

# Exploiting Redundancy in Color Images

THÈSE N° 4582 (2009)

PRÉSENTÉE LE 15 JANVIER 2010

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS  
LABORATOIRE DE COMMUNICATIONS AUDIOVISUELLES 2  
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Joanna MARGUIER

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury  
Prof. S. Süsstrunk, directrice de thèse  
Dr N. Bhatti, rapporteur  
Prof. G. Finlayson, rapporteur  
Prof. R. Hersch, rapporteur



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

Suisse  
2010





# Contents

<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Acknowledgments</b>	<b>ix</b>
<b>Frequently Used Terms, Abbreviations, and Notations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Contributions and overview . . . . .	7
1.1.1 Color assessment of skin tones . . . . .	7
1.1.2 Color assessment of object colors . . . . .	7
1.1.3 Illuminant retrieval for fixed location cameras . . . . .	8
1.1.4 Overview . . . . .	8
<b>2 General background</b>	<b>11</b>
2.1 Image formation model . . . . .	11
2.2 Trichromacy, colorimetry, and metamerism . . . . .	13
2.3 In-camera processing . . . . .	14
2.4 Color spaces . . . . .	14
2.4.1 XYZ color space . . . . .	16
2.4.2 sRGB . . . . .	18
2.4.3 CIELAB . . . . .	20
2.4.4 CIECAM02 . . . . .	22
2.5 Color constancy and white-balancing . . . . .	23
2.6 Linear models for illuminants and reflectances . . . . .	24
2.7 Color constancy algorithms . . . . .	26
2.7.1 Single light methods . . . . .	26
2.7.2 Multiple lights methods . . . . .	28
2.7.3 Illuminant estimation error measure . . . . .	30
2.8 Summary . . . . .	31
<b>3 Skin color assessment from uncalibrated images</b>	<b>33</b>
3.1 Introduction . . . . .	33
3.2 Background . . . . .	35
3.2.1 Cameras as colorimeters . . . . .	35
3.2.2 Skin color measurement . . . . .	35
3.3 Our approach . . . . .	37
3.3.1 Reference target . . . . .	37

---

3.3.2	Color transform . . . . .	39
3.3.3	Approximation by a linear transform . . . . .	41
3.4	Experiments and results . . . . .	42
3.4.1	Color correction accuracy . . . . .	42
3.4.2	Computation of the transform in linear sRGB . . . . .	47
3.4.3	Face color derived from spectral measurements . . . . .	50
3.4.4	Color correction across cameras . . . . .	51
3.5	Discussion . . . . .	57
3.6	Summary . . . . .	57
<b>4</b>	<b>Color assessment for a Home Décor application</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	Our approach . . . . .	60
4.2.1	Reference targets . . . . .	61
4.2.2	Color transform . . . . .	62
4.2.3	Matching metric . . . . .	62
4.2.4	The algorithm . . . . .	63
4.3	Experiments and results . . . . .	64
4.3.1	sRGB gamut size and white-balance . . . . .	65
4.3.2	Color correction accuracy . . . . .	69
4.3.3	Color correction across cameras . . . . .	73
4.3.4	Color correction and expert's rating . . . . .	80
4.4	Discussion . . . . .	84
4.5	Summary . . . . .	85
<b>5</b>	<b>Illuminant retrieval for fixed location cameras</b>	<b>87</b>
5.1	Introduction . . . . .	87
5.2	Our approach . . . . .	88
5.2.1	Linear models and illuminant retrieval . . . . .	88
5.2.2	Building the error function . . . . .	89
5.2.3	Spherical sampling and daylight illuminants . . . . .	91
5.2.4	Gradient descent . . . . .	91
5.2.5	Illuminant and image normalization . . . . .	93
5.3	Experiments and results . . . . .	94
5.3.1	Standard daylight illuminants . . . . .	95
5.3.2	Real daylight illuminants . . . . .	101
5.3.3	Comparison with D'Zmura and Iverson's algorithm . . . . .	105
5.3.4	Gradient descent in presence of noise . . . . .	109
5.3.5	Comparison with a single image algorithm . . . . .	110
5.3.6	Artificial illuminants . . . . .	113
5.3.7	Reflectances . . . . .	117
5.4	Discussion . . . . .	124
5.5	Summary . . . . .	127
<b>6</b>	<b>Conclusion</b>	<b>129</b>
6.1	Color assessment from uncalibrated images . . . . .	129
6.1.1	Summary . . . . .	129
6.1.2	Future research . . . . .	130
6.2	Illuminant retrieval for fixed location cameras . . . . .	131
6.2.1	Summary . . . . .	131

