dataset

4.3 Basic and benchmark color constancy algorithms considered

4.3.1 Simple algorithms

Several computational color constancy algorithms exist in the literature, each based on different assumptions. Recently Van de Weijer et al. van de Weijer *et al.* (2007a) have unified a variety of algorithms. These algorithms approximate the

illuminant color $\mathbf{I}$ in Equation4.1 by implementing instantiations of the following equation:

$$\mathbf{I}(n,p,\sigma) = \frac{1}{k}\left(\iint |\nabla^n \boldsymbol{\rho}_\sigma(x,y)|^p \,\mathrm{d}x\,\mathrm{d}y\right)^{\frac{1}{p}}, \tag{4.4}$$

where n is the order of the derivative, p is the Minkowski norm, $\boldsymbol{\rho}_\sigma(x,y) = \boldsymbol{\rho}(x,y) \otimes G_\sigma(x,y)$ is the convolution of the image with a Gaussian filter $G_\sigma(x,y)$ with scale parameter σ, and k is a constant to be chosen so that the illuminant color $\mathbf{I}$ has unit length. The integration is performed over all pixel coordinates. In this work, varying the three variables (n,p,σ) we have generated four algorithm instantiations that correspond to well known and widely used color constancy algorithms:

1. Gray World (GW) algorithm Buchsbaum (1980), which is based on the assumption that the average reflectance in a scene is achromatic. It can be generated setting $(n,p,\sigma) = (0,1,0)$ in Equation 4.4.

2. White Point (WP) algorithm Cardei *et al.* (1999), also known as Maximum RGB, which is based on the assumption that the maximum reflectance in a scene is achromatic. It can be generated setting $(n,p,\sigma) = (0,\infty,0)$ in Equation 4.4.

3. Shades of Gray (SG) algorithm Finlayson & Trezzi (2004), which is based on the assumption that the $p-$th Minkowski norm of a scene is achromatic. It can be generated setting $(n,p,\sigma) = (0,p,0)$ in Equation 4.4.

4. General Gray World (gGW) algorithm Barndard *et al.* (2002); van de Weijer *et al.* (2007a), which is based on the assumption that the $p-$th Minkowski norm of a scene after local smoothing is achromatic. It can be generated setting $(n,p,\sigma) = (0,p,\sigma)$ in Equation 4.4.

5. Gray Edge (GE1) algorithm van de Weijer *et al.* (2007a), which is based on the assumption that the $p-$th Minkowski norm of the first order derivative in a scene is achromatic. It can be generated setting $(n,p,\sigma) = (1,p,\sigma)$ in Equation 4.4.

6. Second Order Gray Edge (GE2) algorithm van de Weijer *et al.* (2007a), which is based on the assumption that the $p-$th Minkowski norm of the second order derivative in a scene is achromatic. It can be generated setting $(n, p, \sigma) = (2, p, \sigma)$ in Equation 4.4.

A seventh algorithm has been considered — the Do Nothing (DN) algorithm — which gives the same estimation for the color of the illuminant for every image, $\mathbf{I} = [1\ 1\ 1]$.

As can be noticed, the instantiations of GW and WP have all three parameters (n, p, σ) fixed; SG instead, has the parameter p that can be opportunely tuned for a particular image; while gGW, GE1 and GE2 have two parameters (p and σ) which must be tuned.

4.3.2 Combined algorithms

With the aim of improving the illuminant color estimation of the single algorithms, a linear and a nonlinear combining algorithm have also been implemented. The first one is the Least Mean Squares Committee (LMS) proposed by Cardei and Funt Cardei & Funt (1999). The response of this combining algorithm is given by multiplying the responses of all the n single algorithms considered for a fixed $3n \times 3$ weight matrix $\mathbf{W}$. Formally:

$$\mathbf{RGB}_{LMS} = [\mathbf{RGB}_1 \cdots \mathbf{RGB}_n]\, \mathbf{W}. \tag{4.5}$$

The second one is an instantiation of the No$-N-$Max (NNM) algorithm, which has been recently proposed by the authors and has shown to perform well on synthetic and real images Bianco *et al.* (2008c). The underlying idea is that algorithms that give similar illuminant color estimations have to be trusted more than algorithms that give estimates that are far from the others, and thus the latter ones have to be automatically discarded.

Let $\mathbf{rgb}_i = \mathbf{RGB}_i \cdot |\mathbf{RGB}_i\|^{-1}, i = 1, \cdots, n$ be the normalized versions of the illuminant color estimates given by the n single algorithms considered, and let $D_j = \sum_{i=1, i\neq j}^{n} d(\mathbf{rgb}_i, \mathbf{rgb}_j), j = 1, \cdots, n$ be the sum of the Euclidean distances of the illuminant color estimate of the algorithm from all the other estimates. Reorder the n algorithm estimates $\mathbf{rgb}_1, \cdots, \mathbf{rgb}_n$ as $\mathbf{rgb}_{p_1}, \cdots, \mathbf{rgb}_{p_n}$, where

$D_{p_1} \leq D_{p_n} \leq \cdots \leq D_{p_n}$. In other words, the estimates are reordered from the one with the smallest distance from all the others to the one with the highest distance from all the others. This combining method takes the mean value of the estimates of the single algorithms, automatically discarding the N estimates with the highest distance from all the others (i.e. the last N in the reordered sequence). Formally:

$$\mathbf{RGB}_{NNM} = \frac{\sum_{i=p_1,\cdots,p_{n-N}} \mathbf{rgb}_i}{n-N}, \tag{4.6}$$

which is finally normalized as defined before.

4.4 Performance Evaluation

In order to evaluate the performance of the algorithms considered, we have to define an error measure. Since in estimating the scene illuminant it is more important to estimate its color than its overall intensity, the error measure has to be intensity-independent. As suggested by Hordley and Finlayson Hordley & Finlayson (2004), the angle between the RGB triplets of the illuminant color ($\boldsymbol{\rho}_w$) and the algorithm's estimate of it ($\hat{\boldsymbol{\rho}_w}$) is used as error measure:

$$e_{ANG} = \arccos\left(\frac{\boldsymbol{\rho}_w^T \hat{\boldsymbol{\rho}_w}}{\|\boldsymbol{\rho}_w\| \|\hat{\boldsymbol{\rho}_w}\|}\right). \tag{4.7}$$

Hordely and Finlayson Hordley & Finlayson (2004) showed that a good descriptor for the angular error distribution is the median error.

To verify if the performances of different algorithms are statistically different, a test which is able to compare the whole error distribution of different algorithms is needed. Since standard probability models cannot represent underlying errors well, a test that does not make any a-priori assumptions about the underlying error distributions is needed. To compare the performance of two color constancy algorithms in addition to the median angular error, Wilcoxon Sign Test (WST) Wilcoxon (1945) has been used.

Comparing each color constancy algorithm with all the others, a score representative of the number of times that the null hypothesis H_0 has been rejected for the given algorithm can be generated. This score is the number of times that

the performance of the given algorithm has been considered to be better than the others.

4.5 Tuning of the basic color constancy algorithms

Most of the color constancy algorithms considered, (SG, gGW, GE1, GE2 and LMS), needed a training phase to opportunely tune the parameters (n, p, σ) or to calculate the weight matrix $\mathbf{W}$. Since a training set on which to perform the tuning of the parameters was needed, 340 images (the 30%) were randomly extracted from the 1,135 images of the illuminant dataset. Starting from the 340 training images, 40 have been discarded in order to balance the frequency of indoor and outdoor images, resulting in a total of 150 indoor images and 150 outdoor images. The performances of the algorithms are evaluated using the median angular error. Since the median error is a nonlinear statistic, a multidimensional nonlinear optimization algorithm was needed: the selected choice was to use a Pattern Search Method (PSM) Lewis & Torczon (1997, 2000).

4.6 Class-based color constancy

The idea investigated here is to see if the effectiveness of automatic illuminant estimation techniques may be improved if information about the content of the images is taken into account. To this end,an illuminant estimation approach which exploits the information provided by an image classifier was designed. As suggested by Szummer and Picard Szummer & Picard (1998), the indoor and outdoor classes were considered, as these classes correspond to categories of images with different content, usually taken under different illumination conditions. Therefore, these two classes of images may require different color processing procedures.

In this section four different strategies for illuminant estimation are derived and experimentally compared Bianco *et al.* (2008a,b):

- Class-Independent (CI): the same algorithm is applied without taking into account the image class. Among the available algorithms with the parameters optimized on a training set, the best one is chosen on the basis of a robust statistical test.

- Class-Dependent Parameterization (CDP): two instantiations of the same algorithm are used. They differ in the value of the parameters which are optimized for the individual classes. The best algorithm is selected on the basis of the statistical test of the performance on the whole training set. Given an unseen image, it is firstly classified as indoor or outdoor, and then processed with the algorithm tuned for that class.

- Class-Dependent Algorithms (CDA): for each class a different algorithm is applied. The parameters of each algorithm are optimized for the corresponding class. The best algorithm for indoor and the best algorithm for outdoor are selected on the basis of the statistical test of the performance on the corresponding subsets of the training set. Given an unseen image, it is firstly classified, and then processed with the algorithm selected for the predicted class.

- Class-Dependent Algorithms with Uncertainty Class (CDAUC): a rejection class is introduced to identify images on which the classifier is not confident enough. Therefore the images are classified as indoor, outdoor, or uncertain. Images classified as indoor or outdoor are processed according to the CDA strategy. Images classified as uncertain are processed according to the CI strategy.

The above strategies are independent from the illuminant estimation algorithms. Eight algorithms are considered(see sec.4.3): six derived from the framework recently proposed in van de Weijer *et al.* (2007a), and two obtained by a linear and a nonlinear algorithm combination.

4.6.1 Image Classification

There have been several efforts to automate the classification of digital images to date. Szummer and Picard Szummer & Picard (1998) have constructed algorithms for indoor/outdoor image classification. Vailaya *et al.* Vailaya *et al.* (2001) have considered the hierarchical classification of vacation images: at the highest level the images are sorted into indoor/outdoor classes, outdoor images are then assigned to city/landscape classes, and finally landscape images are classified in sunset, forest, and mountain categories. In Schettini *et al.* (2004) low-level visual features are related to semantic image categories, such as indoor, outdoor and close-up, using CART classifiers. Several classification strategies were designed and experimentally compared, producing a classifier that can provide a reasonably good performance and robustness. In this section a similar approach is used to classify the images into indoor and outdoor classes.

Each image is described by a set of low-level features related to color, texture and edge distribution. The extracted features, organized in a feature vector, are fed to a decision forest trained to distinguish between indoor and outdoor images. After a feature selection phase, which consisted in training several classifiers and evaluating them on an independent validation set, the five general-purpose features described in sec.4.1.1 were selected. The only modification has been made for the extraction of the information about color distribution, which is captured by spatial color moments: the images are transformed into the YCbCr color space, divided into seven horizontal bands, and the mean and the standard deviation of each of the three color bands are computed. The subdivision in horizontal bands adequately describes some characteristics which are very useful for indoor/outdoor classification (images with blue sky in the upper part, or green grass in the lower part...). Table 4.1 summarizes the features considered.

For classification, decision trees built according to the CART methodology Breiman *et al.* (1984) are used. The CART approach to classification presents several advantages: first of all it is a non-parametric and non-metric method so that no a-priori knowledge about the distribution of the values of the features is needed and the issue of feature normalization may be ignored. The hierarchical structure of the trees is rather easy to analyze making it possible to understand

Table 4.1: Summary of the features used to describe the images for the class-based classification.

Name	No. components	Category
YCbCr color moments	$2 \times 3 \times 7 = 42$	Color
RGB color histogram	27	Color
Edge direction histogram	18	Edges
Wavelet statistics	20	Texture

which features play a major role in the classification process. Moreover, CART trees have been previously applied with satisfactory results to other image classification problems Schettini *et al.* (2002a, 2003, 2004).

In CART methodology the size of a tree is treated as a tuning parameter, and the optimal size is adaptively chosen from the data. A very large tree is grown and then pruned, using a cost-complexity criterion which governs the tradeoff between size and accuracy. Although the pruning process prevents overfitting, pruned trees still present instability (a small change in data may result in a very different tree). Decision forests can be used to overcome this problem improving, at the same time, generalization accuracy Breiman (1996). The trees of a decision forest are generated by running the training process on bootstrap replicates of the training set. The classification results produced by the single trees are combined by majority vote. The number of concordant votes may also be used as a measure of confidence of the combined classifier on the classification result.

4.6.2 Algorithm selection approaches

Two algorithm selection approaches are proposed. In the first one, the output of the classifier is used to decide which color constancy algorithm (and/or which set of parameters) to apply to the specific image under consideration as shown in Figure 4.10. This approach is used in the CDP and CDA strategies. In the second selection approach, the confidence measure provided by the classifier is taken into account as shown in Figure 4.11. The difference with respect to the first one is the introduction of an uncertainty class constituted by the images for

which the classifier is not sure about their membership to indoor or outdoor class (i.e. CDAUC strategy).

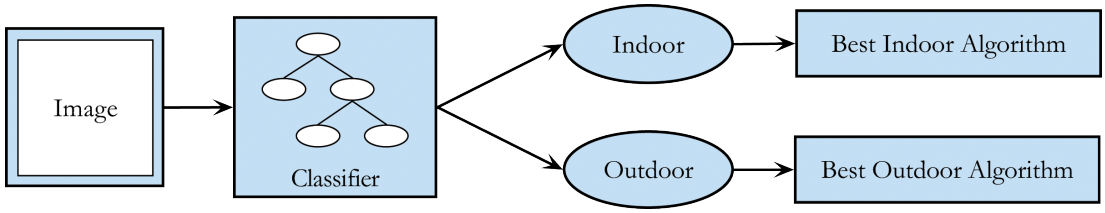


Figure 4.10: Scheme of the first proposed approach for the selection of the algorithms. The image is classified by the decision forest and the output class is used to select the appropriate color constancy algorithm.

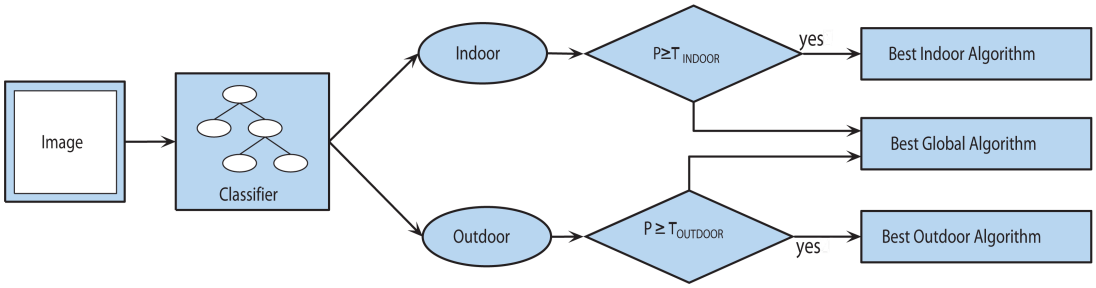


Figure 4.11: Scheme of the second proposed approach for the selection of the algorithms. The image is classified by the decision forest and the appropriate color constancy algorithm is selected according to the output class and the confidence measure.

Let P be the confidence measure, that is, the fraction of concordant votes in the trees in the forest w.r.t. the output class. Given an input image, it is considered to be indoor if the predicted class is indoor and $P \geq T_{INDOOR}$; the image is considered outdoor if the predicted class is outdoor and $P \geq T_{OUTDOOR}$; the image is considered to be in the uncertainty class otherwise. Two different thresholds are chosen because the classifier adopted does not guarantee a uniform confidence measure for both classes. The two thresholds are tuned by analyzing the final performance of the illuminant estimation algorithms.

If the input image is classified as indoor or outdoor and satisfies the constraints above, the image is processed with the best color constancy algorithm for that

class. Otherwise, since the membership of the image to either class cannot be reliably inferred, applying a specific algorithm tuned for indoor or outdoor images may significantly worsen the appearance of the image processed. In these cases, a more conservative, general-purpose algorithm is applied instead.

4.6.3 Experimental Setup

In the proposed approach the classifier needs to be trained, using a set of images manually labeled as indoor or outdoor, and the parameters of some color constancy algorithms need to be tuned. The last point requires a dataset of images labeled with ground truth illuminants, and the definition of a suitable error measure which can quantify the accuracy of the algorithms in the estimation of the illuminant. In the following sections the datasets and the setup procedures are introduced.

4.6.3.1 Decision Forest Training

In order to obtain a classifier with good generalization capabilities, a rather large dataset of images is needed. For this purpose, 6,785 images were collected, downloaded from the web, or acquired by a scanner or digital cameras. All the material varied in size, resolution, and quality. The images were resized to 256 pixels on the largest dimension, and proportionally on the other dimension in such a way that the aspect ratio was maintained. The images have been manually annotated yielding to 2,105 indoor images and 4,680 outdoor images. No enhancement procedure (such as white balancing) has been applied to the images.

In order to select the features and to determine the size of the forest we partitioned the images into a training set of 2,000 images (1,000 indoor and 1,000 outdoor), and a validation set containing the remaining 4,785 images. As a result, the four features in Table 4.1 Section 2.1 were selected and the size of the decision forest set to 50 trees.

Within this framework a classification accuracy of about 93.1% on the validation set was obtained. On the images of the illuminant dataset (Section 3.1) the results summarized in Table 4.2 were obtained. Note that the test images have

been manually annotated as indoor (481 images) or outdoor images (654 images).

Table 4.2: Confusion matrix obtained on the test set by the decision forest. The number of misclassifications was 169 (86 indoor and 83 outdoor images).

	Predicted indoor	Predicted outdoor
True indoor	82.1%	17.9%
True outdoor	12.7%	87.3%

The overall classification accuracy obtained on the test set was 85.1%. The difference with respect to the performance obtained on the validation set can be explained considering that the test set includes several images with little information about the context in which they were taken. For instance, the test set includes several close-ups of various objects, which cannot be classified as indoor or outdoor without exploiting high level knowledge and reasoning. Figure 4.12 shows a sample of the misclassified images.

4.6.4 Automatic Parameters Tuning

The parameters found by the pattern search algorithm are reported in Table 4.3.

It can be seen that the optimal values found for the parameter p for the indoor class are lower than the ones for the outdoor class. The optimal values found for the parameter σ instead, are higher for the indoor class than the ones for the outdoor class. While for the indoor class the pattern search optimization found different values for the parameters of the different color constancy algorithms, we can see that in the choice of the parameters for the outdoor class of the gGW and SG algorithms tended to be asymptotically convergent to the WP's ones. Regarding the general-purpose class, we can see instead that the optimal choice for the parameters of the SG algorithm is very similar to the GW's ones.

Two combining algorithms are also considered, the No-2-Max (N2M) Bianco *et al.* (2008c) and the Least Mean Squares Committee (LMS) Cardei & Funt (1999). The first one does not require any training process, while the weight

Figure 4.12: Example of indoor images misclassified as outdoor images (a-l); and some outdoor images misclassified as indoor images (m-x).

matrices for the LMS need to be computed on the training set using a least squares regression. Thus, for each of the classes considered, we have a different weight matrix for the LMS Committee which multiplies the responses of the six algorithms with the tuned parameters for that class.

4.6.5 Experimental Results

The focus of the experimentation is to establish which of the following color constancy strategies is preferable (at least on the dataset considered):

- Class-Independent (CI): the parameters of the algorithms are tuned without considering the class of input images (see the "general purpose" column in Table 4.3).

Table 4.3: The parameters found by the pattern search algorithm. Only the values reported in bold have been computed, the others have been set according to the definition of the algorithms.

	Indoor			Outdoor			General purpose		
	n	p	σ	n	p	σ	n	p	σ
GW	0	1	0	0	1	0	0	1	0
WP	0	∞	0	0	∞	0	0	∞	0
SG	0	**1.27**	0	0	$\boldsymbol{\infty}^*$	0	0	**1.06**	0
gGW	0	**1.32**	**1.00**	0	$\boldsymbol{\infty}^*$	**0.00**	0	**1.08**	**0.83**
GE1	1	**0.60**	**1.72**	1	**1.10**	**0.83**	1	**1.10**	**1.08**
GE2	2	**1.06**	**2.96**	2	**1.91**	**0.04**	2	**1.55**	**1.83**

* Values which diverge towards infinity.

- Class-Dependent Parameterization (CDP): for a given algorithm the parameters to use are selected on the basis of the class predicted by the decision forest (see the "indoor" and "outdoor" columns in Table 4.3).
- Class-Dependent Algorithms (CDA): for each class the best algorithm (and its corresponding parameters) is selected.
- Class-Dependent Algorithms with Uncertainty Class (CDAUC): the same as above, but with the introduction of the uncertainty class. Images falling in the uncertainty class are processed by the algorithm that has proved to be the best class-independent algorithm.

In order to perform such a comparison Hordley and Finlayson Hordley & Finlayson (2004) showed that together with the summary statistics of the angular errors a test able to compare the whole error distribution between different algorithms is needed. Since standard probability models cannot represent the underlying errors well, a test that does not make any a-priori assumptions about the underlying error distributions is needed. In this work, to compare the performance of two color constancy algorithms, the Wilcoxon Sign Test (WST) Wilcoxon (1945) with a significance level $\alpha = 0.01$ has been used.

In order to select the best algorithms for each class, the algorithms are evaluated on the training set introduced in Section 5.3. The results are reported in Table 4.4. More in detail, the first column of the table reports the results obtained on the indoor images of the training set by the algorithms whose parameters have been tuned for that specific class (that is, the class-dependent parameterization strategy). Similarly, the second column summarizes the results obtained on outdoor images. The third column details the results of the general purpose version of the algorithms (that is, the class-independent strategy) when applied to the whole training set.

Table 4.4: Median angular error obtained by the color constancy algorithms on the training set. The best results for each column are reported in bold.

	Indoor Images†		Outdoor Images†		Whole Training Set	
	Median	WSTs	Median	WSTs	Median	WSTs
GW*	4.91	3	7.86	0	5.62	1
WP*	11.83	0	2.81	2	7.76	0
SG	**4.31**	**6**	2.81	2	5.56	1
gGW	4.32	**6**	2.81	2	5.57	1
GE1	5.40	1	3.72	1	5.45	1
GE2	5.57	1	**2.48**	**7**	5.47	1
N2M	5.13	3	2.83	2	5.02	6
LMS	4.58	5	2.71	2	**4.50**	**7**

* Algorithms with fixed, class independent, parameters.

† Test with algorithms tuned specifically for the class.

The results with the general purpose algorithms show that combinatorial algorithms, and in particular LMS, performed better than the others. The Wilcoxon test confirms that the difference in performance is significant (even the difference of 0.43 degrees between N2M and GE1). However, if only class-specific algorithms are considered, SG and GE2 outperform the combining methods for indoor and outdoor images respectively.

On the basis of the results obtained it is possible to select the best algorithm for each class. For indoor images both SG and gGW may be chosen; in fact, the

WST score shows that they exhibit an indistinguishable behavior. We selected the SG because its angular error is slightly better, and it is computationally less expensive than the gGW. For outdoor images the GE2 algorithm is clearly superior to the others.

It can be seen that the errors on the two classes are very different. In particular, with the exception of the GW algorithm, all the errors obtained on the outdoor class are significantly lower than those obtained on the indoor class. This can be explained by analyzing the dataset. The majority of the outdoor images were shot under a near ideal illumination condition (clear sky without any color cast). Moreover, the outdoor illuminants do not present the same variability of the indoor illuminants. Looking at the parameters and results of the WP, SG, and gGW algorithms on the outdoor images, it can be seen that they behave identically as a WP algorithm. This can be explained considering that real outdoor images tend to exhibit color channel clipping in the high intensity range. These very bright pixels are taken as reference white by the three WP-like algorithms and this reference is often very close to the real scene illuminant.

Table 4.5 summarizes the results on the 835 images (331 indoor and 504 outdoor) in the test set. The first column reports the results obtained by the algorithms using the class-independent strategy while the second reports those using the class-dependent parameterization strategy. For the sake of brevity we omit the results obtained on the two classes. The errors of the GW and WP algorithms are the same for both strategies because their parameters are actually class independent. The results of the CI strategy are lower than those obtained on the training set. This is due to the fact that in the test set the outdoor class is more represented than the indoor class (504 vs. 331 images). The introduction of a class-dependent parameterization, improved the results of all the algorithms (with the obvious exception of GW and WP). Four different algorithms (SG, GE2, N2M and LMS) obtained a median error of about four degrees, which is better than the error obtained by the best class-independent algorithm.

In the last two experiments we considered the application of different algorithms, on the basis of the classification outcome (namely the CDA and CDAUC strategies). According to the results obtained on the training set, the CDA strategy consists in applying the SG algorithm to the images classified as indoor, and

Table 4.5: Median angular error obtained by the color constancy algorithms on the test set using the CI and CDP strategies. The best results for each column are reported in bold.

	CI Strategy		CDP Strategy	
	Median	WSTs	Median	WSTs
GW*	5.95	0	5.95	0
WP*	5.48	3	5.48	1
SG	5.80	0	4.08	**4**
gGW	5.80	0	5.39	1
GE1	4.47	5	4.32	3
GE2	4.65	5	**3.94**	**4**
N2M	4.79	4	4.01	**4**
LMS	**4.18**	**7**	4.05	**4**

* Algorithms with fixed, class independent, parameters.

the GE2 algorithm to those classified as outdoor. The CDAUC strategy consists in applying the same algorithms as the CDA but adding the uncertainty class for which we used the general purpose version of the LMS algorithm. The two thresholds of the CDAUC strategy were chosen using a five-fold cross validation approach on the 300 images of the training set. The final thresholds ($T_{INDOOR} = 0.82$, $T_{OUTDOOR} = 0.67$) are the average of the best thresholds found by each cross validation iteration. Using these values about 19.75% of the images in test set were classified as "uncertain". The results obtained on the test set are reported in Table 4.6. They are compared with the performance of the best algorithms in the CI and CDP strategies (LMS and GE2, respectively). It can be noted that the median error decreases as more complex strategies are applied. However, according to the Wilcoxon Sign Test, it is not possible to claim that class dependent strategy is superior to class-dependent parameterization. The results obtained demonstrate that, at least on the dataset considered, a classification based strategy which uses an uncertainty class outperforms a general purpose algorithm.

Table 4.6: Summary of the results obtained on the test set by the four strategies proposed.

Strategy	Underlying algorithms	Median	WSTs
CI	LMS, general purpose parameters	4.18	0
CDP	GE2, indoor and outdoor parameters	3.94	1
CDA	SG for indoor and GE2 for outdoor	3.78	1
CDAUC	SG ind., GE2 out., LMS gen. purpose	**3.54**	**3**

To determine the influence of the performance of the classifier on the final color constancy performance, the results obtained with the CDA strategy considering correctly classified and misclassified images are considered separately. On correctly classified indoor images the median angular error is 4.85; on indoor images misclassified as outdoor we obtained an error of 9.79. For outdoor images, the median errors on correctly classified and misclassified images are 2.31 and 5.07 respectively. Thus, image misclassification approximately doubles median angular errors. To avoid this decrease in performance an accurate classifier is crucial so that misclassifications rarely occur. To assess how much angular error may be improved using a better classifier, the obtained results are compared with those of an "optimal" classifier (i.e. a classifier which correctly classifies all test images). By running the CDA strategy with the optimal classifier we obtained a median angular error of 3.48. We also considered a "random" classifier (i.e. a classifier which randomly misclassifies the images with a probability of 0.5) obtaining an error of 5.63. It is possible to note that the results obtained using the implemented classifier (3.78 of median angular error) are much closer to the results of the optimal classifier than to the random classifier ones. The results of this experiment give the upper bound of 0.3 degrees of angular error to the improvement that could be achieved by adopting a more powerful classification methodology (Support Vector Machines Cortes & Vapnik (1995), boosting Shapire (1999), ...).

The obtained results of the experimentation can be summarized as follows:

- if no knowledge of the image content is exploited (CI strategy), combining methods perform better than the single ones.

- The algorithms that can be tuned on the basis of image contents benefit by the classification process.
- For the indoor class the SG (or equivalently the gGW) shows better results than the other methods. For outdoor class the best performance is obtained by the GE2 algorithm. For the specific classes, combining methods do not seem to be the best choice.
- When the same algorithm is used on both classes but with different parameters settings (Class-Dependent Parameterization strategy), there is not a single best algorithm. Four algorithms performed equally well obtaining the best result. Among these, being the least computational expensive, SG seems to be the best choice.
- Using different algorithms for indoor and outdoor images (Class-Dependent Algorithm strategy), improves the results with respect to the Class-Independent strategy. From our experiments, the best combination of algorithms consisted in the selection of the SG algorithm for the indoor images and GE2 for the outdoor images. We also observed a small improvement with respect to the Class-Dependent Parameterization strategy.
- The introduction of a third image class containing the images on which the classifier is not confident enough (CDAUC strategy), further improves the results. The algorithms selected in the CDAUC strategy are the SG for indoor images, GE2 for outdoor images and LMS (with the general purpose parameterization) for the other images. The improvement is statistically significant with respect to the other strategies considered.

4.7 Feature-based color constancy

In this section a classification approach which improves the performance of existing color constancy algorithms is proposed. Since it is difficult to select an exhaustive and comprehensive set of image categories (using either supervised or unsupervised classification), the proposed approach does not classify the images into high level categories and then process each image with an ad-hoc algorithm

for that class, but it learns from the images themselves some intrinsic, low level properties that can be used to drive the selection of the best algorithm for that image. That is, the selection of the algorithm is not class-based but feature-based.

To select the algorithm to be used with a given image, a decision forest tree approach was used. This approach is based on the CART methodology where each classifier in the forest votes for one of the illuminant estimation algorithms to be used. The most voted algorithm is then applied to the input image. In the training of the classifier the error costs of the erroneous algorithm's selection are considered. That is, the algorithm selected by the classifier is the one that minimizes the expected error in the illuminant estimation. A combination approach that estimates the illuminant as a weighted sum of the algorithms' estimations is also designed. For each algorithm, its weight is computed using the votes of the classifiers in the forest and is proportional to the number of classifiers that have selected that algorithm. The features used in the classification process are heterogeneous and representative of the image content and can be related to visual properties of the images such as color, texture, composition, etc... For the most part they are taken from the content-based image retrieval research field and have been selected because they are widely used in different applications. A few features have been added, which are related to the illuminant estimation problem that may be helpful in the algorithm's selection process.

4.7.1 Image classification

The choice to adopt classification trees as classifiers is motivated by the fact that they present several advantages which make them particularly suitable for the problem considered:

- classification trees can be trained to distinguish an arbitrarily high number of classes, so that we are free to consider any set of illuminant estimation algorithms;
- they allow feature vectors to be composed of several heterogeneous features since they do not require any feature normalization or decorrelation;

- classification trees can exploit information about the a-priori probabilities of the classes and their misclassification costs, making it easy to integrate information about the correlation between the errors of different algorithms in the algorithm selection model.

In our approach, decision trees are built according to the Classification and Regression Trees (CART) methodology Breiman *et al.* (1984) which has proven to be effective for image classification tasks Schettini *et al.* (2002a, 2004).

In the proposed approach, the classifier is used to select the most appropriate illuminant estimation algorithm on the basis of the content of the images. Illuminant estimation algorithms are modeled as classes: an image is of class j if the j-th algorithm is the one which produces the lowest estimation error on that image among the algorithms considered. The feature vector which describes each image is built by computing a set of low-level features (see Section 4.1) and by concatenating the values obtained.

The straightforward application of the CART training process to this problem leads to poor results. This is due to the fact that some properties of the problem are not taken into account in the formulation:

- some algorithms generally perform better than others;
- the performance of the algorithms are correlated so that the consequences of a non-optimal choice may present a high variability.

The first point is addressed by estimating the a-priori probability for each algorithm that it is the best algorithm. They are estimated as the ratio N_j/N, where N_j is the number of training cases for which the j-th algorithm is the best choice and N is the size of the training set. The a priori probabilities are then used during training to compute the resubstitution estimates of conditional probabilities in (3.1), (3.2), and (3.3).

For each pair of algorithms, class correlation is modeled by considering, the average difference in performance obtained when one of the two algorithms corresponds to the best choice:

$$c(k|h) = \frac{\sum_{j:y_j=h} e_j^{(k)} - e_j^{(h)}}{|\{j : y_j = h\}|}, \quad h, k \in \{1, \ldots, K\}, \tag{4.8}$$

where $e_j^{(k)}$ is the error of the k-th algorithm on the j-th training sample. In other words, $c(k|h)$ is the expected cost (i.e. degradation in performance) caused by the choice of algorithm k when algorithm h is the best choice.

Misclassification costs are used during training to influence pruning and label assignment. Equation (3.2) is replaced by:

$$\hat{y} = \arg\min_{j \in \{1,\dots,K\}} \sum_{h=1}^{K} c(j|h) P(y = h|L), \tag{4.9}$$

while $R(T)$ is now the estimated average cost of the decisions of tree T:

$$R(T) = \sum_{L \in \tilde{T}} \left(\min_{j \in \{1,\dots,K\}} \sum_{h=1}^{K} c(j|h) P(y = h|L) \right) P(L), \tag{4.10}$$

where $\tilde{T}$ is the set of the leaves of T and $P(L)$ is the resubstitution estimate of the probability that a case falls in the leaf L.

Class correlation is also exploited by using the *twoing* criterion as the impurity function: the impurity of a node is computed by dividing the set of the classes into two macro-classes, and then by applying the Gini diversity index to the distribution of the two macro-classes. In practice, the twoing criterion is implemented by substituting (3.1) with the following expression:

$$\Delta I_{twoing}(j, \tau) = \frac{P_L P_R}{4} \left[\sum_{k=1}^{K} |P(y = k|T_L) - P(y = k|T_R)| \right]^2. \tag{4.11}$$

The effect of the twoing criterion is that during the first steps of the tree growing process few resources are wasted trying to discriminate between similar classes (i.e. highly correlated algorithms). The discrimination of such classes occurs only near the leafs.

Table 4.7 summarizes the features considered.

4.7.2 Experimental setup

The color constancy algorithms considered are an independent subset of the ones described in Section 4.3.1: DN, GW, WP, GE1, GE2. The parameters used are the same as in Table 4.3 in the column labeled general purpose. The training set has now also been used to:

Table 4.7: Summary of the features used to describe the images for the feature-based classification.

Name	No. components	Category
YCbCr color moments	6	Color
RGB color histogram	27	Color
Number of colors	1	Color
Cast indexes	2	Color
Color clipping	8	Color
Edge magnitude histogram	5	Edges
Edge direction histogram	18	Edges
Wavelet statistics	20	Texture

- make an estimate of the a-priori related to the algorithms (i.e. the probability that an algorithm is the best one);
- estimate the matrix of misclassification costs (4.8).

A cross validation on the test set has been adopted to train and evaluate the decision forest and to assess the overall performance of the strategy.

4.7.2.1 Decision forest training

To train the decision forest and to evaluate the performance of the proposed strategy a cross validation procedure is adopted. First, the five illuminant estimation algorithms are applied to the whole dataset and their angular errors are computed. The first step of the training process for the decision forest consists of the estimation of the a-priori probability for each algorithm that is the best choice, and of the matrix of misclassification costs. These values, estimated on the 340 images of the training set, are reported in Table 4.8 and in Table 4.9. In more than one third of the images, the Gray World algorithm corresponds to the best choice. In another 30% of the cases the images are already well balanced, and thus the "Do Nothing" algorithm produces the most accurate illuminant estimation. The remaining one third of the images are best processed by the GE1

Table 4.8: A-priori probabilities, corresponding to the five illuminant estimation algorithms, estimated on the images of the training set.

Algorithm	Probability
DN	0.33
GW	0.34
WP	0.04
GE1	0.12
GE2	0.17

Table 4.9: Matrix of the misclassification costs estimated on the images of the training set (4.8).

	Predicted Algorithm				
Best Algorithm	DN	GW	WP	GE1	GE2
DN	0.00	10.90	1.98	6.41	4.10
GW	8.43	0.00	5.67	4.13	6.28
WP	0.50	10.19	0.00	4.93	2.68
GE1	2.80	5.48	2.29	0.00	0.77
GE2	2.86	6.18	1.89	0.67	0.00

or the GE2 algorithms. For less than 4% of the images the WP algorithm corresponds to the best choice. The matrix of misclassification costs tell us that the results of the GE1 and GE2 algorithms show a high correlation. Detecting the images for which the GW algorithm should be used is crucial. In fact, in these cases the errors of the other algorithms are rather high.

At this point, a ten-fold cross validation is used to train and to evaluate the proposed strategy. The test set is randomly partitioned into ten subsets. Then a decision forest composed of 30 classification trees is trained on all the images of the dataset (including the training set) with the exclusion of a single subset of the test set, which is finally used to measure the performance of the decision forest. The procedure is repeated ten times, one for each subset of the test set. The results of the ten forests are finally merged.

Figure 4.13 reports the distribution of the occurrences of the features within

the splits of the ten forests. All the features have been used. The RGB histogram is used in more than 30% of the splits; however, it must be considered that the RGB histogram is the feature with the highest number of components. Very compact features, such as the number of colors and cast indexes are important as well, if we consider that they are formed by only one and two components, respectively.

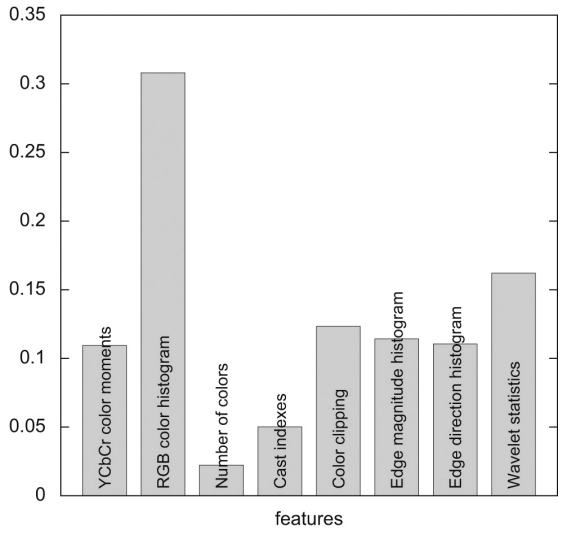


Figure 4.13: Histogram of the occurrences of the features in the splits of the trained trees.