---

6.2.2	Future research . . . . .	131
<b>A</b>	<b>CIECAM02 model</b>	<b>133</b>
<b>B</b>	<b>Two-stage linear recovery for bilinear models</b>	<b>137</b>
<b>C</b>	<b>Supplementary material for Chapter 3</b>	<b>141</b>
C.1	Images of the subjects' faces . . . . .	141
C.2	$\gamma_{\text{var}}$ values for Section 3.4.2 . . . . .	151
C.3	Kruskal-Wallis $p$ -values for Section 3.4.2 . . . . .	152
<b>D</b>	<b>Historical review of color harmony theories</b>	<b>153</b>
<b>E</b>	<b>Supplementary material for Chapter 4</b>	<b>157</b>
E.1	Decoration samples and complementing color palettes . . . . .	157
<b>F</b>	<b>Supplementary material for Chapter 5</b>	<b>179</b>
F.1	MatLab code . . . . .	179
F.1.1	Parameters . . . . .	183
F.2	Test illuminants . . . . .	185
F.3	Illuminant and reflectance basis functions . . . . .	192
F.4	Additional results . . . . .	194
F.4.1	Error distributions for standard daylight illuminants . . . . .	194
F.4.2	Error distributions for real daylight illuminants . . . . .	196
F.4.3	Error distributions for artificial illuminants . . . . .	202
F.4.4	Comparison with D'Zmura and Iverson's algorithm . . . . .	204
F.4.5	Error distributions for standard daylight illuminants (SoG) . . . . .	205
F.4.6	Errors for real daylight illuminants (SoG) . . . . .	207
F.4.7	Error distributions for real daylight illuminants (SoG) . . . . .	208
F.4.8	Errors for Shades of Gray in presence of shot noise . . . . .	211
F.4.9	Reflectances estimates . . . . .	215
	<b>Bibliography</b>	<b>225</b>
	<b>Curriculum Vitae</b>	<b>233</b>



# Résumé

Lorsque l'on capture une scène à l'aide d'un appareil photographique, les couleurs de l'image résultante dépendent non seulement des propriétés de réflexion des éléments constituant la scène, mais également de la lumière ambiante lors de la prise de vue ainsi que des caractéristiques de l'appareil. Il est souvent nécessaire de dissocier les contributions de ces trois éléments, notamment de séparer l'influence des propriétés de réflexion des objets de celle de la lumière, grandeurs définies sur toute la gamme de longueurs d'onde visibles. La détermination d'illuminant ou de réflectances à partir d'une image RGB est un problème sous-déterminé qui requiert des connaissances ou hypothèses supplémentaires pour sa résolution.

Dans cette thèse, nous étudions comment de l'information redondante présente dans une série d'images peut être exploitée afin d'en déterminer l'illuminant ou les réflectances. Nous démontrons qu'il est possible de déterminer avec précision la couleur d'objets sous la forme de valeurs tristimulus à partir d'une image RGB standard capturée par une caméra inconnue sous des conditions d'éclairage non contrôlées et sans accès aux données du capteur. La scène contient par contre un nombre réduit de couleurs de référence en la forme d'une cible de calibration spécifique. La correction doit être limitée à une gamme restreinte de couleurs similaires à celle de l'objet qui seront comparées à des références permettant de calculer une transformation linéaire. Cette méthode est utilisée dans deux applications visant à fournir un conseil personnalisé au consommateur en maquillage et décoration d'intérieur, respectivement. La précision de la correction est de  $\Delta E_{ab}^* \simeq 1$  à 2 selon la composition de la cible, ce que nous démontrons être suffisant pour les deux applications présentées et permettant des résultats satisfaisants.

Nous avons également développé une méthode permettant de calculer le spectre d'illuminants pour une série d'images capturées par une caméra fixe. Dans ces images, il y a aussi bien des éléments variables, tels des changements de lumière et de contenu, que des éléments constants. Nous proposons d'utiliser ces derniers comme références afin de résoudre le problème de la constance des couleurs. Plus exactement, nous exploitons le fait que les réflectances de ces objets, bien qu'inconnues, sont les mêmes dans chaque image. Nous inversons une série de modèles de formation d'images en parallèle et, en forçant les réflectances résultantes à être égales, déduisons l'illuminant de chaque image. Des simulations utilisant des valeurs RGB synthétiques modélisant les éléments constants ont démontré qu'il était possible de déterminer les illuminants naturels avec une erreur angulaire médiane inférieure à 3°.

**Mots-clés:** Constance des couleurs, balance des blancs, redondance, illuminant.



# Abstract

The colors in a digital image do not only depend on the scene’s reflectances, but also on the lighting conditions and on the characteristics of the imaging device. It is often necessary to separate the influence of these three elements, in particular to separate the contribution of the object’s reflection properties from the one of the incident light. Both are usually represented by spectral quantities, i.e., defined over the visible spectrum. The retrieval of illuminant and reflectances from an RGB image, which provides three values per pixel, is an under-determined problem and requires additional information or assumption on the scene’s content in order to be solved.

In this thesis, we study how redundant information across a set of images can be exploited to recover either illuminant or reflectance knowledge. We demonstrate that accurate object color can be retrieved as tristimulus values from standard RGB images captured with an unknown camera under uncontrolled lighting conditions and no access to raw data, but containing a limited number of reference colors in the form of a specific calibration target. The correction must be constrained to a limited range of colors similar to the one of the object of interest, which are compared to pre-computed references for the derivation of a linear transform. We implement this method for two consumer oriented applications in Makeup and Home Décor, respectively, which provide users with color advice. The correction accuracy is of  $\Delta E_{ab}^* \simeq 1$  to 2 depending on the design of the target. We show that this accuracy is sufficient for the applications at hand.

We also developed a method to retrieve illuminant spectra from a set of images taken with fixed location cameras, such as panoramic or surveillance ones. In pictures captured with such devices, there will be changes in lighting and dynamic content, but there will also be constant objects. We propose to use these elements as reference colors to solve for color constancy. More precisely, we exploit that their reflectances, while unknown, remain constant across images. We invert a series of image formation models in parallel for a set of test illuminants and, by forcing the output reflectances to match, deduce the illuminant under which each image was captured. Simulations on synthetic RGB patches demonstrate that real daylight illuminants can be retrieved with a median angular under  $3^\circ$  from sets containing as little as four images using a limited number of reference surfaces.

**Keywords:** Color constancy, white-balancing, redundancy, illuminant.





# Acknowledgments

First of all, I would like to thank my advisor Prof. Sabine Süsstrunk for her support and the opportunity of doing a PhD in the IVRG group. I was also lucky to work abroad and I wish to thank Dr. Nina Bhatti, who supervised my two internships at Hewlett-Packard Laboratories, in Palo Alto, in 2006 and 2008. Our collaboration went beyond these two summers in California to become an important part of the present thesis. Not only have they provided me with a friendly and intellectually stimulating environment, but they are also both two very inspiring models for a young female scientist.

I would also like to thank the other members of my jury, Prof. Emre Telatar, Prof. Graham Finlayson, and Prof. Roger Hersch for being on my thesis committee and for helping me improving my work through their useful comments and suggestions.

I am also grateful to all my IVRG and LCAV colleagues for their friendship and for the many conversations which, sometimes accidentally, gave me new directions for my work. In particular, I would like to thank Dr. Clément Fredembach for his valuable input.

While a thesis is principally a scientific and academic achievement, it runs over several years in a life and I probably would have never graduated without the help, support, and love of my friends. I would like to heartily thank all the great people that I am lucky to have in my life, in particular Magali Baillif, Daniel Hammer, and Noémi Massard, who have been there for me with a motivational speech and a drink, or two, every time I felt overwhelmed. Finally, I am especially thankful to Fred Semoroz for his unconditional support and love, and for bearing with me during these last months.