4.7.3 Experimental results

Table 4.10 shows the confusion matrix obtained on the test set. Each row corresponds to an algorithm and reports the distribution of the output of the decision forest estimated on the subset of the test set for which that algorithm is the best choice. Most of the images for which the DN algorithm is the best choice are correctly classified (85% of accuracy). For the other algorithms the correct classification rate ranges from 61% (GW) to 11% (WP). However, considering the a-priori distribution of the five algorithms, the aggregated classification accuracy is about 55%, as shown in Figure 4.14 which reports the histogram of the rank corresponding to the choice of the decision forest on the test set. The best

Table 4.10: Confusion matrix of the decision forest used for algorithm selection, estimated on the images of the test set.

	Predicted Algorithm				
Best Algorithm	DN	GW	WP	GE1	GE2
DN	0.85	0.06	0.01	0.04	0.04
GW	0.24	0.61	0.01	0.10	0.05
WP	0.37	0.00	0.11	0.37	0.15
GE1	0.39	0.29	0.04	0.17	0.11
GE2	0.45	0.15	0.02	0.13	0.26

algorithm is chosen 55% of the time, the second best algorithm is chosen 11% of the time; and the frequency of the selection of the third, the fourth, and the worst algorithm are 16%, 12%, and 5%, respectively. It should be considered that the forest has not been trained with the aim of finding the best algorithm, but with the aim of finding the algorithm with the lowest expected error, taking into account the errors determined by misclassifications. This means that the performance of the decision forest should not be evaluated in terms of classification accuracy, but in terms of the angular error of the selected algorithms. Figure 4.15 reports the distribution of the loss determined by the choice of the decision forest with respect to the best algorithm. In more than 70% of the test cases this loss is below one degree of angular error with respect to the best algorithm. The average angular error of our Classification-based Algorithm Selection (CAS) Bianco *et al.* (2009b) strategy is about 4.76 degrees, while the median angular error is about 3.21 degrees. These results are compared in Table 4.11 with those obtained by the five single algorithms. A comparison with the results of a semantic driven approach Bianco *et al.* (2008b) (on the same data) is also reported. The performance of the CAS approach is clearly superior to that of the single algorithms and of the semantic-based approach, at least on the dataset considered. It is interesting to note that DN is the worst algorithm on average. However, in about 33% of the cases it is the best choice. This means that in the remaining 67% of the images its error is very high. Thus, from a color correction point of view, detecting which images need to be corrected and which do not is crucial. The

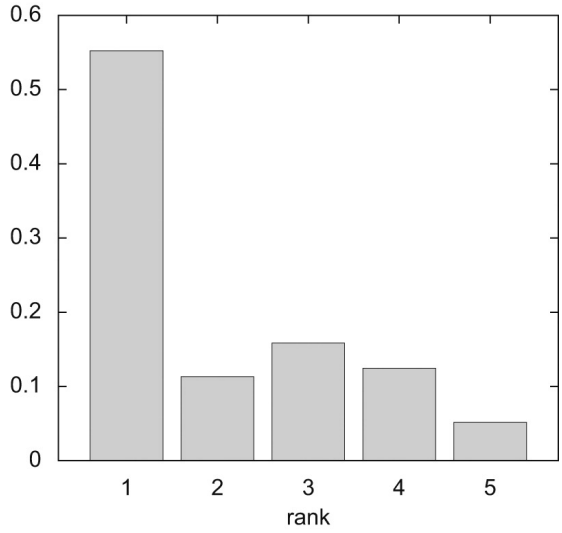


Figure 4.14: Distribution of the rank of the algorithm selected by the decision forest on the images of the test set. The first bar represents the fraction of test images for which the best algorithm is selected; the second bar represents the fraction of cases in which the second best is selected, and so on.

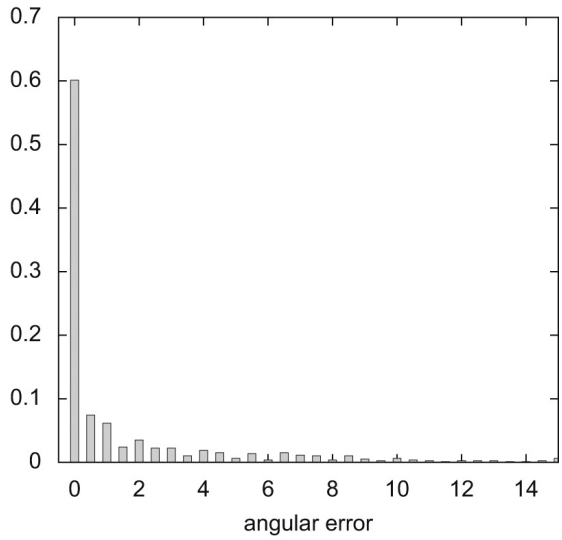


Figure 4.15: Distribution of the difference in angular error between the algorithms selected by the decision forest and the best choice for each image of the test set.

proposed selection strategy seems quite effective in doing this (see the confusion matrix in Table 4.10).

Table 4.11: Summary of the results obtained on the test set by the Classification-based Algorithm Selection (CAS) strategy, compared with the performance of the five simple algorithms and with the results obtained by an algorithm selection strategy based on semantic classification Bianco *et al.* (2008b). The best score for each column is reported in bold.

Algorithm	Median	Mean	WSTs
DN	6.05	8.07	0
GW	5.95	7.27	0
WP	5.48	7.45	2
GE1	4.47	5.84	4
GE2	4.65	6.23	3
CAS	**3.21**	**4.76**	**6**
Semantic	3.54	4.89	5
Ideal classifier	2.31	3.27	–

In order to determine which part of the error is due to the illuminant estimation algorithms and which part should be accounted to classification errors, the proposed selection strategy is compared with a strategy based on an ideal classifier. The ideal classifier selects the best algorithm for each image among the five considered. Using the ideal classifier a median angular error of 2.31 degrees on the test set is obtained. This means that the performance of the proposed strategy may be improved up to 0.9 degrees of median angular error.

In order to obtain better illuminant estimations, a common approach is to combine the results of several different algorithms. Cardei and Funt Cardei & Funt (1999) obtained good illuminant estimation by combining the results of Gray World, White Patch and neural net methods, considering both linear and non-linear committee methods. Schaefer et al. Schaefer *et al.* (2005) introduced a combined physical and statistical color constancy algorithm that integrates the statistics-based Color by Correlation method with a physics-based technique, based on the dichromatic reflectance model, using a weighted combination of their likelihoods for a given illumination set and taking the maximum likelihood entry.

In the Classification-based Algorithms Combination (CAC) strategy Bianco

et al. (2009c) the five algorithms are linearly combined, using as weights the consensus of the decision forest; that is, each algorithm is weighted proportionally to the number of trees which "voted" for it. The number of votes is, in fact, related to the error of the algorithms, as shown in Figure 4.16 which reports the average angular error (on the test set) obtained by the algorithms as a function of the votes that they received. The higher the number of votes is, the lower the error is. Table 4.12 reports the results obtained by the combination strategy and com-

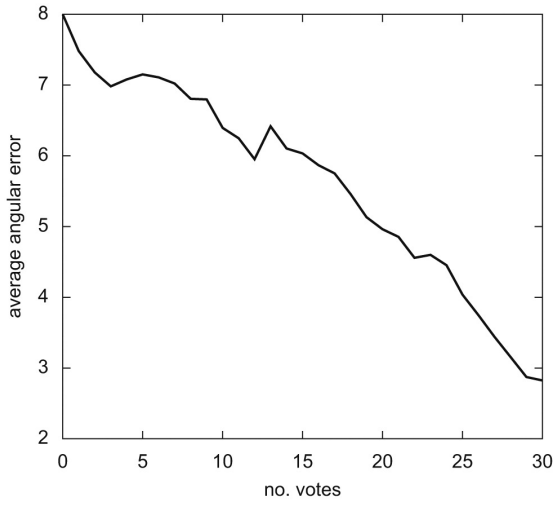


Figure 4.16: Average angular error obtained by the five illuminant estimation algorithms on the images of the set, as a function of the number of votes received by the trees of the decision forest.

pares it with other combining methods. The first combining method considered (AVG), simply averages the results of the estimations given by the five algorithms considered Cardei & Funt (1999). The second one (LMS) is a weighted average of the outputs of the individual algorithms Cardei & Funt (1999). The weights were optimized in the Least Mean Squares sense. This combining method was trained and tested using the same ten-fold cross validation used before. The last combining method considered (N2M), averages the outputs of the three individual algorithms which gave the closest illuminant estimations, automatically excluding the two that gave the furthest estimations Bianco *et al.* (2008c). Combining is more effective than simple selection, in fact, it obtained a median angular error

Table 4.12: Summary of the results obtained on the test set by the Classification-based Combining (CAC) strategy compared with the performance of other popular combining methods. The best score for each column is reported in bold.

Algorithm	Median	Mean	WSTs
AVG	4.66	5.99	0
LMS	4.12	5.29	2
N2M	4.79	5.82	0
CAC	**3.04**	**4.46**	**3**

of 3.04 degrees (versus 3.21 of the selection strategy). Simple combining methods are clearly outperformed.

4.8 Conclusions

In Table 4.13 the results obtained on the test set by the proposed classification based color constancy algorithm compared with the performance of other popular combining methods are reported. The best values for each column are reported in bold.

It is possible to notice that the feature-based approaches clearly outperform the class-based ones. Furthermore, looking at the WST scores we can say that this difference in performance is statistically significant.

Table 4.13: Summary of the results obtained on the test set by the proposed classification based color constancy algorithm compared with the performance of other popular combining methods. The best score for each column is reported in bold.

Algorithm Type	Algorithm Name	Median	Mean	WSTs
Simple	DN	6.05	8.07	0
	GW	5.95	7.27	0
	WP	5.48	7.45	4
	GE1	4.47	5.84	8
	GE2	4.65	6.23	7
	SG	5.80	7.03	2
	gGW	5.80	7.01	2
Combining	AVG	4.66	5.99	5
	LMS	4.12	5.29	9
	N2M	4.79	5.82	5
Class-based	CDA	3.78	5.04	10
	CDAUC	3.54	4.89	11
Feature-based	CAS	3.21	4.76	12
	CAC	**3.04**	**4.46**	**13**

Chapter 5

Color accuracy under varying illuminant estimation precision

The color rendering accuracy of digital imaging acquisition devices is a key factor to the overall perceived image quality Ramanath *et al.* (2005). In order to render the acquired image as close as possible to what a human observer would have perceived if placed in the original scene, the first stage of the color correction pipeline reported in eq.2.5 aims to emulate the color constancy feature of the human visual system (HVS), i.e. the ability of perceive relatively constant colors when objects are lit by different illuminants Hordley (2006). The second stage of the color correction pipeline transforms the image data into a standard RGB color space. This transformation, usually called color correction matrix or color matrixing, is needed because the spectral sensitivity functions of the sensor color channels rarely match those of the desired output color space (usually sRGB Stokes *et al.* (1996)).

Typically this transformation is a 3-by-3 matrix with 9 entries to be optimally determined.

There are several different methods to find the best color space conversion matrix, spacing from interpolation-based methods Amidror (2002); Hung (1988, 1991) to best fitting methods, from algebraic Hubel *et al.* (1997) to optimization-based methods Bianco *et al.* (2007). The color matrixing, is often defined as the best linear least squares mapping for the particular calibration set of patches under a given illuminant. The least-squares regression optimally maps camera raw

RGB values to XYZ values so that the residual squared error for the calibration data set is minimized Seber (1977). Given the importance of white (and also the gray-scale) in color reproduction, Finlayson and Drew have developed the White Point Preserving Least Squares fit procedure Finlayson & Drew (1997). As the name suggests, WPPLS is a method for determining the best least-squares transformation that maps the RGB values to the XYZ values subject to the constraint that the RGB corresponding to the white point is mapped without error. We underline that any constrained mapping is no longer optimal in the usual sense, since it produces larger errors (in terms of residual error) for any set of calibration patches used.

The Least Squares method (LS) uses a full rank matrix (rank-3, i.e. 9 degrees of freedom, DoF), while a constrained LS method only uses a rank-2 (i.e. 6 DoF) matrix in order not to affect the exact mapping of the gray scale. The constrained regression method could be extended to preserve two different colors. In this case the transformation matrix would be restricted to account for the best least-squares fit in a single direction. This direction is orthogonal to the plane spanned by the two vectors defining the color constraints. As expected, in this case the color transformation result is worse than in the case of a single color constraint. One could try to exactly map three colors, but working with a 3-by-3 transformation matrix, there would not be any free dimension to apply LS fitting. In order to satisfy three constraints, one has to define them as the principal components of the patches under consideration M.J. Vrhel (1992). With the intent of adding more information to the mapping, Vrhel and Trussel have developed the Non-Maximum Ignorance method. This method exploits not only the spectral sensitivities of the device, but also the specific nature of a real reflectance set M.J. Vrhel (1993).

Optimization-based methods Bianco *et al.* (2007) give the possibility of using even non-linear and non-derivable functionals to find the transformation matrix. This comes at the cost of no longer having a closed-form solution, and thus iterative optimization methods are required.

The color correction matrix transformation is usually optimized assuming that the illuminant in the scene has been successfully estimated and compensated for Bianco *et al.* (2007); Hubel *et al.* (1997). Both the illuminant estimation

process and the color correction matrix concur in the formation of the overall perceived image quality. The two processes have always been studied separately, thus ignoring the interactions between them.

In this chapter the interactions between the illuminant estimation process and the color correction matrix in the formation of the overall color accuracy are investigated, especially when the white point estimation is imperfect. How the color correction transform amplifies the illuminant estimation errors is also investigated. Furthermore, it is shown that it is possible to incorporate knowledge about the illuminant estimation behavior in the optimization of the color correction matrix to alleviate the error amplification. It is demonstrated that a fixed device chromatic response characterization, which is often adopted, is not able to produce good color accuracy in most situations. New strategies to improve color accuracy under both perfect and imperfect white point estimation are proposed, which clearly suggest that adaptive color transformations have to be preferred in order to improve the color accuracy Bianco *et al.* (2009a).

Usually the color matrix transform is optimized for a single illuminant and is applied as it is for all the illuminants that can occur. This could lead to high colorimetric accuracy if the occurring illuminant is the one for which the matrix has been derived (assuming that it is correctly compensated by the AWB module), and low colorimetric accuracy for different illuminants. The first part of the proposed strategy shows how to compute a combined matrix for different classes of commonly occurring illuminants. If only a-priori probability distribution about the illuminant occurrences is known, the best color matrix can be found offline and applied as it is for all the shots; if the AWB is able to give a probability distribution about the illuminant in the scene (as color-by-correlation Finlayson & Hordley (2001) does), an adaptive optimal matrix transform could be found for each shot.

The second part of the proposed strategy starts from the consideration that since the illuminant estimation is an ill-posed problem Yang & Yuille (1991), the AWB is one of the most delicate modules of the entire color processing pipeline. In order to limit the colorimetric errors in case of an imperfect illuminant estimation, a strategy to derive color correction matrices with white point estimation error buffer is also described.

5.1 Illuminant Varying Color Correction Matrix

In the following a more compact version of Eq.2.5 is used:

$$\mathbf{RGB}_{out} = (\alpha \mathbf{AI} \cdot \mathbf{RGB_{in}})^{\gamma} \tag{5.1}$$

where α, $\mathbf{I}$ and $\mathbf{A}$ respectively represent the exposure compensation gain, the diagonal matrix for the illuminant compensation and the color matrix transformation.

Given a set of n different patches whose sRGB values $\mathbf{r}$ are known, and the corresponding camera raw values $\mathbf{c}$ measured by the sensor when the patches are lit by the chosen illuminant, what is usually done is to find the matrix $\mathbf{M}$ that satisfies:

$$\mathbf{M} = \arg\left(\min_{\mathbf{A}\in\mathbb{R}^{\mathbf{3\times 3}}} \sum_{\mathbf{k=1}}^{\mathbf{n}} \mathcal{E}\left(\mathbf{r_k}, (\alpha \mathbf{AIc_k})^{\gamma}\right)\right) \tag{5.2}$$

where $\mathcal{E}$ is the chosen error metric, and the subscript k indicates the triplet in the k^{th} column of the matrix. In this chapter the error metric $\mathcal{E}$ simply consists of the computation of the average ΔE_{94} colorimetric error between the reference and calculated sRGB values mapped in the CIELab color space, but more complex functionals may be used Bianco *et al.* (2007, 2008e, 2009d). The values of α and I are previously computed in order to perfectly expose the scene and compensate for the illuminant. Given the importance of neutral tones in color reproduction, the 9 degrees of freedom (DoF) of the color matrix transformation are usually reduced to 6 by a white point preserving constraint, i.e. a neutral color in the device dependent color space should be mapped to a neutral color in the device independent color space. This can be easily obtained by constraining each row to sum to one.

In order to be able to optimize the color matrix transformation under multiple illuminants we have to extend Eq.5.2. Let us suppose to consider m different illuminants, and to have an a-priori probability distribution $\mathbf{w} = \{w_1, \ldots, w_m\}$

about them. Eq.5.2 can then be easily extended as:

$$\mathbf{M} = \arg\left(\min_{\mathbf{A}\in\mathbb{R}^{3\times 3}} \sum_{\mathbf{j=1}}^{\mathbf{m}} \mathbf{w_j}\left(\sum_{\mathbf{k=1}}^{\mathbf{n}} \mathcal{E}\left(\mathbf{r_k}, (\alpha_\mathbf{j}\mathbf{A}\mathbf{I_j}\mathbf{c_k})^\gamma\right)\right)\right) \tag{5.3}$$

$$\text{subject to} \quad \sum_{j=1}^{3} A_{(i,j)} = 1, \forall i \in \{1,2,3\}$$

Obviously the probability distribution $\mathbf{w}$ can also represent the relative importance that we want to give to the errors under each considered illuminant. As for the the single illuminant case, the color matrix can be calculated offline and then applied as it is to each different shot.

In the case we use an AWB algorithm that is able to give information about the probability of the illuminant in the scene, we can use this information to give the best color matrix transform for that illuminant probability distribution. In this case an adaptive color matrix is applied for each shot, thus leading to more computational requirements. Different strategies can be adopted, ranging from higher to lower computational and memory requirements:

- for each shot a new optimization could be carried out by using into Eq.5.3 the illuminant probability distribution as it comes out from the AWB process;
- all the possible illuminant probability distributions that the AWB could produce in real situations could be quantized and for each of them a different optimization could be carried out; the transforms obtained could be stored into a LUT. Then for each shot the best color matrix transform could be found by interpolation of the LUT distributions;
- a different optimization could be carried out for each illuminant by using Eq.5.2 and storing the color matrix transforms obtained. Hence the best color matrix transform for each shot could be obtained through a linear combination of the stored transforms by using as weighting vector the illuminant probability distribution provided by the AWB estimation.

5.2 Color Correction Matrix with White Balance Error Buffer

In order to give a faithful representation of the scene in the sRGB color space, all the existing color matrix transformations rely on the assumption that the illuminant has been correctly estimated and compensated by the AWB. As already said in the previous chapter, the illuminant estimation is an ill-posed problem and the AWB estimation module often fails. It has been also demonstrated that the best AWB algorithm of all does not exist, and for each different algorithm there are images on which it produces poor results Bianco *et al.* (2008c). Moreover, when the AWB fails, the errors in the illuminant estimation and compensation could be even amplified by the color matrix transformation. Inspired by this consideration, a strategy to compute color matrix transformations which assumes an illuminant white point estimation error has been developed with the aim of having color matrix transformations more robust to illuminant estimation and compensation errors.

Let us consider the case of a single illuminant optimization. The generalization to the multiple illuminant case is straightforward. Suppose that the best gain coefficients $\mathbf{g_0} = [r_0, g_0, b_0]$ have already been determined and reshaped in the diagonal transform G_0 to compensate the considered illuminant; we then generate a set $\mathbf{g} = \{g_1, \ldots, g_s\}$ of s gain coefficients with different ΔE_{94} from $\mathbf{g_0}$. This can be used to simulate errors that may occur in the AWB. Knowledge about error trends in the AWB estimation process can also be exploited in order to assess if there are more probable hue directions to consider and/or different maximum chroma values for the different hue directions. Furthermore, a weights distribution $\mathbf{u} = \{u_0, \ldots, u_s\}$ can be adopted to reflect the error acceptability for different hue and chroma values. The optimization problem can be thus formulated as:

$$\mathbf{M} = \arg\left(\min_{\mathbf{A}\in\mathbb{R}^{\mathbf{3\times 3}}} \sum_{\mathbf{j=0}}^{\mathbf{s}} \mathbf{u_j}\left(\sum_{\mathbf{k=1}}^{\mathbf{n}} \mathcal{E}\left(\mathbf{r_k}, (\alpha_\mathbf{j}\mathbf{A}\mathbf{G_j}\mathbf{c_k})^\gamma\right)\right)\right) \tag{5.4}$$

$$\text{subject to} \quad \sum_{j=1}^{3} A_{(i,j)} = 1, \forall i \in \{1,2,3\}$$

where $\mathbf{G}_j$, $j = \{0, \ldots, s\}$ are the diagonal matrices obtained respectively by reshaping the gain coefficients $\{g_0, \ldots, g_s\}$. Although very similar to Eq.5.3, the idea behind Eq.5.4 is quite different. Its natural extension to the case of multiple illuminants is straightforward:

$$\mathbf{M} = \arg\left(\min_{\mathbf{A}\in\mathbb{R}^{3\times 3}} \sum_{\mathbf{i=1}}^{\mathbf{m}} \mathbf{w_i} \sum_{\mathbf{j=0}}^{\mathbf{s}} \mathbf{u_j} \left(\sum_{\mathbf{k=1}}^{\mathbf{n}} \mathcal{E}\left(\mathbf{r_k}, (\alpha_\mathbf{j} \mathbf{A} \mathbf{G_{i,j}} \mathbf{c_k})^\gamma\right) \right) \right) \tag{5.5}$$

$$\text{subject to} \quad \sum_{j=1}^{3} A_{(i,j)} = 1, \forall i \in \{1, 2, 3\}$$

5.3 Experimental Setup

All the experiments were performed by using the ISET Digital Camera Simulator Farrell *et al.* (2003) developed at Stanford University. This system makes it possible to simulate the entire image processing pipeline of a digital camera combining both optical modeling and sensor technology simulation. Moreover, the ISET is able to emulate different kinds of noise sources involved in the image acquisition process. As reference camera we used the widely diffused Nikon DSLR D70, which sensor spectral sensitivities are known.

5.4 Experimental Results and Discussion

For the single illuminant case, two different benchmarking algorithms in the state-of-the-art have been used to evaluate the proposed strategy and to show that all of them lead to suboptimal or even poor performance when the color matrix transform derived for a given illuminant is used for different illuminants. These benchmarking algorithms are the White Point Preserving Least Squares (WP-PLS) Hubel *et al.* (1997) and an optimization-based algorithm (WPPPS) proposed by the authors Bianco *et al.* (2007), capable of considering multiple error statistics. The results are compared with the single illuminant strategy (SILL) described in Eq.5.2. For this comparison fourteen different training illuminants have been considered: six CIE daylight illuminants (D48, D55, D65, D75, D100, D200), three CIE standard illuminants (A, B, C), a 2000K Planckian black body

radiator, a uniform white (UW), and three fluorescent illuminants (F2, F7, F11). These are the same used by Finlayson Finlayson (1995) with the addition of the three fluorescent illuminants. The training scenes consisted of the Macbeth ColorChecker (MCC) chart illuminated by each of the training illuminant. The data used for the computation of the color correction matrices were derived directly from the sensor raw data by extracting, for each different patch of the MCC, the mean value of the central area.

In Table 5.1 the minimum, mean, maximum and standard deviation of the ΔE_{94} colorimetric error obtained by the considered approaches on the training scenes are reported. Fourteen different color matrix transformations were derived (one for each illuminant) and the results on the optimized illuminant are reported. In Table 5.2 the mean error obtained under all the illuminants considered is reported. The results are reported only for the most performing SILL strategy.

In Fig. 5.1 a radar plot is reported to graphically compare the performance obtained by the single illuminant SILL approach. Four different transformations are compared. They are the ones derived for the D65, A, 2000K and F11 illuminants. It can be noticed that for each illuminant the lowest ΔE_{94} error is reached by the transformation optimized for that specific illuminant. In particular, the overall best results are obtained near the illuminant D65, which could suggest a color filter design aimed to optimize the camera color response under more frequent illuminants. These results suggest that relying on a single transformation optimized for a single illuminant is not the best strategy for the color correction module.

An alternative approach could be based on the computation of a color matrix transformation optimized simultaneously for multiple illuminants, taking eventually into account an a-priori probability distribution of the training illuminants. For greater generality a uniform a-priori distribution for the illuminant probability in Eq.5.3 is adopted here, i.e. $\mathbf{w} = \{w_{D48}, w_{D55}, \ldots, w_{F7}, w_{F11}\} = \{1/14, \ldots, 1/14\}$. The results obtained are reported in Table 5.3, where the most expensive and the cheapest strategy exposed in Sec.5.1 are compared. The first one (HILL) is the result of the minimization using Eq.5.3 with the uniform a-priori distribution, the second one (HILLA) is a linear approximation of it: it is

Table 5.1: Statistics for the ΔE_{94} colorimetric error obtained by the matrixing optimized for the different illuminants, evaluated on the same illuminant for which the optimization is carried out

Illuminant	Method	min	mean	median	max	std
D48	WPPLS	0.4179	1.5814	1.3563	3.9946	1.0080
	WPPPS	0.0230	0.8585	0.6299	4.3109	0.9612
	SILL	0.0114	0.8213	0.5623	4.4062	0.9577
D55	WPPLS	0.4399	1.8060	1.5026	4.9417	1.1727
	WPPPS	0.0692	0.7420	0.5533	3.6860	0.8024
	SILL	0.0257	0.7175	0.5212	3.6407	0.7831
D65	WPPLS	0.4056	1.2924	0.9478	4.0091	0.9552
	WPPPS	0.0171	0.7847	0.6603	3.2856	0.7468
	SILL	0.0533	0.6454	0.4674	3.0602	0.6596
D75	WPPLS	0.4080	1.4321	1.2494	3.9956	0.8514
	WPPPS	0.0641	0.7743	0.5132	2.8967	0.7017
	SILL	0.0760	0.6333	0.4985	2.8013	0.6131
D100	WPPLS	0.7918	2.2523	1.8076	5.1838	1.2901
	WPPPS	0.0093	0.7710	0.5056	2.9846	0.7647
	SILL	0.0780	0.6871	0.6009	2.8135	0.6404
D200	WPPLS	0.8414	2.3075	2.4154	3.3730	0.6708
	WPPPS	0.0217	0.9529	0.6037	3.8108	0.9297
	SILL	0.0390	0.8745	0.6948	3.5975	0.8381
A	WPPLS	0.7814	2.9431	1.9984	9.7471	2.1096
	WPPPS	0.0129	1.8278	1.1124	9.1925	2.1240
	SILL	0.0491	1.7083	0.8344	8.9678	2.1411
B	WPPLS	0.2074	2.2336	1.8791	5.8973	1.7048
	WPPPS	0.0317	0.8640	0.6627	4.1517	0.9303
	SILL	0.0292	0.8337	0.6168	4.3472	0.9624
C	WPPLS	0.4818	1.5291	1.3728	3.0041	0.6567
	WPPPS	0.0396	0.6704	0.4380	3.0910	0.7027
	SILL	0.0279	0.6289	0.4475	2.9747	0.6605
2000K	WPPLS	1.9435	4.7032	4.2830	8.0833	1.8531
	WPPPS	0.5961	3.1238	1.6654	14.2835	3.2635
	SILL	0.2112	2.9595	1.4862	15.3819	3.6402
UW	WPPLS	0.1984	1.3010	1.2808	2.9917	0.7966
	WPPPS	0.1286	1.7990	1.2915	6.1698	1.5562
	SILL	0.0915	0.7028	0.5855	3.0210	0.6685
F2	WPPLS	0.5355	2.2654	1.7426	4.9969	1.2462
	WPPPS	0.0037	1.4226	0.6329	4.8860	1.4983
	SILL	0.0036	1.3488	0.8667	5.5046	1.4042
F7	WPPLS	0.6094	1.4263	1.4061	2.6802	0.5767
	WPPPS	0.0970	3.3418	3.1368	10.8883	2.9092
	SILL	0.0110	0.5683	0.4116	2.9742	0.6324
F11	WPPLS	0.7488	2.6258	2.3248	8.1779	1.8282
	WPPPS	0.0129	1.4277	0.9201	5.2315	1.4583
	SILL	0.0054	1.3085	0.8711	6.1566	1.4445

Table 5.2: Average ΔE_{94} colorimetric error obtained by the matrixing optimized individually for the different illuminants, evaluated on all the considered illuminants

Opt. Illuminant	D48	D55	D65	D75	D100	D200	A	B	C	2000K	UW	F2	F7	F11
D48	0.8213	0.8317	1.0129	1.2220	1.6491	2.3626	2.8787	0.8723	1.1586	6.0337	0.9039	3.5215	1.0178	2.9638
D55	0.9093	0.7175	0.7952	0.9852	1.4092	2.1496	3.1261	0.9454	0.9482	6.1841	0.7674	3.6511	0.9251	3.1802
D65	1.1327	0.8442	0.6454	0.7357	1.1323	1.8831	3.4034	1.1794	0.7225	6.3473	0.8435	3.9105	1.0456	3.5043
D75	1.3468	1.0297	0.7414	0.6333	0.9256	1.6689	3.6139	1.3973	0.6573	6.4685	1.0264	4.1693	1.2333	3.7911
D100	1.7653	1.4380	1.0913	0.8645	0.6871	1.2904	3.9945	1.8143	0.9547	6.6999	1.4309	4.7326	1.6906	4.3710
D200	2.5161	2.1995	1.8357	1.5540	1.1329	0.8745	4.6530	2.5532	1.6620	7.1161	2.1830	5.6369	2.5495	5.2055
A	2.1816	2.4159	2.6934	2.9193	3.3120	3.9167	1.7083	2.1853	2.8036	5.1161	2.4118	3.4623	2.6127	2.7670
B	0.8511	0.8259	1.0160	1.2312	1.6613	2.3652	2.9022	0.8337	1.1192	6.0660	0.8528	3.5571	1.0447	2.9772
C	1.2740	0.9760	0.7198	0.6609	0.9909	1.7328	3.5564	1.3010	0.6289	6.4578	0.9490	4.1136	1.2004	3.7020
2000K	6.6634	6.8535	7.0755	7.2447	7.5241	7.9369	5.3768	6.6859	7.1736	2.9595	6.9148	7.4405	7.0510	6.8837
UW	0.9462	0.7574	0.7910	0.9793	1.4049	2.1320	3.1356	0.9419	0.9010	6.1806	0.7028	3.6831	0.9644	3.1513
F2	3.1898	3.2267	3.3946	3.5966	4.0300	4.7823	3.8891	3.2199	3.5582	6.6765	3.2962	1.3488	2.6552	2.1881
F7	1.1856	1.0180	1.0394	1.1953	1.6242	2.4070	3.3278	1.2051	1.1908	6.3351	1.0176	3.1048	0.5683	2.9159
F11	2.3482	2.4477	2.6733	2.8996	3.3302	4.0055	3.0374	2.3541	2.7909	5.9770	2.4343	2.1346	2.2188	1.3085

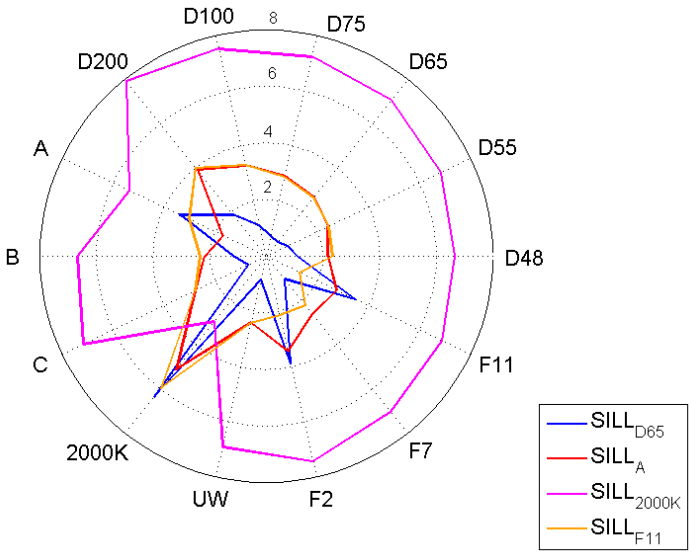


Figure 5.1: Radar plot of the mean ΔE_{94} errors obtained under the different illuminants considered by four different SILL approaches optimized for four different illuminants; D65, A, 2000K and F11

the result of the linear combination of the results obtained on the fourteen different illuminants considered. It is possible to notice that the two strategies convey almost identical results, making it possible to use the cheapest strategy without affecting the color accuracy. A further analysis shows how both the HILL and HILLA strategies, at least on the simulation carried out in these experiments, lead to almost identical results to those obtained by the best SILL approaches (i.e. SILL_{D55} and SILL_{UW}).

Table 5.3: Average ΔE_{94} colorimetric error obtained by the matrixing optimized simultaneously for the different illuminants, evaluated on all the considered illuminants

Illuminant	SILL_{D55}	SILL_{UW}	HILL	HILLA
D48	0.9093	0.9462	0.9325	0.9326
D55	0.7175	0.7574	0.7495	0.7495
D65	0.7952	0.7910	0.7972	0.7971
D75	0.9852	0.9793	0.9886	0.9870
D100	1.4092	1.4049	1.4220	1.4237
D200	2.1496	2.1320	2.1704	2.1699
A	3.1261	3.1356	3.1444	3.1445
B	0.9454	0.9419	0.9507	0.9510
C	0.9482	0.9010	0.9433	0.9442
2000K	6.1841	6.1806	6.1903	6.1908
UW	0.7674	0.7028	0.7237	0.7239
F2	3.6511	3.6831	3.5429	3.5415
F7	0.9251	0.9644	0.8432	0.8442
F11	3.1802	3.1513	3.0754	3.0770
avg	1.9067	1.9051	1.8910	1.8912

The analysis of Fig. 5.1 and Table 5.3 shows that there is not enough room for improvement for the HILL and HILLA strategies. In fact, it is possible to notice that, for example, the SILL_{D65} color transform conveys small colorimetric errors for illuminant with a Correlated Color Temperature (CCT) close to the

one for which the transformation has been optimized (i.e. 6500K) and high errors for very distant CCTs. On the other hand, the transformation optimized for very low CCT (for example the SILL_{2000K} with a CCT of 2000K) conveys high colorimetric errors for a large number of illuminants. This means that if we want to lower the colorimetric errors for very low CCTs we have to decrease the color accuracy for less extreme CCTs. This, at least on the simulation carried out in these experiments with the illuminant probability adopted, does not lead to a significant improvement.

The behavior of the different color transformation matrices under the different illuminants suggests that a greater improvement in color accuracy for all the illuminants could be reached if we are able to identify the actual illuminant and choose the best color correction matrix for it. In order to test this hypothesis 1,000 different test scenes have been generated. Each of them was composed of a random power of two different patches ($2^k,\ k \in \mathbb{N},\ k \leq 11$) extracted from the ISO reflectance database ISO (2003) and illuminated by a random illuminant extracted from the illuminant test dataset Barnard *et al.* (2002). Three different approaches are tested: the first one is the multiple illuminant ideal case (MILL), i.e. for each of the test illuminants the best color correction matrix is computed and applied. In the second one (MILLA), the illuminant CCT is first computed and the color correction matrix optimized for the training illuminant with the closest CCT is applied. In the last case (MILLA_2), the illuminant CCT is first computed, the two training illuminants ILL_i and ILL_j with the closest CCTs are identified and the color correction matrix is calculated as follows:

$$M = \alpha SILL_i + (1 - \alpha) SILL_j \tag{5.6}$$

where

$$\alpha = \frac{d(CCT, CCT_j)}{d(CCT, CCT_i) + d(CCT, CCT_j)} \tag{5.7}$$

The mean colorimetric error obtained on the test images by all the proposed strategies are reported in Table 5.4. The percentage accuracy improvement with respect to the SILL_{D65} is also reported. This is chosen as benchmarking strategy, since a single color correction matrix optimized for the D65 illuminant is what is usually used in the state of the art.

Table 5.4: Average ΔE_{94} colorimetric error and percentage colorimetric accuracy improvement respect to the most performing strategy, obtained by the all the proposed strategies

Method	Opt.Illuminant	mean ΔE_{94}	perc. improv. (%)
SILL	D48	3.0386	7.31
SILL	D55	3.1250	4.67
SILL	D65	3.2782	*
SILL	D75	3.4274	-4.55
SILL	D100	3.7436	-14.20
SILL	D200	4.3225	-31.86
SILL	A	3.3380	-1.82
SILL	B	3.0628	6.57
SILL	C	3.3945	-3.55
SILL	2000K	7.9029	-141.07
SILL	UW	3.1422	4.15
SILL	F2	3.1371	4.30
SILL	F7	3.0307	7.55
SILL	F11	2.9857	8.92
HILL	$\mathbf{w} = \{1/14, \ldots, 1/14\}$	2.9524	9.94
HILLA	$\mathbf{w} = \{1/14, \ldots, 1/14\}$	2.9578	9.77
MILL	*	2.2002	32.88
MILLA	*	2.7711	15.47
MILLA_2	*	2.6318	19.72

All the experiments made rely on the assumption that the scene illuminant has been correctly estimated and compensated for. This hypothesis does not often hold. It is known, in fact, that the different white balance algorithms make errors in the illuminant estimation. Let us examine how the color correction matrix propagates the illuminant estimation error. Let us consider, for example, what happens under the D65 illuminant. To this end, starting from the optimal D65

compensation gains, a set of gains with varying illuminant estimation accuracy levels is generated. The error measure chosen to generate them is the Perceptual Euclidean Distance (PED) recently proposed by Gijsenij et al. Gijsenij *et al.* (2008), but a different choice could be made. The gains are generated at 10 different PED magnitudes in 64 different directions in the YCbCr color space, i.e. fixed a direction in the YCbCr color space, the gain along that direction with the desired PED error is found. In Fig. 5.2(a) a cylindrical plot is reported where to each combination (ρ, θ) representing the magnitude and direction of the PED error, an altitude information is associated representing the average ΔE_{94} error produced on the MCC acquired under the D65 illuminant, corrected with the distorted illuminant gains. In Fig. 5.2(b) the same plot is represented after the color correction with the SILL_{D65} matrix. It is possible to notice that the SILL color correction matrix greatly reduces the ΔE_{94} in the case that the illuminant has been estimated and corrected with a high accuracy (central part of the plots). The errors become larger as the illuminant estimation precision lowers. The average slopes of the surfaces plotted, 0.6589 for the left one and 1.7306 for the right one, reflect the higher dependency on the illuminant estimation precision due to the color correction matrix SILL_{D65}.