# Frequently Used Terms, Abbreviations, and Notations

## Terms and abbreviations

**Achromatic color** Perceived color devoid of hue.

**Brightness** The attribute of visual perception according to which an area appears to emit more or less light.

**Chroma** Colorfulness of a visual sensation relative to an achromatic stimulus of the same brightness.

**Chromatic color** Perceived color possessing a hue.

**Color adaptation or constancy** Ability of the human visual system to maintain the color appearance of an object roughly constant across a wide range of illumination conditions.

**Color space** Generic representation of colors in space, usually of three dimensions (CIE 1987).

**Colorfulness** The attribute of visual perception according to which the perceived color of an area appears more or less chromatic.

**Colorimetry** The part of color science of measurements of physically defined color stimuli and their numerical representation.

**CIE** Commission Internationale de l'Eclairage.

**CMF** Color matching functions.

**D65 illuminant** Standard CIE average daylight illuminant with a correlated color temperature of 6500K.

**HSV** Human visual system.

**Hue** The attribute of color perception denoted by blue, green, yellow, purple, and so on.

**Illuminant** Spectral power distribution of a light source.

**Lambertian** Attribute of a surface reflecting light equally in all directions.

**Lightness** Relative brightness. Variations in lightness range from “light” to “dark.”

**LMS** Least mean square.

**Luma** Weighted sum of gamma-compressed RGB components, denoted by the letter *Y*.

**Luminance** Photometric measure of the luminous intensity.

**Metamerism** Metameric color stimuli have the same tristimulus values, but different reflectance spectra. Metamers are objects that have the same color appearance under one illuminant, but appear different under another illuminant.

**Munsell Color System** Color space specifying colors by three attributes: hue, value (lightness), and chroma.

**Opponent colors theory** Colors are encoded in three opponent channels: *red–green*, *yellow–blue*, and *light–dark*. It explains why combinations such as reddish-green and blueish-yellow do not occur (Hering 1878).

**Output-referred image state** Image state associated with image data that represents the color space coordinates of the elements of a image that has undergone color rendering appropriate for a specified real or virtual device and viewing conditions (ISO 2004).

**Saturation** Colorfulness of a visual sensation relative to an achromatic stimulus regardless of their brightness.

**Scene-referred image state** Image state associated with an image data that represents estimates of the color space coordinates of the elements of a scene (ISO 2004).

**SPD** Spectral power distribution.

**Trichromacy** Condition of possessing three types of independent photoreceptor for color vision.

**Tristimulus values** Generally CIE XYZ tristimulus values. Triplet of values describing a color stimulus under a specified illuminant.

**White-point** Set of tristimulus values of a white surface under a given illuminant.

---

**Notations and variables**

$E(\lambda)$	Illuminant spectra
$\mathbf{e}(\lambda_i), 1 \leq i \leq 31$	Illuminant spectra in vector form
$\mathcal{E}_i(\lambda), 1 \leq i \leq N_\varepsilon$	Illuminant basis functions
$\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_{N_\varepsilon})$	Illuminant descriptors
$\theta_{n_E}, \phi_{n_E}$	Spherical angles indexing daylight illuminants
$N_\varepsilon$	Number of illuminant descriptors
$N_e$	Number of illuminant candidates
$N$	Number of images
$N_E$	Number of illuminants
$S(\lambda)$	Reflectance spectra
$\mathbf{s}(\lambda_i), 1 \leq i \leq 31$	Reflectance spectra in vector form
$\mathcal{S}_i(\lambda), 1 \leq i \leq N_\sigma$	Reflectance basis functions
$\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_{N_\sigma})$	Reflectance descriptors
$N_\sigma$	Number of reflectance descriptors
$R_k(\lambda), 1 \leq k \leq N_\rho$	Sensor sensitivities
$\mathbf{r}_k(\lambda_i), 1 \leq i \leq 31, 1 \leq k \leq N_\rho$	Sensor sensitivities in vector form
$\rho_k(\lambda), 1 \leq k \leq N_\rho$	Sensor responses
$N_\rho = 3$	Number of sensors
$\mathbf{p} = (p_1, \dots, p_{N_p})$	Average patch pixel values
$N_p$	Number of selected patches
$n$	runs over the images
$n_E$	runs over the illuminants
$n_\varepsilon$	runs over the illuminant descriptors
$n_p$	runs over the patch numbers
$n_\sigma$	runs over the reflectance descriptors
$f_e$	Error function
$e_{ang}$	Angular error
$\mathbf{A}, \mathbf{M}, \boldsymbol{\mathcal{E}}, \dots$	Matrices
$\mathbf{x}, \mathbf{p}, \boldsymbol{\varepsilon}, \dots$	Vectors