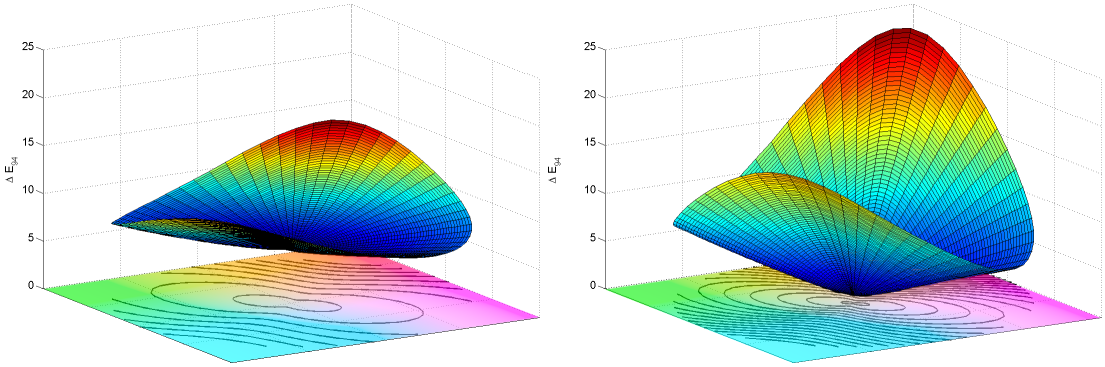


Figure 5.2: ΔE_{94} error distribution as the error in the illuminant estimation and compensation changes under the D65 illuminant: no color correction (left), SILL_{D65} color correction

The next experiment aims to find color correction matrices less dependent on the precision with which the illuminant scene is estimated. To this end, for each

of the training illuminant, 1,000 different scenes composed of a random power of two different patches ($2^k,\ k \in \mathbb{N},\ k \leq 11$) extracted from the ISO reflectance database are created. In order to estimate such color correction matrices, we have to know or at least suppose, the error trends of the AWB errors (i.e. the weight distribution $\mathbf{u}$ in Eq.5.4). Instead of using a general uniform probability the use of the error probability distribution of a real illuminant estimation algorithm is preferred here: the Gray World algorithm has been selected, but a different choice could be made.

Each of the new color correction matrices found, optimized for a single illuminant with white balance error buffer (SILL-WEB), is compared with the previous one (SILL) optimized for the same illuminant on 1,000 randomly generated test scenes. Each of them was composed of a random power of two different patches ($2^k,\ k \in \mathbb{N},\ k \leq 11$) extracted from the ISO reflectance database and illuminated by the same training illuminant for which the matrices have been optimized. The results of the comparisons are reported in Table 5.5. It is possible to notice that both the average ΔE_{94} errors and the average slope of the SILL-WEB matrices are less than the one obtained by the SILL ones. In particular, the lower slope values, reflect the minor dependence on the illuminant estimation precision. The lower ΔE_{94} errors reflect instead the fact that the new color correction matrices have learned and are able to compensate to some extent the way the illuminant estimation algorithms fails.

In Fig. 5.3(a) a multispectral acquisition of the Macbeth Color Checker DC (MDC) rendered in sRGB under the D65 illuminant is reported. The correctly white balanced version is reported in the center of the figure. Using the PED error measure 8-directions 1-level distortions in the illuminant estimation are considered. The PED value considered is the average error of the Gray World illuminant estimation algorithm on the training set. Images obtained using the imperfect illuminant estimation are reported around the correctly white balanced image in a concentric fashion. The sCIELAB Zhang & Wandell (1997b) error maps between the correctly white balanced image and the other ones are reported in Fig. 5.3(b).

In Fig. 5.4(a) the same MDC of Fig. 5.3(a) after color correction using the SILL approach are reported. The corresponding sCIELAB error maps between

Table 5.5: Average ΔE_{94} colorimetric error and average slope of the SILL and SILL-WEB color correction matrices

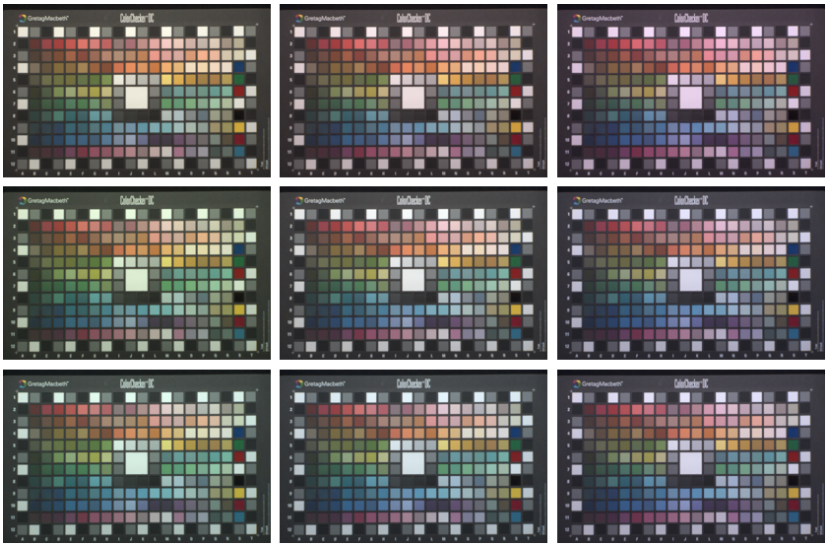
	SILL		SILL-WEB	
Opt.Illuminant	mean ΔE_{94}	mean slope	mean ΔE_{94}	mean slope
D48	3.5666	1.5056	3.3481	1.3469
D55	3.5170	1.6056	3.3043	1.4844
D65	3.4892	1.7306	3.2642	1.5893
D75	3.4653	1.8283	3.2220	1.7118
D100	3.4615	1.9940	3.2277	1.8201
D200	3.5125	2.2206	3.2706	2.0165
A	4.3060	1.4236	4.0250	1.3379
B	3.6232	1.4563	3.4134	1.2810
C	3.4786	1.7486	3.2524	1.6258
2000K	5.4033	2.3989	5.2324	2.1032
UW	3.5039	1.5232	3.2892	1.3853
F2	3.8222	1.3453	3.6657	1.2536
F7	3.4308	1.6958	3.2157	1.6033
F11	3.9514	1.1698	3.7766	1.0743

the correctly white balanced version and the others are reported in Fig. 5.4(b).

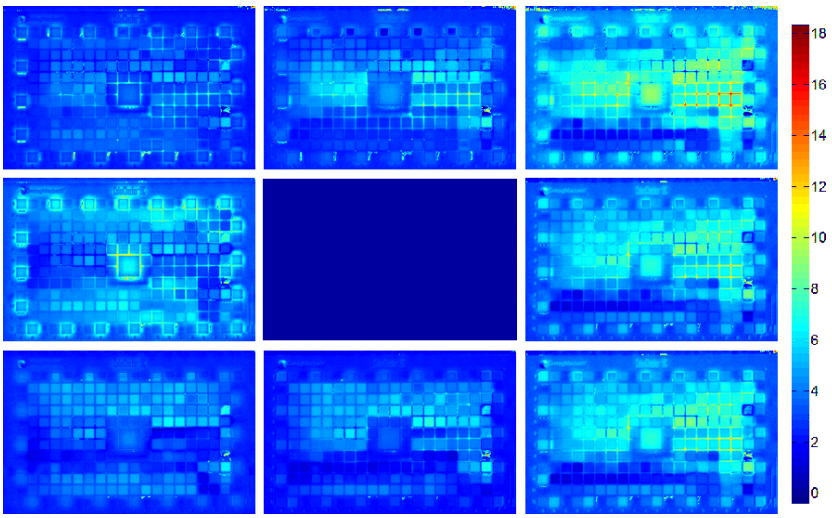
In Fig. 5.5(a) the same MDC of Fig. 5.3(a) after color correction using the SILL-WEB approach are reported. The corresponding sCIELAB error maps between the correctly white balanced version and the others are reported in Fig. 5.5(b).

For a more in-depth comparison of the amplification of the error in the white point estimation by the color correction matrix, the mean and median sCIELAB values of the error maps in Fig. 5.3(b), 5.4(b) and 5.5(b) are reported in Table 5.6. It is possible to notice that, as expected, for all the PED error directions, both the SILL and SILL-WEB color correction matrices amplify the error in the white point estimation and correction.

In Table 5.4 was found that an improvement in color accuracy for all the illu-



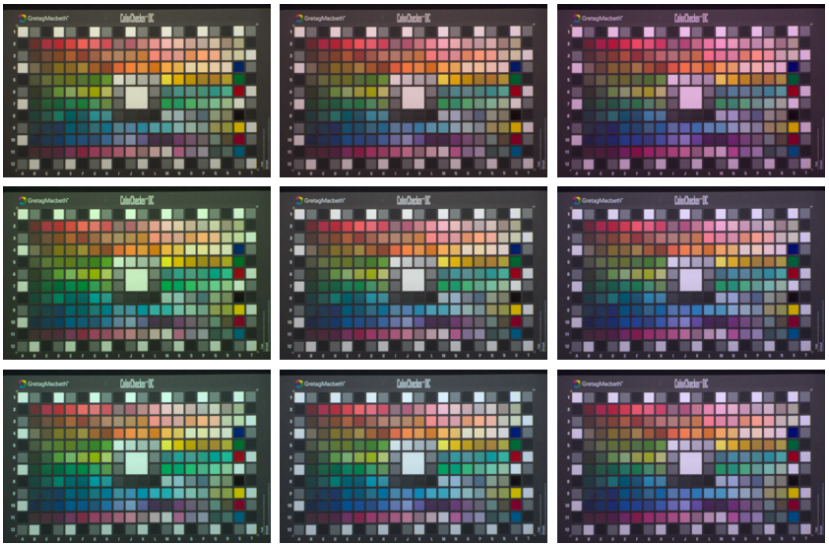
(a)



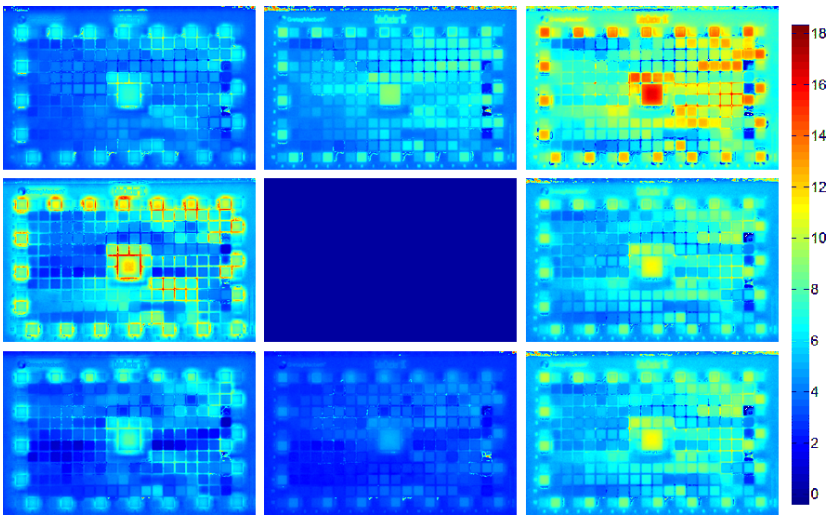
(b)

Figure 5.3: The MDC under D65 with different white balance errors (a) without color correction matrix; the sCIELAB error maps between the correctly white balanced image and the others (b)

minants could be reached if one is able to identify the actual illuminant and choose the best color correction matrix for it (i.e. the strategy named MILL, MILLA and MILLA$_2$). Let us compare here the MILL, MILLA and MILLA$_2$ strategies



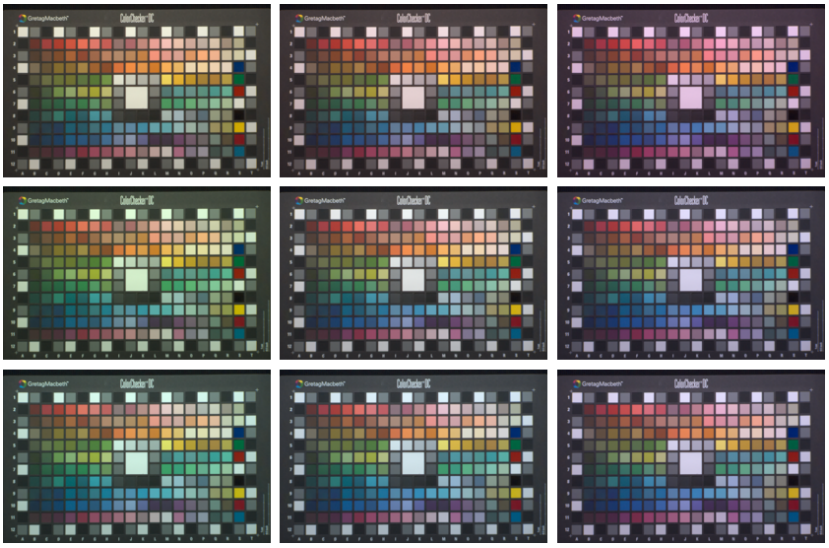
(a)



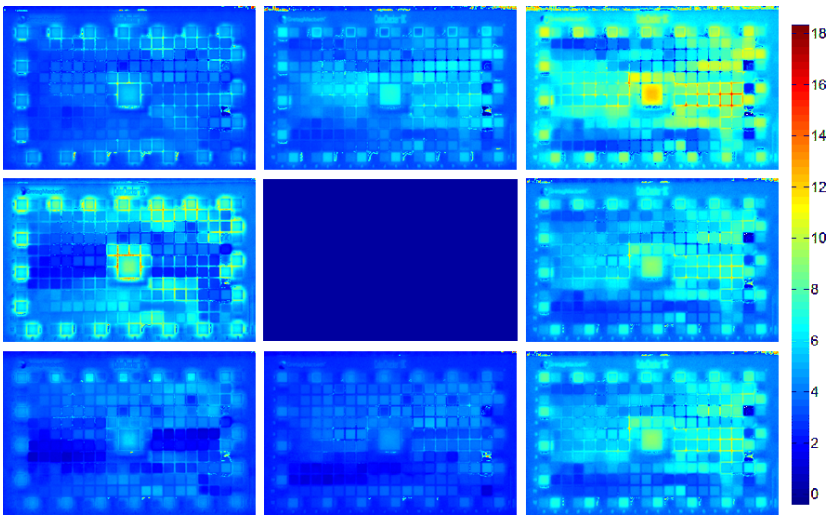
(b)

Figure 5.4: The MDC under D65 with different white balance errors (a) with the SILL color correction matrix; the sCIELAB error maps between the correctly white balanced image and the others (b)

based on the SILL color correction matrices, against the MILL-WEB, MILLA-WEB and MILLA$_2$-WEB counterparts based on the SILL-WEB color correction matrices. To this end 1,000 different test scenes were generated. Each of them



(a)



(b)

Figure 5.5: The MDC under D65 with different white balance errors (a) with the SILL-WEB color correction matrix; the sCIELAB error maps between the correctly white balanced image and the others (b)

was composed of a random power of two different patches ($2^k,\ k \in \mathbb{N},\ k \leq 11$) extracted from the ISO reflectance database and illuminated by a random illuminant extracted from the illuminant test dataset. Three new different approaches

Table 5.6: Mean and median sCIELAB ΔE_{94} error between the correctly white balanced image and the same image with two different PED error magnitudes in eight different directions for the different color correction matrices considered: no color correction, SILL and SILL-WEB

PED error	Color Correction					
	No color correction		SILL		SILL-WEB	
direction	mean	median	mean	median	mean	median
0π	4.8701	4.5964	6.3522	6.0137	5.4265	5.2044
$1/4\pi$	5.7334	5.4786	8.5881	8.2866	6.9051	6.7197
$1/2\pi$	3.8458	3.4943	5.5519	5.2311	4.6457	4.3838
$3/4\pi$	3.4821	3.3528	4.6693	4.3774	4.0539	3.8657
π	4.6925	4.5101	6.5226	5.9941	5.2987	4.9317
$5/4\pi$	3.5040	3.2696	4.4709	4.2556	3.7049	3.6304
$3/2\pi$	3.3620	2.9542	3.4407	3.2909	3.3645	3.2227
$7/4\pi$	4.8701	4.5964	6.3522	6.0137	5.4265	5.2044

are tested: the first one is the ideal case (MILL-WEB), i.e. for each scene of the test set, the illuminant is estimated and compensated for using the Gray World (GW) algorithm and then the best color correction matrix is computed and applied. In the second one (MILLA-WEB), the illuminant is first estimated and compensated for using the GW algorithm and then the color correction matrix optimized for the training illuminant with the closest gains (estimated with GW) is applied. In the last case (MILLA$_2$-WEB), the illuminant compensation gains are first computed with the GW, then the two training illuminants ILL$_i$ and ILL$_j$ with the closest gains are identified and the color correction matrix is calculated as follows:

$$M = \alpha SILL\text{–}WEB_i + (1-\alpha) SILL\text{–}WEB_j \tag{5.8}$$

where

$$\alpha = \frac{PED(gains, gains_j)}{PED(gains, gains_i) + PED(gains, gains_j)} \tag{5.9}$$

The mean colorimetric error obtained on the test images by all the proposed

strategies are reported in Table 5.7. The percentage colorimetric accuracy improvement respect to the SILL_{D65} strategy is reported. This has been chosen as benchmarking strategy since a single color correction matrix, optimized for the D65 illuminant, is what is generally used in the state of the art.

Table 5.7: Average ΔE_{94} colorimetric error and percentage colorimetric accuracy improvement respect to the state of the art strategy (SILL_{D65}), obtained by the all the proposed strategies

Method	mean ΔE_{94}	perc. improv. (%)
SILL_{D65}	5.2252	*
MILL	4.5907	12.14
MILLA	4.9384	5.49
MILLA_2	4.7171	9.72
SILL-WEB_{D65}	4.7260	9.55
MILL-WEB	3.8805	25.73
MILLA-WEB	4.4420	14.99
MILLA_2-WEB	4.2214	19.21

5.5 Conclusions

In this chapter the interactions between the illuminant estimation process and the color correction matrix in the formation of the overall color accuracy have been investigated, especially when the white point estimation is imperfect. The amplification of the illuminant estimation errors by the color correction transform has also been investigated. Furthermore, it has been shown that it is possible to incorporate knowledge about the illuminant estimation behavior in the optimization of the color correction matrix in order to alleviate the error amplification. It has been demonstrated that a fixed device chromatic response characterization, which is often adopted, is not able to produce a good color accuracy in most

situations. New strategies to improve color accuracy under both perfect and imperfect white point estimation have been proposed, which clearly suggests that adaptive color transformations have to be preferred in order to improve the color accuracy. The experimental results showed that respect to a fixed device chromatic response characterization optimized for the single D65 illuminant, the best adaptive color transform proposed decreased the colorimetric error on average by 19.72% for the perfect illuminant estimation case (i.e. ideal illuminant estimation) and by 19.21% for the imperfect illuminant estimation case (i.e. using a real illuminant estimation algorithm).

Chapter 6

Color correction pipelines evaluation

The aim of this chapter is to investigate how the proposed methods can be combined in order to design a new color correction pipeline. In particular, we investigate the color accuracy improvement that the illuminant estimation algorithms of chapter 4 and the color space conversion strategies of chapter 5 can give when used both individually and when are properly combined.

To test the performance of the investigated processing pipelines, a standard dataset of RAW camera images having a known color target is used Gehler *et al.* (2008). This dataset is captured using a high-quality digital SLR camera in RAW format, that is therefore free of any color correction. The procedure adopted is given in figure 6.2. The Macbeth ColorChecker (MCC) chart is included in every scene acquired, and this allows to estimate accurately the actual illuminant within each acquired image. The dataset consists of a total of 568 images, both indoor (246) and outdoor (322). For each image, the MCC chart corners coordinates are manually extracted and mapped through a perspective transformation into a fixed size rectangle. The mean value of the centers of each color patch are then extracted and are used to evaluate the color accuracy of the whole pipeline. The values of the achromatic patches excluding the white and the black ones (which may be clipped) are used to estimate the illuminant ground truth. Some typical examples of the images belonging to the dataset used are shown in figure 6.1.

The theoretical RGB values of the MCC chart under the CIE D65 illuminant are tabulated in Pascale (2006).

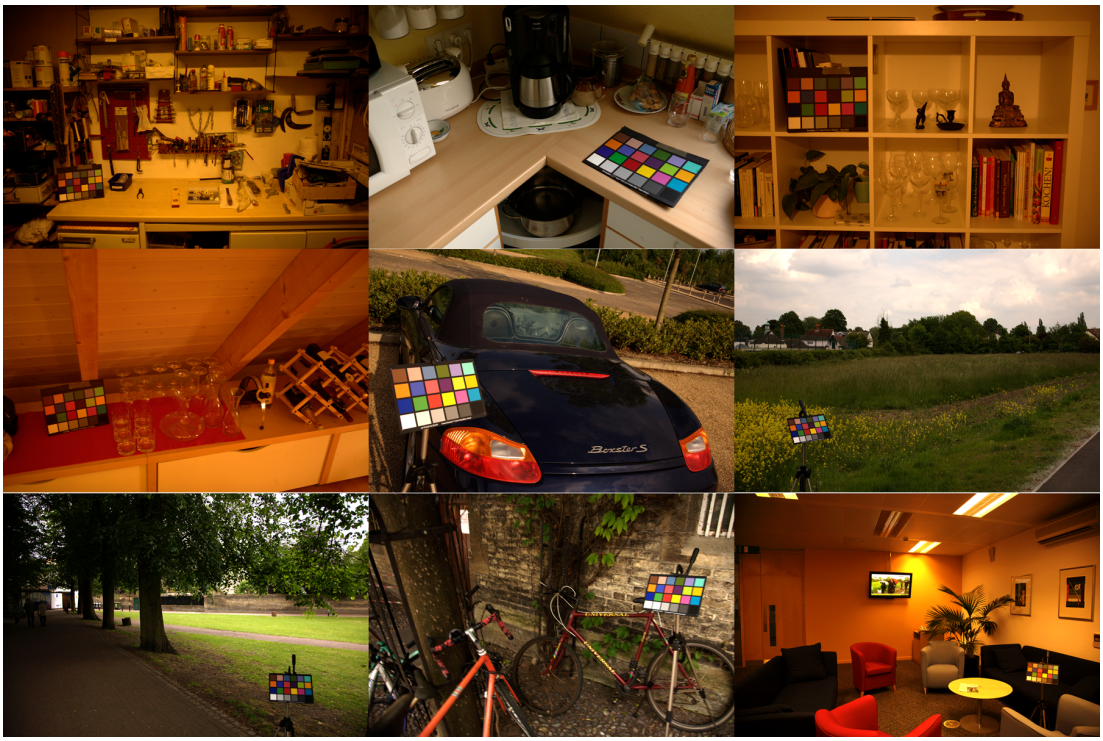


Figure 6.1: Typical examples of the images belonging to the RAW dataset used

The flowchart of the evaluation procedure adopted is given in figure 6.3, where it can be seen that the only step in which the MCC chart is cropped is the illuminant estimation one.

The investigated illuminant estimation algorithms described in chapter 4 are individually run on the images of the dataset excluding the MCC chart regions that have been previously cropped. Given these estimations, the illuminant corrections are then performed on the whole images (therefore including also the MCC chart).

The color matrix transformations found according to the computational strategies described in chapter 5 are then applied to the whole, white balanced, images. For each processed image, the MCC chart is then extracted and the average RGB values of the central area of each patch are calculated. The color rendition accuracy of the pipeline is measured in terms of average ΔE_{94} error between the

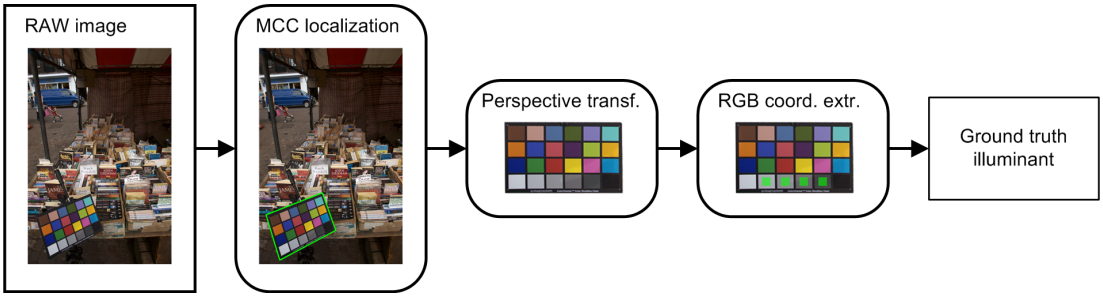


Figure 6.2: Ground truth illuminant measurement: localization of the MCC in the image, projective transformation into a fixed size rectangle, RGB coordinates extraction for each MCC patch, ground truth illuminant estimation

CIELAB color coordinates of the color corrected MCC patches and their theoretical CIELAB values that are computed using standard equations from their theoretical RGB values.

6.1 Pipeline training and testing

In this section the color correction pipelines composed of the combination of the modules for the illuminant estimation algorithms of chapter 4 and the color space conversion strategies of chapter 5 are tested. In total twenty different pipelines are tested; they are generated as an exhaustive combination of the modules proposed. The pipelines tested are clustered into four different groups depending on the color correction strategy adopted:

- single illuminant color matrixing (SILL):
  - best single AWB and single illuminant color matrixing (BS-SILL);
  - class-based AWB with ideal classifier and single illuminant color matrixing (CB-SILL$_{ideal}$);
  - class-based AWB with real classifier and single illuminant color matrixing (CB-SILL);
  - feature-based AWB with ideal classifier and single illuminant color matrixing (CB-SILL$_{ideal}$);

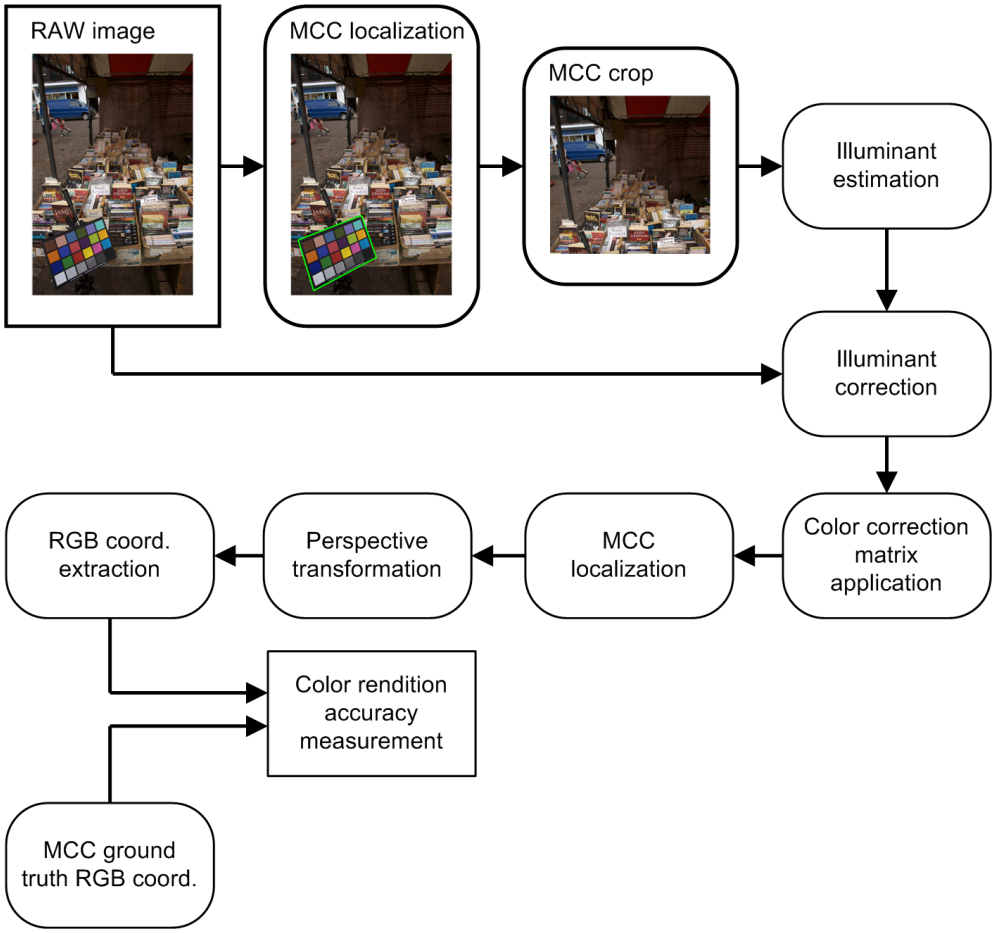


Figure 6.3: Pipeline evaluation: the MCC is localized and cropped; the illuminant is estimated on the cropped image and the un-cropped image is illuminant corrected on the basis of this estimate; to this image the color correction matrix is then applied, the MCC is localized and perspectively transformed and the RGB coordinates are extracted and compared with the MCC theoretical ones

  - feature-based AWB with real classifier and single illuminant color matrixing (CB-SILL);

- multiple illuminant color matrixing (MILLA):
  - best single AWB and multiple illuminant color matrixing (BS-MILLA);
  - class-based AWB with ideal classifier and multiple illuminant color

matrixing (CB-MILLA$_{ideal}$);

- class-based AWB with real classifier and multiple illuminant color matrixing (CB-MILLA);
- feature-based AWB with ideal classifier and multiple illuminant color matrixing (CB-MILLA$_{ideal}$);
- feature-based AWB with real classifier and multiple illuminant color matrixing (CB-MILLA);

- single illuminant color matrixing with white balance error buffer (SILL-WEB):
  - best single AWB and single illuminant color matrixing with white balance error buffer (BS-SILL-WEB);
  - class-based AWB with ideal classifier and single illuminant color matrixing with white balance error buffer (CB-SILL-WEB$_{ideal}$);
  - class-based AWB with real classifier and single illuminant color matrixing with white balance error buffer (CB-SILL-WEB);
  - feature-based AWB with ideal classifier and single illuminant color matrixing with white balance error buffer (CB-SILL-WEB$_{ideal}$);
  - feature-based AWB with real classifier and single illuminant color matrixing with white balance error buffer (CB-SILL-WEB);
- multiple illuminant color matrixing with white balance error buffer (MILLA-WEB):
  - best single AWB and multiple illuminant color matrixing with white balance error buffer (BS-MILLA-WEB);
  - class-based AWB with ideal classifier and multiple illuminant color matrixing with white balance error buffer (CB-MILLA-WEB$_{ideal}$);
  - class-based AWB with real classifier and multiple illuminant color matrixing with white balance error buffer (CB-MILLA-WEB);

- feature-based AWB with ideal classifier and multiple illuminant color matrixing with white balance error buffer (CB-MILLA-WEB$_{ideal}$);
- feature-based AWB with real classifier and multiple illuminant color matrixing with white balance error buffer (CB-MILLA-WEB);

Since the considered modules need a training phase, 30% of the images in the dataset were randomly selected and used as training set; the remaining 70% were used as test set.

6.1.1 Best single AWB and single illuminant color matrixing (BS-SILL)

The best single (BS) AWB algorithm among the general purpose ones considered in section 4.6.4 is identified on the training set and used to estimate the illuminant on the test set. Among the training images the one with the illuminant closest to the one for which the sensor's filters of camera are optimized is chosen and used to optimize the color transformation matrix used (SILL methodology of chapter 5).

The average ΔE_{94} colorimetric error obtained by this pipeline (named BS-SILL) on the MCCs on the test images is computed and reported in Table 6.1. This value will be used as benchmark for the investigated pipelines, since this pipeline is what is usually adopted in digital cameras.

6.1.2 Class-based AWB and single illuminant color matrixing (CB-SILL)

The best single AWB algorithms for the indoor and for the outdoor scenes are identified on the training set, since the images in the whole dataset were labeled as indoor or outdoor scenes. The same color correction matrix of the benchmark pipeline in section 6.1.1 is also used here. For each image in the test set, the best indoor AWB algorithm is used if the image was labeled as indoor, the best outdoor AWB otherwise. The AWB module used is thus the class-based (CB) described in section 4.6. This pipeline, named CB-SILL$_{ideal}$, gives the ideal performance that the pipeline would reach if a perfect classifier were used. Another instantiation

of this pipeline, named CB-SILL, uses a real indoor/outdoor classifier instead. Since the images in the training set were not enough to train a new classifier, the same classifier used in section 4.6 is employed here. It is the case to underline that this classifier is not optimal for the images in this dataset because it was trained on pre-processed images, while the images used here are RAW files.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1

6.1.3 Feature-based AWB and single illuminant color matrixing (FB-SILL)

The first instantiation of this pipeline, named FB-SILL$_{ideal}$, uses an ideal classifier to identify which AWB algorithm is the best for each of the images in the test set. The AWB module used is thus the one described in section 4.7. The color correction matrix used is the same as in section 6.1.1. The second instantiation, named FB-SILL, employs the real classifier used in section 4.7. The same observation about the non-optimality of such a classifier pointed out in section 6.1.2 also holds here.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.1.4 Best single AWB and multiple illuminant color matrixing (BS-MILLA)

The best single (BS) AWB algorithm among the general purpose ones considered in section 4.6.4 is identified on the training set and used to estimate the illuminant on the test set. The ground truth illuminants of the images in the training set have been clustered into seven different clusters using a k-means algorithm Hartigan & Wong (1979). For each centroid of the clusters the best color correction matrix is calculated. For each image in the test set, according to the illuminant correction gains estimated by the AWB algorithms an adaptive linear combination of the optimized color correction matrices is computed and applied (MILLA$_2$ methodology of chapter 5).

The average ΔE_{94} colorimetric error obtained by this pipeline (named BS-MILLA) on the MCCs on the test images is computed and reported in Table 6.1.

6.1.5 Class-based AWB and multiple illuminant color matrixing (CB-MILLA)

The AWB module of this pipeline is the same as in section 6.1.2. What changes is the color correction matrix, which is adaptively calculated like in section 6.1.4. There are two different instantiations for this pipeline, the first with an ideal classifier (CB-MILLA$_{ideal}$) and the second with the real classifier (CB-MILLA) of section 6.1.2.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.1.6 Feature-based AWB and multiple illuminant color matrixing (FB-MILLA)

The AWB module of this pipeline is the same as in section 6.1.3. What changes is the color correction matrix, which is adaptively calculated like in section 6.1.4. There are two different instantiations for this pipeline, the first with an ideal classifier (FB-MILLA$_{ideal}$) and the second with the real classifier (FB-MILLA) of section 6.1.3.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.1.7 Best single AWB and single illuminant color matrixing with white balance error buffer (BS-SILL-WEB)

The best single (BS) AWB algorithm among the general purpose ones considered in section 4.6.4 is identified on the training set and used to estimate the illuminant on the test set. Among the training images the one with the illuminant closest to the one for which the sensor's filters of camera are optimized is chosen and used to optimize the color transformation matrix also taking into account the statistics of

how the selected AWB tends to make errors in the illuminant estimation (SILLWEB methodology of chapter 5).

The average ΔE_{94} colorimetric error obtained by this pipeline (named BS-SILL) on the MCCs on the test images is computed and reported in Table 6.1.

6.1.8 Class-based AWB and single illuminant color matrixing with white balance error buffer (CB-SILLWEB)

The AWB module of this pipeline is the same as in section 6.1.2. The color correction matrix is calculated like the one in section 6.1.7 optimizing the color rendition accuracy also taking into account the statistics of how the class-based AWB strategy tends to make errors in the illuminant estimation. There are two different instantiations for this pipeline, the first with an ideal classifier (CB-SILLWEB$_{ideal}$) and the second with the real classifier (CB-SILLWEB) of section 6.1.2.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.1.9 Feature-based AWB and single illuminant color matrixing with white balance error buffer (FB-SILLWEB)

The AWB module of this pipeline is the same as in section 6.1.3. The color correction matrix is calculated like the one in section 6.1.7 optimizing the color rendition accuracy also taking into account the statistics of how the feature-based AWB strategy tends to make errors in the illuminant estimation. There are two different instantiations for this pipeline, the first with an ideal classifier (FB-SILLWEB$_{ideal}$) and the second with the real classifier (FB-SILLWEB) of section 6.1.3.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.1.10 Best single AWB and multiple illuminant color matrixing with white balance error buffer (BS-MILLA-WEB)

The best single (BS) AWB algorithm among the general purpose ones considered in section 4.6.4 is identified on the training set and used to estimate the illuminant on the test set. The ground truth illuminants of the images in the training set have been clustered into seven different clusters using a k-means. For each centroid of the clusters the best color correction matrix is calculated taking into account also the statistics o how the AWB algorithm tends to make errors in the illuminant estimation of the images belonging to the same cluster. For each image in the test set, according to the illuminant correction gains estimated by the AWB algorithms an adaptive linear combination of the optimized color correction matrices is computed and applied (MILLA$_2$-WEB methodology of chapter 5).

The average ΔE_{94} colorimetric error obtained by this pipeline (named BS-MILLAWEB) on the MCCs on the test images is computed and reported in Table 6.1.

6.1.11 Class-based AWB and multiple illuminant color matrixing with white balance error buffer (CB-MILLA-WEB)

The AWB module of this pipeline is the same as in section 6.1.2. What changes is the color correction matrix, which is adaptively calculated like in section 6.1.10. There are two different instantiations for this pipeline, the first with an ideal classifier (CB-MILLAWEB$_{ideal}$) and the second with the real classifier (CB-MILLAWEB) of section 6.1.2.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.1.12 Feature-based AWB and multiple illuminant color matrixing with white balance error buffer (FB-MILLA-WEB)

The AWB module of this pipeline is the same as in section 6.1.3. What changes is the color correction matrix, which is adaptively calculated like in section 6.1.10. There are two different instantiations for this pipeline, the first with an ideal classifier (FB-MILLAWEB$_{ideal}$) and the second with the real classifier (FB-MILLAWEB) of section 6.1.3.

The average ΔE_{94} colorimetric error obtained by these pipelines on the MCCs on the test images are computed and reported in Table 6.1.