# Chapter 1

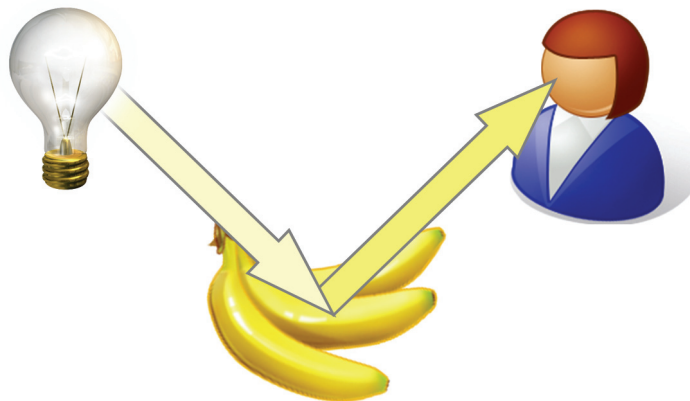
## Introduction

We all know what color is, we experience it everyday. But when it comes to giving a precise definition, we are not so confident of our understanding. So, what is color? According to the *Collier's Dictionary*, color is the:

quality of an object, or substance, perceived as a visual sensation resulting from its transmission or reflection of light of any or all of the various parts of the spectrum.

The *Oxford American Dictionary*, on the other hand, defines color as:

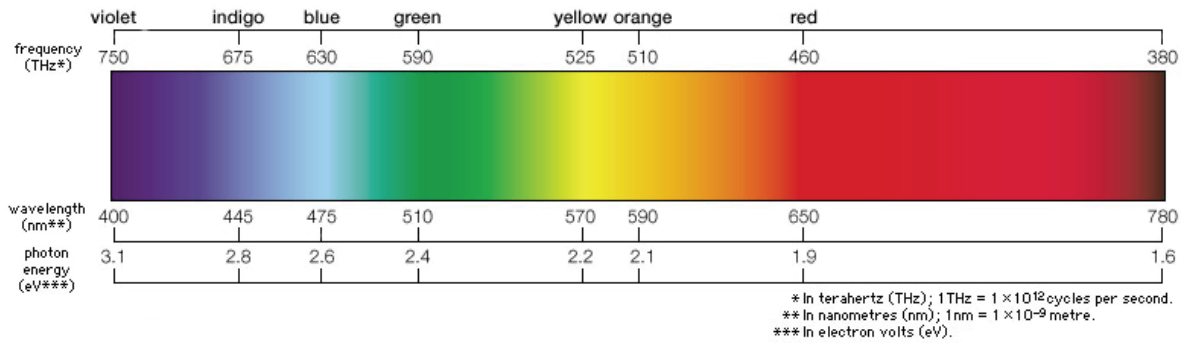
the sensation produced by various rays of light of different wavelength [...].



**Figure 1.1:** Colors arise from the interaction of light reflected by the objects' surfaces with the human visual system.

The comparison of these two definitions points out a fundamental question: Is color an objective property of objects or is it purely subjective? Is color physics or physiology? Color indeed arises from the interaction of light with objects, but only exists as it is seen by an observer, as depicted by Figure 1.1.