6.2 Experimental result discussion and conclusions

It is possible to notice in table 6.1 that in all the groups of pipelines proposed (groups that share the same color correction matrix strategy, i.e. SILL, MILLA, SILLWEB and MILLAWEB) the use of the feature-based AWB leads to a higher color rendition accuracy with respect to the use of the class-based AWB and of the best single AWB. These results confirm what was found in chapter 4. It is interesting to notice that significant improvements in the color rendition accuracy can be reached even if the classifiers used for the feature and class-based AWB strategies are not optimal.

Analyzing the behavior of the pipelines sharing the same AWB strategy, it is possible to notice that the results of chapter 5 are also confirmed. In fact the multiple illuminant color correction (MILLA) performs better than the single illuminant one (SILL). The single illuminant color correction which is optimized taking into account the statistics of how the AWB algorithm tends to make errors (SILLWEB) performs better than the multiple illuminant color correction (MILLA). Finally, the multiple illuminant color correction with white balance error buffer (MILLAWEB) performs better than the corresponding single illuminant instantiation (SILLWEB).

Table 6.1: Color correction pipelines accuracy comparison

Pipeline	average ΔE_{94}	improvement
BS-SILL	7.5309	-.- -%
CB-SILL$_{ideal}$	7.3875	1.90%
CB-SILL	7.5541	-0.31%
FB-SILL$_{ideal}$	6.5324	13.26%
FB-SILL	7.3684	2.16%
BS-MILLA	6.9636	7.53%
CB-MILLA$_{ideal}$	6.5930	12.45%
CB-MILLA	6.8954	8.43%
FB-MILLA$_{ideal}$	5.9695	20.73%
FB-MILLA	6.7199	10.77%
BS-SILLWEB	6.8079	9.60%
CB-SILLAWEB$_{ideal}$	6.3627	15.51%
CB-SILLAWEB	6.6753	11.36%
FB-SILLAWEB$_{ideal}$	5.8362	22.50%
FB-SILLAWEB	6.4811	13.94%
BS-MILLAWEB	6.4654	14.15%
CB-MILLAWEB$_{ideal}$	6.2949	16.41%
CB-MILLAWEB	6.4058	14.94%
FB-MILLAWEB$_{ideal}$	5.3232	29.32%
FB-MILLAWEB	6.1368	18.51%

For what concerns the best pipeline proposed, which is the FB-MILLAWEB, it can be observed that when using the ideal classifier, 48% of improvement (from 0% to 14.15% with respect to the benchmarking pipeline) is due to the use of the MILLA-WEB color correction matrix approach, and the remaining 52% (from 14.15% to 29.32% with respect to the benchmarking pipeline) to the feature-based AWB approach. When using the real classifier the greatest part of the improvement is due to the use of the of the MILLA-WEB color correction matrix approach, and the remaining part to the feature-based AWB approach. This

can be considered a lower bound of the pipeline performance, since as already explained before the classifier used is not optimal for the images used.

In Figure 6.4 the workflow of the best performing proposed pipeline, i.e. the FB-MILLAWEB, is reported. The low level features considered are extracted from the RAW image and fed to the classifier which as output gives the weights to use for the linear combination of the five simple AWB algorithms considered. The AWB correction gains given by the five simple AWB algorithms considered are then combined to give the AWB correction gains to use. The image is then corrected with these gains to obtain an AWB corrected image. The correction gains are used to identify the two training illuminants most similar to the estimated one. The two color correction matrices computed for the two identified illuminants are retrieved and combined accordingly to the illuminant similarity. The image is finally color corrected using this color correction matrix.

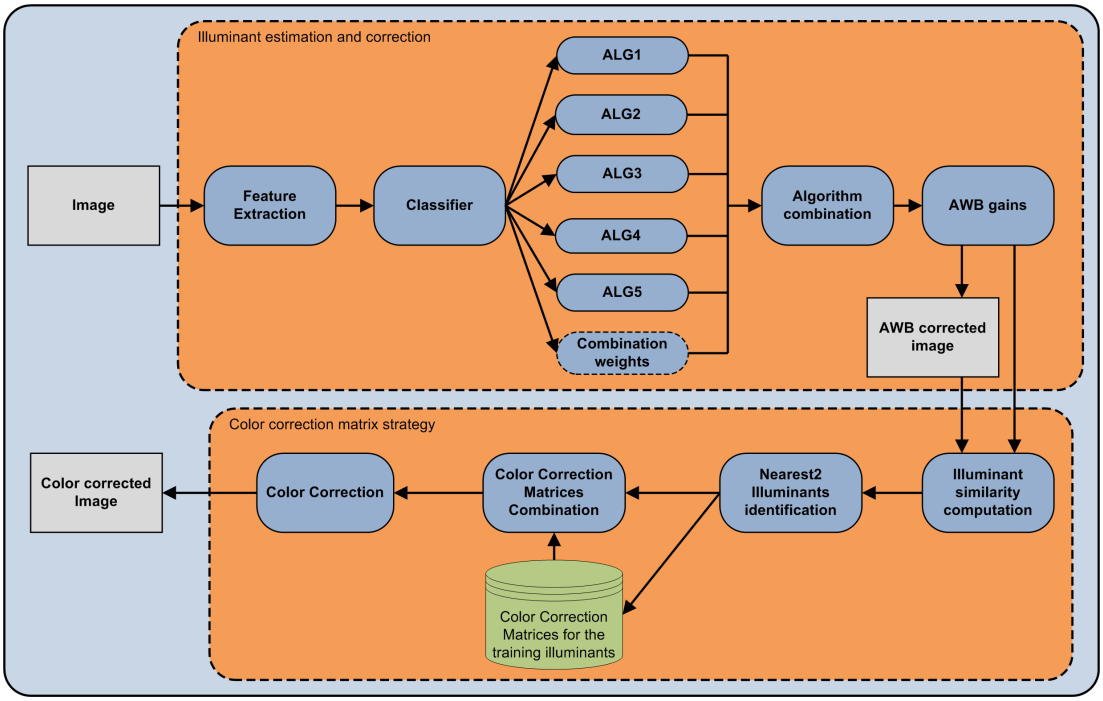


Figure 6.4: Workflow of the pipeline composed of the feature-based AWB and the multiple illuminant color matrixing with white balance error buffer (FB-MILLAWEB)

Chapter 7

Conclusions

Digital camera sensors are not perfect and do not encode colors in the same way in which the human eye does. A processing pipeline is thus needed to convert the RAW image acquired by the camera to a faithful artifact-free representation of the original scene. The modules of which the processing pipeline is composed of, can be divided into four different categories. The first one is composed of modules dedicated to the systematic sensor data correction, such as the sensor non linearity, the dark floor subtraction and the structured noise reduction. The second one is dedicated to the color filter array data correction, such as the stochastic noise reduction and the exposure and white balance correction. The third one is dedicated to the reconstruction of the full-color image from the bayer-sampled RAW image through the demosaicking module and to the conversion of the encoded colors from the camera-dependent color space to a standard color space. The fourth one is dedicated to the postprocessing modules dedicated to the color rendering and image enhancement.

It is known that the color rendering accuracy of a digital imaging acquisition device is a key factor to the overall perceived image quality Ramanath *et al.* (2005), and that there are mainly two modules responsible for it: the former is the illuminant estimation and correction module, the latter is the color matrix transformation. These two modules together form what may be called the color correction pipeline.

This thesis had the objective to design and test new and more robust modules for the color correction pipeline, studying and exploiting the existing crosstalks

in order to obtain a higher color rendering accuracy.

The first module that has been considered is the illuminant estimation and correction one; many illuminant estimation solutions have been proposed in the last few years although it is known that the problem addressed is actually ill-posed as its solution lack uniqueness or stability Funt *et al.* (1998). To cope with this problem different solutions usually exploit some assumptions about the statistical properties of the expected illuminants and/or of the object reflectances in the scene. Hordley in his review paper Hordley (2006) highlighted two research areas that are important in the context of improving the performance of color constancy algorithms: making additional measurements at the time of image capture (i.e. using more color pixel information), and algorithm combining (i.e. using two or more estimations of the illuminants). In this thesis a third hypothesis has been investigated: the use of low level visual information to improve illuminant estimation. This hypothesis has been investigated in two different ways:

- The selected low-level features are fed to a classifier which classifies the images in the indoor/outdoor classes; each image is then processed with the best AWB algorithm for the predicted class (class-based strategy in section 4.6). If the classifier is not confident enough about the membership of the image to the indoor or to the outdoor class, then the image is assigned to the uncertain class, and the image is processed with the best general purpose AWB algorithm. The experimental results on a widely used dataset of real images showed that the use of this strategy improved the illuminant estimation precision in terms of angular error by 20.81% with respect to the best single AWB algorithm in the state of the art and by 14.08% with respect to the best combining AWB algorithm in the state of the art Bianco *et al.* (2008a,b).

- The selected low-level features are fed to a classifier which no longer classifies the images in semantic classes. The classifier predicts which AWB algorithm is best for the given image and processes it accordingly (feature-based strategy in section 4.7). The experimental results on a widely used dataset of real images showed that the use of this strategy improved illuminant estimation precision in terms of angular error by 31.99% with respect

to the best single AWB algorithm in the state of the art and by 26.21% with respect to the best combining AWB algorithm in the state of the art Bianco *et al.* (2009b,c).

The second module that has been considered is the transformation of the camera-dependent RGB image data into a standard RGB color space. This transformation, usually called color correction matrix or color matrixing, is needed because the spectral sensitivity functions of the sensor color channels rarely match those of the desired output color space (usually sRGB Stokes *et al.* (1996)).

The color correction matrix transformation is usually optimized assuming that the illuminant in the scene has been successfully estimated and compensated for Bianco *et al.* (2007); Hubel *et al.* (1997). Both the illuminant estimation process and the color correction matrix concur in the formation of the overall perceived image quality. The two processes have always been studied separately, thus ignoring the interactions between them.

In this thesis the interactions between the illuminant estimation process and the color correction matrix in the formation of the overall color accuracy have been investigated, especially when the white point estimation is imperfect. It has also been investigated how the color correction transform amplifies the illuminant estimation errors. Furthermore, it has been shown that it is possible to incorporate knowledge about the illuminant estimation behavior in the optimization of the color correction matrix to alleviate the error amplification. It has been demonstrated that a fixed device chromatic response characterization, which is often adopted, is not able to produce good color accuracy in most situations. New strategies to improve color accuracy under both perfect and imperfect white point estimation have been proposed, which clearly suggests that adaptive color transformations have to be preferred in order to improve the color accuracy. Experimental results on a dataset of synthetic images showed that the use of multiple color correction matrices optimized for different illuminants improved the ΔE_{94} colorimetric error by 19.72% if an ideal AWB agorithm is used, with respect to the state of the art solution of using a single color correction matrix. These performances drop to a 9.72% of improvement if a real AWB algorithm is used. In order to bridge this gap, a new strategy has been proposed, which

incorporates knowledge about the behavior of the AWB algorithm used. This strategy, in its single-matrix implementation, improved the ΔE_{94} colorimetric error by 9.55% with respect to the state of the art solution, while in its multiple-matrix implementation, improved the ΔE_{94} colorimetric error by 19.21% Bianco *et al.* (2009a).

In order to test the improvement of color rendering accuracy of a color correction pipeline in which the AWB and the color matrixing strategies proposed are simultaneously present, tests have been conducted on a dataset of RAW images where a MCC chart was present in each image. The investigated illuminant estimation algorithms described in chapter 4 are individually run on the images of the dataset, excluding the MCC chart regions that have been previously cropped. Given these estimations, the illuminant corrections are then performed on the whole images (therefore also including the MCC chart). The color matrix transformations found according to the computational strategies described in chapter 5 are then applied to the whole, white balanced, images. For each processed image, the MCC chart is then extracted and the average RGB values of the central area of each patch are calculated. The color rendition accuracy of the pipeline is measured in terms of average ΔE_{94} error between the CIELAB color coordinates of the color corrected MCC patches and their theoretical CIELAB values that are computed using standard equations from their theoretical RGB values. Since both the AWB strategies needed a training set for the training of the classifier, and the cardinality of this dataset was not enough to divide it into training and test sets, two different approaches have been considered:

- Ideal classifiers are used to investigate the theoretical color rendition accuracy that can be obtained by the proposed processing pipelines. For the class-based AWB approach, an ideal classifier which is able to correctly identify the indoor and outdoor images is used, without using the uncertainty class. For the feature-based approach, an ideal classifier which is able to correctly identify the best AWB algorithm to be used for each image, is exploited.

- The classifiers trained in chapter 4 are used, even if they are not optimal for the new dataset. The new dataset is in fact composed of RAW images, while

the dataset used to train the classifiers is composed of pre-processed images. Therefore the real classifiers used are not optimal for the new dataset.

The experimental results showed that the results obtained for the two modules considered separately are confirmed when used together. The feature-based AWB outperformed the class-based AWB in all the implemented pipelines. Concerning the color matrixing module, the multiple illuminant approach (MILLA) outperformed the single illuminant approach (SILL) in all the implemented pipelines; the approaches with white balance error buffer (SILL-WEB and MILLA-WEB) outperformed the corresponding approaches which did not incorporate knowledge about the behavior of the illuminant estimation algorithm used (SILL and MILLA), showing that this information can also be incorporated and exploited for real images. Using as a benchmark pipeline the one composed of the best single illuminant estimation algorithm in the state of the art with single illuminant color correction matrix (BS-SILL), the use of the multiple illuminant color correction matrix (BS-MILLA) improves the color rendition accuracy by 7.53%. The use of the single illuminant and multiple illuminant with white balance error buffer (BS-SILLWEB and BS-MILLAWEB) improves the color rendition accuracy by 9.60% and 14.15% respectively. The substitution of the best single illuminant estimation algorithm with the class-based AWB coupled with the ideal classifier further improves the accuracy by a factor in the range of 0.90-2.26% (1.71% on average on the four different color matrixing used, with a standard deviation of 0.58); the substitution with the feature-based AWB coupled with the ideal classifier instead further improves the accuracy by a factor in the range of 12.90-15.17% (13.63% on average on the four different color matrixing used, with a standard deviation of 1.04).

Further improvements for the illuminant estimation and correction module, could be obtained using non-diagonal illuminant corrections Funt & Jiang (2003) or using chromatic adaptation transforms (CAT) to model the illuminant change between the estimated one, present in the acquired scene, and the chosen canonical one. Several CATs exist in the literature CIE (2004), and new and more stable ones have been recently proposed Bianco & Schettini (2009). The way of integrating one or more CATs in the processing pipeline of a digital camera has to be further investigated.

Improvements for the color space transformation module, could be obtained using color transformation matrices with dimensions greater than 3×3. Such transformations have already been successfully studied and used for scanners Bianco *et al.* (2008d,e) and digital cameras Bianco *et al.* (2009d) connected to a PC, so that the memory requirements for the storage of a full size image with more than three color channels is not a problem. In fact, the conversion matrices with larger sizes, e.g. $n \times 3$, require the presence of n different color channels for each pixel, thus increasing the memory requirements. These additional color channels can be obtained by polynomially combining the original R, G, B ones. It has been shown that different devices need different polynomial terms Bianco *et al.* (2008d,e, 2009d). In the future, these methods could be simply integrated within the proposed pipelines and lead to further improvements in the color rendition accuracy.

Appendix A

Appendix A

A.1 Details about Dataset Selection

A video clip is reconstructed from each set of images removing the right part of the images containing the gray sphere. This video clip is then fed to a key frame extraction algorithm Ciocca & Schettini (2006a) which selects a set of candidate images. These images are dynamically selected within the video clip by analyzing the pictorial differences between consecutive images. The algorithm first identifies the shots present in the clip i.e. uninterrupted sequences of images recorded with a single camera and usually taken in a single location. From each shot, a variable number of images (key frames) is extracted. The number of key frames depends on the visual contents of the shot. That is, shots showing high variability in their pictorial contents will have a high number of images extracted. Shots showing little or no variability will have only a single image extracted. The algorithm, by analyzing each shot independently, is able to remove most of the correlated images while preserving the overall structure and contents of the video clip. The set of key frames extracted corresponds to the visual summary of the clip. An example of key frames extracted from the "Camelback" clip is shown in Figure A.1.

The dataset to be used for testing the algorithms should be of significant size

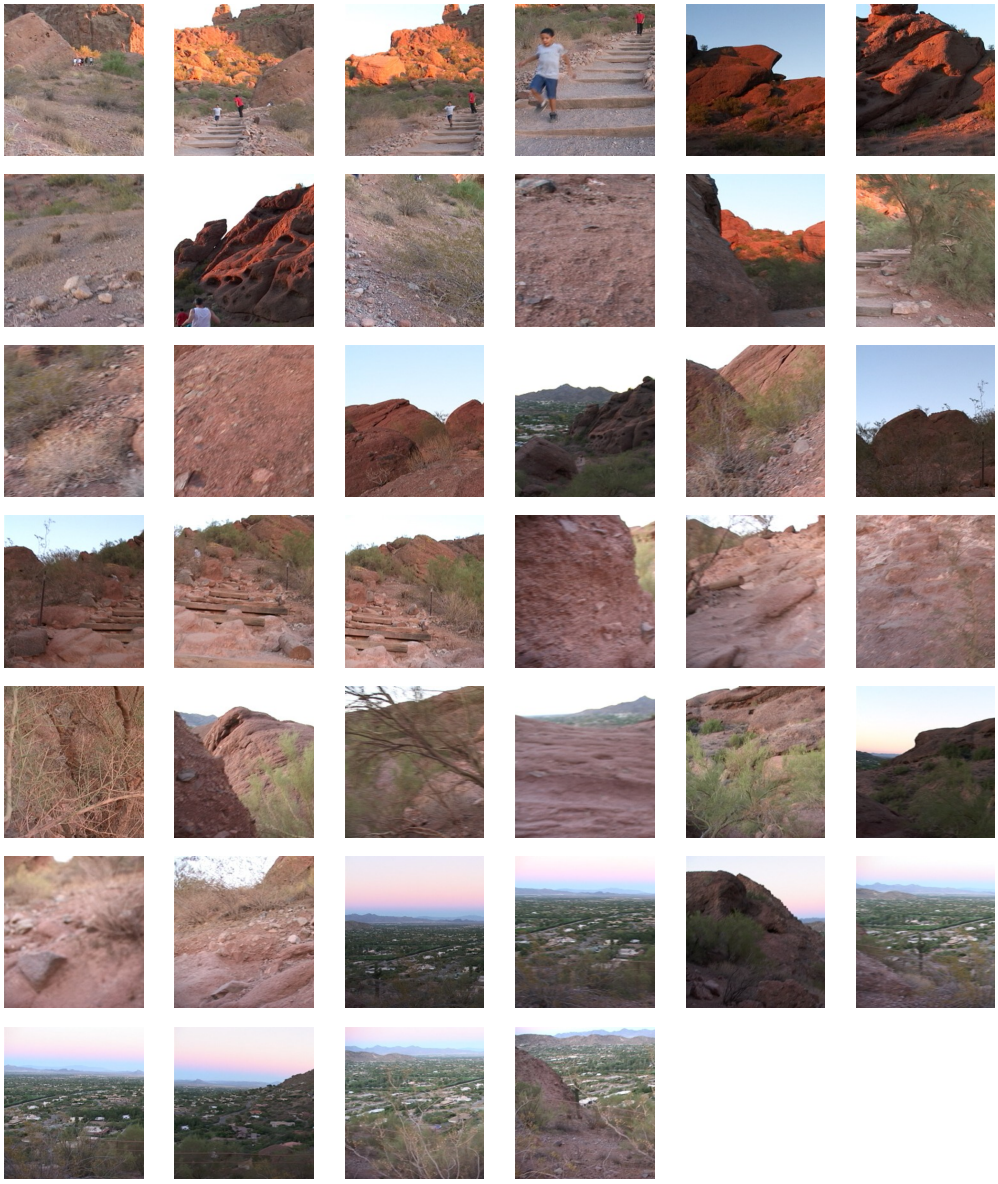


Figure A.1: The 40 Key Frames extracted from the "Camelback" video clip by the key frame extraction algorithm.

and without correlated images. As a trade-off between the number of images and the correlation problem, it has been decided to extract a number of images from each video clip corresponding to 10% of the size of the clip. To do this, the parameters of the key frames extraction algorithm were set so that the generated

visual summary will contain at least that number of key frames. The same parameters were used for all the video clips. To select the exact number of required key frames for each clip, we processed the visual summary further with a visual summary post-processing algorithm Ciocca & Schettini (2006b). The algorithm is composed of three processing steps: key frame removal, key frame grouping and selection of the default visual summary. For this work, only the second processing step was exploited. All the key frames selected by the previous algorithm were processed and the final visual summary was selected according to the number of key frames required for that video clip. An example is shown in Figure A.2.

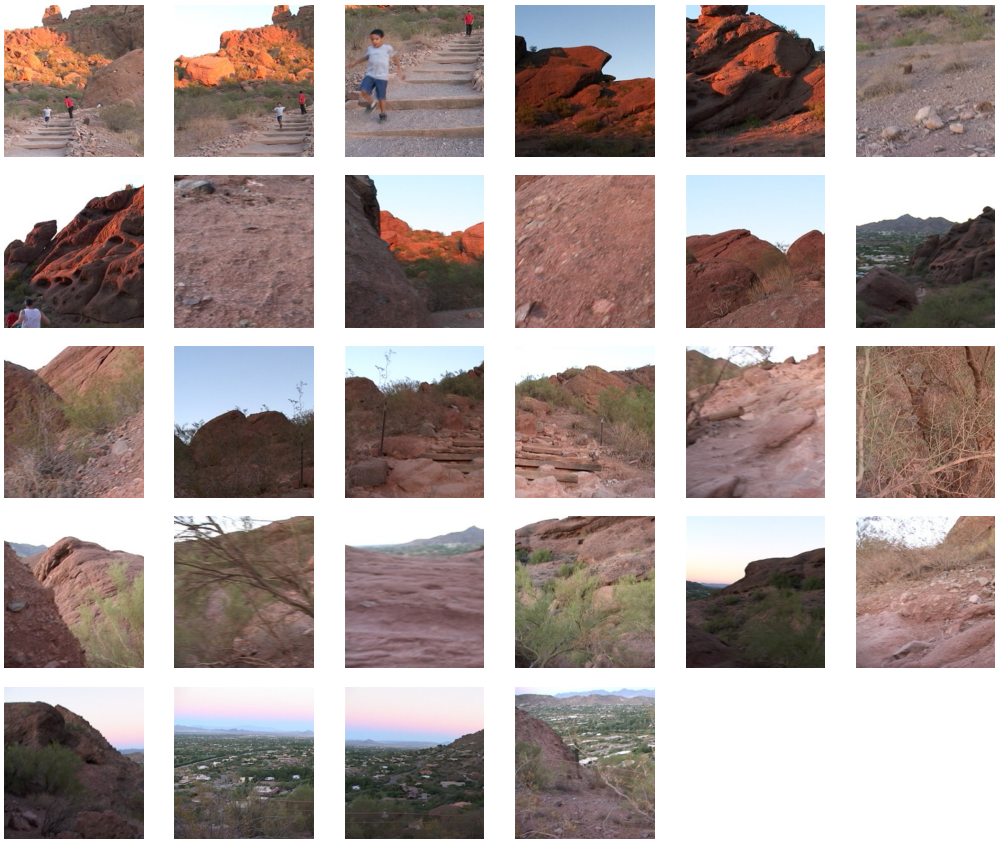


Figure A.2: "Camelback" images included in the final data set. Among the hierarchy of visual summaries generated by the visual summary post processing algorithm, the one containing exactly 28 key frames is selected.

This algorithm is able to iteratively build a hierarchy of visual summaries. The set of initial key frames is processed using a hierarchical clustering algorithm that, at each step, merges consecutive key frames that are visually similar. Thus, the clustering algorithm further removes redundancies within the set of key frames. At each step a new visual summary is generated which contains one image less than the previous one until only a single key frame remains in the summary. The hierarchy can be used to browse the clip contents at different levels of details, and we selected the one which contains the required number of key frames for the given sequence.

Table A.1 shows the statistics of each video clip: video name, number of images contained in each video clip, the number of key frames automatically extracted by the key frame extraction algorithm and the number of images required in our data set (corresponding to 10% of the images in each clip). It can be noted that the number of key frames extracted is very low compared to the number of images in the clip, even with the algorithm parameters set to deliberately extract a large number of key frames. This is an indication that the images within the video clip are truly highly correlated. For similar clips (e.g. CIC2 and CIC3) a similar number of key frames were extracted. Note also that for the "WhiteCliff" video, the key frames extracted are less than the required number of images. In this case, we kept all six images.

Table A.1: Composition of the images in the original 15 video clips, number of extracted key frames and number of images we included in our data set.

	Clip	# Frames	# Key frames	# Required frames
1	Apache	1,273	181	127
2	Burnabay	953	136	95
3	Camelback	276	40	28
4	CIC1	985	156	99
5	CIC2	406	75	41
6	CIC3	499	72	50
7	Deerlake	956	167	96
8	Fallcreek	708	114	71
9	Marine	513	82	51
10	Market1	555	86	56
11	Market2	1,098	144	110
12	Metrotown	1,313	206	131
13	Scottsdale	541	80	54
14	SFU	1,198	184	120
15	WhiteCliff	81	6	8 (6)
	TOTAL	11,355	1,729	1,135

References

Amidror, I. (2002). Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, **11(4)**, 157–176. 69

Antani, S., Kasturi, R. & Jain, R. (2002). Survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, **35**, 945–965. 30

Barnard, K., Cardei, V. & Funt, B. (2002). A comparison of computational color constancy algorithmspart i: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, **11(9)**, 972–983. 26, 80

Barndard, K., Cardei, V. & Funt, B. (2002). A comparison of computational color constancy algorithms; part two: experiments with image data. *IEEE Transactions on Image Processing*, **11(9)**, 985–996. 40

Bayer, B. (1976). Color imaging array, u.s. patent 3 971 065, july 1976. 5

Bianco, S. & Schettini, R. (2009). Two new von kries based chromatic adaptation transforms found by numerical optimization. *Color research and application (accepted)*. 108

Bianco, S., Gasparini, F., Russo, A. & Schettini, R. (2007). A new method for rgb to xyz transformation based on pattern search optimization. *IEEE Transactions on Consumer Electronics*, **53(3)**, 1020–1028. 2, 69, 70, 72, 75, 106

Bianco, S., Ciocca, G., Cusano, C. & Schettini, R. (2008a). Classification-based color constancy. *Proceedings of 10th International Conference on Visual Information Systems (VISUAL2008), LNCS*, **5188/2008**, 104–113. 29, 43, 105

Bianco, S., Ciocca, G., Cusano, C. & Schettini, R. (2008b). Improving color constancy using indoor-outdoor image classification. *IEEE Transactions on Image Processing*, **17(12)**, 2381–2392. xiii, 29, 43, 63, 65, 105

Bianco, S., Gasparini, F. & Schettini, R. (2008c). A consensus based framework for illuminant chromaticity estimation. *Journal of Electronic Imaging*, **17(2)**, 023013. 28, 41, 49, 66, 74

Bianco, S., Gasparini, F., Schettini, R. & Vanneschi, L. (2008d). An evolutionary framework for colorimetric characterization of scanners. *10th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing (EvoIASP2008) incorporated in Evo* 2008, in Applications of Evolutionary Computing, LNCS*, **4974/2008**, 245–254. 109

Bianco, S., Gasparini, F., Schettini, R. & Vanneschi, L. (2008e). Polynomial modeling and optimization for colorimetric characterization of scanners. *Journal of Electronic Imaging*, **17(4)**, 043002. 72, 109

Bianco, S., Bruna, A., Naccari, F. & Schettini, R. (2009a). Color accuracy under varying illuminant estimation precision. *IEEE Transactions on Image Processing (submitted)*. 71, 107

Bianco, S., Ciocca, G. & Cusano, C. (2009b). Color constancy algorithm selection using cart. *Proceedings of 2009 Computational Color Imaging Workshop (CCIW'09), LNCS*, **5646/2009**, 31–40. 29, 63, 106

Bianco, S., G.Ciocca, Cusano, C. & Schettini, R. (2009c). Automatic color constancy algorithm selection and combination. *Pattern recognition (accepted)*. 29, 65, 106

Bianco, S., Schettini, R. & Vanneschi, L. (2009d). Empirical modeling for colometric characterization of digital cameras. *Accepted for IEEE International Conference on Image Processing (ICIP'09)*. 72, 109

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **24**, 123–140. 19, 46

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks/Cole. 18, 45, 58

Buchsbaum, G. (1980). A spatial processor model for object color perception. *J. of Franklin Institute*, **310**, 1–26. 40

Cardei, V. & Funt, B. (1999). Committee-based colour constancy. *Proc. IS&T/SID 7th Color Imaging Conference*, 311–313. 41, 49, 65, 66

Cardei, V., Funt, B. & Barnard, K. (1999). White point estimation for uncalibrated images. *Proc. IS&T/SID 7th Color Imaging Conference*, 97–100. 40

CIE (1932). *Commission internationale de l'Eclairage proceedings*. Cambridge University Press. 11

CIE (2004). 160:2004. a review of chromatic adaptation transforms. *ISBN 9783901906305*, 1–36. 108

Ciocca, G. & Schettini, R. (2006a). An innovative algorithm for key frame extraction in video summarization. *Journal of Real-Time Image Processing*, **1(1)**, 69–88. 38, 110

Ciocca, G. & Schettini, R. (2006b). Supervised and unsupervised classification post-processing for visual video summaries. *IEEE Transactions on Consumer Electronics*, **52(2)**, 630–638. 38, 112

Ciurea, F. & Funt, B. (2003). A large image database for color constancy research. *Proc. IS&T/SID 11th Color Imaging Conference*, 160–164. 25, 28, 37

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**, 273–297. 55

Eakins, J. (2002). Towards intelligent image retrieval. *Pattern Recognition*, **35**, 3–14. 30

Farrell, J., Xiao, F., Catrysse, P. & Wandell, B. (2003). A simulation tool for evaluating digital camera image quality. *Image Quality and System Performance*, **5294**, 124–131. 15, 75

Finlayson, G. (1995). Color constancy in diagonal chromaticity space. *IEEE Proc. Fifth International Conference on Computer Vision*, 218–223. 76

Finlayson, G. & Drew, M. (1997). Constrained least-squares regression in color spaces. *Journal of Electronic Imaging*, **6(10)**, 484–493. 70

Finlayson, G. & Hordley, S. (2001). Color by correlation: a simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23(11)**, 1209–1221. 71

Finlayson, G. & Trezzi, E. (2004). Shades of gray and colour constancy. *Proc. IS&T/SID 12th Color imaging conference*, 37–41. 40

Funt, B. & Jiang, H. (2003). Nondiagonal color correction. *Proceedings of 2003 International Conference on Image Processing (ICIP 2003)*, **1**, I–481–4. 108

Funt, B., Barnard, K. & Martin, L. (1998). Is machine colour constancy good enough? *Proc. 5th European Conf. on Computer Vision*, 445–459. 1, 27, 105

Gasparini, F. & Schettini, R. (2004). Color balancing of digital photos using simple image statistics. *Pattern Recognition*, **37(6)**, 1201–1217. 28, 37

Gehler, P., Rother, C., Blake, A., Minka, T. & Sharp, T. (2008). Bayesian color constancy revisited. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 1–8. 26, 91

Gijsenij, A. & Gevers, T. (2007). Color constancy using natural image statistics. *Proc. International Conference on Computer Vision and Pattern Recognition*, 1–8. 29

Gijsenij, A., Gevers, T. & Lucassen, M. (2008). A perceptual comparison of distance measures for color constancy algorithms. *Lecture Notes In Computer Science, Proceedings of the 10th European Conference on Computer Vision (ECCV'08)*, **5302**, 208–221. 82

Gong, Y., Chuan, C. & Xiaoyi, G. (1996). Image indexing and retrieval using color histograms. *Multimedia Tools and Applications*, **2**, 133–156. 33

Hartigan, J. & Wong, M. (1979). Algorithm as136: A k-means clustering algorithm. *Applied Statistics*, **28**, 100–108. 97

Hordley, S. (2006). Scene illuminant estimation: Past, present, and future. *Color Research & Application*, **31(4)**, 303–314. 1, 28, 69, 105

Hordley, S. & Finlayson, G. (2004). Re-evaluating color constancy algorithms. *Proc. 17th Int. Conf. on Pattern Recognition*, 76–79. 20, 42, 51

Hubel, P., Holm, J., Finlayson, G. & Drew, M. (1997). Matrix calculations for digital photography. *Proceedings of the IS&T/SID Fifth Color Imaging Conference*, 105–111. 2, 69, 70, 75, 106

Hung, P. (1988). Color rendition using three-dimensional interpolation. *Imaging applications in the Work World, Proc. SPIE 900*, 111–115. 69

Hung, P. (1991). Colorimetric calibration for scanners and media. *Proc. SPIE 1448*, 164–174. 69

Idris, F. & Panchanathan, S. (1997). Storage and retrieval of compressed images using wavelet vector quantization. *Journal of Visual Languages and Computing*, **8**, 289–301. 33

ISO (2003). Graphic technology - standard object colour spectra database for colour reproduction evaluation (socs), technical report iso/tr 16066:2003(e). 26, 80

ITU-R (1995). Rec. bt.601: Encoding parameters of digital television for studios, itu, geneva, switzerland. 35

Johnson, G. & Fairchild, M. (2003). A top down description of s-cielab and ciede2000. *Color Research and Application*, **28**, 425–435. 23, 24

Lewis, R. & Torczon, V. (1997). On the convergence of pattern search algorithms. *SIAM J. on Optimization*, **7**, 1–25. 17, 43

Lewis, R. & Torczon, V. (1999). Pattern search algorithms for bound constrained minimization. *SIAM J. on Optimization*, **9**, 1082–1099. 17

Lewis, R. & Torczon, V. (2000). Pattern search methods for linearly constrained minimization. *SIAM J. on Optimization*, **10**, 917–941. 17, 43

Lukac, R. (2008). *Single-sensor imaging: methods and applications for digital cameras*. CRC Press. 5

M.J. Vrhel, H.T. (1992). Color correction using principal components. *Color Research and Application*, **17**, 328–338. 70

M.J. Vrhel, H.T. (1993). Optimal scanning filters using spectral reflectance information. *Human Vision, Visual Processing and Digital Display IV, Proc. SPIE 1913*, 404–412. 70

Mojsilovic, A., Rackov, D. & Popovic, M. (2000). On the selection of an optimal wavelet basis for texture characterization. *IEEE Transactions on Image Processing*, **9(12)**, 2043–2050. 34

Pascale, D. (2006). Rgb coordinates of the macbeth colorchecker. *http://www.babelcolor.com/download/*, 1–16. 92

Ramanath, R., Snyder, W., Yoo, Y. & Drew, M. (2005). Color image processing pipeline. *IEEE Signal Processing Magazine*, **22(1)**, 34–43. 1, 5, 69, 104

Schaefer, G., Hordley, S. & Finlayson, G. (2005). A combined physical and statistical approach to colour constancy. *Proc. International Conference on Computer Vision and Pattern Recognition*, 148–153. 65

Schettini, R., Brambilla, C., Ciocca, G., Valsasna, A. & Ponti, M.D. (2002a). A hierarchical classification strategy for digital documents. *Pattern Recognition*, **35**, 1759–1769. 18, 46, 58

Schettini, R., Ciocca, G. & Zuffi, S. (2002b). Indexing and retrieval in color image databases. *Color Imaging Science: Exploiting Digital Media*, 183–211. 30

Schettini, R., Brambilla, C., Cusano, C. & Ciocca, G. (2003). On the detection of pornographic digital images. *Visual Communications and Image Processing, Proc. SPIE 5150*, 2105–2113. 46

Schettini, R., Brambilla, C., Cusano, C. & Ciocca, G. (2004). Automatic classification of digital photographs based on decision forests. *International Journal of Pattern Recognition and Artificial Intelligence*, **18(5)**, 819–845. 18, 45, 46, 58

Scheunders, P., Liven, S., de Wouwer, G.V., Vautrot, P. & Dyck, D.V. (1997). Wavelet-based texture analysis. *International Journal Computer Science and Information management*, **1(2)**, 22–34. 33

Schröder, M. & Moser, S. (2001). Automatic color correction based on generic content-based image analysis. *Proc. Color Imaging Conference 9*, 42–45. 28

Seber, G. (1977). *Linear regression analysis*. John Wiley & Sons, New York. 70

Shapire, R. (1999). A brief introduction to boosting. *Proc. of 16th Int. Joint Conf. on Artificial Intelligence*, 1401–1406. 55

Sikora, T. (2001). The mpeg-7 visual standard for content description - an overview. *IEEE Transactions on circuits and system for video technology*, **11(6)**, 696–702. 30

Stokes, M., Anderson, M., Chandrasekar, S. & Motta, R. (1996). A standard default color space for the internet - srgb. *www.w3.org/Graphics/Color/sRGB.html*. 2, 11, 12, 69, 106

Stricker, M. & Orengo, M. (1995). Similarity of color images. *Proc. SPIE Storage and Retrieval for Image and Video Databases III Conference*, 381–392. 35

Swain, M. & Ballard, D. (1991). Color indexing. *International Journal of Computer Vision*, **7(1)**, 11–32. 33

Szummer, M. & Picard, R. (1998). Indoor-outdoor image classification. *Proc. Int. Workshop on Content-Based Access of Image and Video databases*, 42–51. 43, 45

Vailaya, A., Figueiredo, M., Jain, A.K. & Zhang, H.J. (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing*, **10(1)**, 117–130. 45

van de Weijer, J., Gevers, T. & Gijsenij, A. (2007a). Edge-based color constancy. *IEEE Transactions on Image Processing*, **16(9)**, 2207–2214. 39, 40, 41, 44

van de Weijer, J., Schmid, C. & Verbeek, J. (2007b). Using high-level visual information for color constancy. *Proc. IEEE 14th International Conference on Computer Vision*, 1–8. 28

von Kries, J. (1902). Chromatic adaptation, [translation: D.l. macadam, *Sources of color science*, mit press, cambridge, 1970]. *Festschrift der Albrecht-Ludwig-Universität*. 29

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, **1**, 80–83. 20, 42, 51

Yang, Y. & Yuille, A. (1991). Sources from shading. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '91*, 534–539. 71

Zhang, X. & Wandell, B. (1997a). A spatial extension of cielab for digital color-image reproduction. *Journal of the Society for Information Display*, **5(1)**, 61–63. 23

Zhang, X. & Wandell, B.A. (1997b). A spatial extension of cielab for digital color-image reproduction. *SID Journal of the Society for Information Display*, **5(1)**, 61–63. 83