From a physical measurements point of view, color depends on the interaction of the visible light with objects. The term *visible light* refers to the part of the electromagnetic spectrum that can be detected by the human eye, which corresponds typically to wavelengths from about 380 to 760 *nm*. Figure 1.2 illustrates the visible spectrum. The energy distribution of a light source,



**Figure 1.2:** The visible spectrum. Source: Encyclopedia Britannica.

or *illuminant*, as a function of the wavelength is given by its *spectral power distribution*. The intrinsic “color” of an object is specified by its *reflectance spectrum*, which describes the percentage of the incoming light that its surface reflects as a function of the wavelength and, in general, of the angle under which we view the surface. A “white” surface, for instance, reflects all visible wavelengths equally, while a “red” surface absorbs the visible wavelengths under about 480 *nm* and only reflects the “red” part of visible light. Mathematically, the light reaching our eye after being reflected by the surface of an object is the product of the light source’s spectral power distribution with the object’s surface reflectance and is called the *color signal*.

From a subjective point of view, the “red” light that enters our eye after being reflected by a “red” object only becomes “red” after it has been captured and processed by the human visual system and interpreted as such. Furthermore, the perceived red does not only depend on the incoming color signal, but also depends on the object’s texture and size, on the illumination level, on the color of the surrounding objects, and on the observer.

The human visual system adapts to the color of the illuminant, a phenomenon referred to as *chromatic adaptation*. As a consequence, we are able to perceive the color of objects mostly unchanged across significant ranges of lighting conditions, which is referred to as *color constancy*. *Memory colors*, which also contribute to color constancy, refer to the phenomenon that many objects have a color associated with them. For example, people have typical memory colors for grass, skin, and blue sky. It has been shown that such colors are often remembered as lighter and more saturated than the actual objects (Bartelson 1960), that is, we recall the sky bluer and the grass greener than they really are. For an overview of these phenomena, see (Fairchild 2005).

In this thesis, we will consider color as a physical property, i.e., that the color of an object is characterized by its reflectance in the sense that once it is known, we can compute the corresponding color signal for any illuminant. Color is regarded as captured by a standard color camera that will be characterized



by the sensitivity of its three sensors to the red, green, and blue portions of the visible light. Image formation will be modeled as

$$\rho_k = \int_{\lambda_{\text{visible}}} S(\lambda)E(\lambda)R_k(\lambda) \quad k = \{\text{red; green; blue}\}, \quad (1.1)$$

where  $S(\lambda)$ ,  $E(\lambda)$ ,  $R_k(\lambda)$  are, respectively: the object's reflectance factor, the illuminant spectral power distribution, and the sensitivities of the imaging device, either the eye or a camera sensor, to the incoming color signal. This simple model cannot account for the color appearance phenomena mentioned earlier, but is a good approximation for digital cameras. If we consider a digital color image, each pixel has three numerical RGB (for Red, Green, and Blue) components  $\rho_k$  to represent a color. These components are modeled by Equation (1.1).

We are now asking the following question: What information on the object and light can be retrieved from a standard RGB image? More precisely, can we separate the influence of the illuminant from the influence of the reflectances using only the information given by the image?

This is not trivial. The first difficulty comes from the ambiguity of the information carried by a color signal about both the reflectance of objects and the light. As the color signal enters our eye or a camera, the information on the illuminant and reflectance is already combined: A red surface viewed under a white light may give rise to the exact same color signal as a white surface viewed under a red light. The second difficulty is that retrieving spectral quantities, i.e., defined over the visible wavelengths, from only three pixel values per location is an ill-posed problem. Formally, at each pixel, we have three equations of the form (1.1); the reflectance and illuminant being integrated over the visible wavelengths, the spectral information is lost and the image formation cannot be inverted exactly.

There are many applications that require to separate the influence of the illuminant and reflectance. For example, it is generally desirable for image pixels to be representative of the actual surface reflectance of the object independently of the illuminant. *White-balancing* refers to the operation of rendering a scene as if it had been captured under a pre-defined illuminant by globally adjusting the proportion of the red, green, and blue in an image. Automatic white-balancing is the camera's equivalent of the human visual system's chromatic adaptation. The scaling coefficients are computed from the image content through a *color constancy algorithm*, which often returns an estimation of the illuminant in the form of its *white-point*, that is, the RGB values of a white surface viewed under the scene's illuminant.

Illuminant compensation plays also an important role in some computer vision applications: Color has been shown to bring useful information in tasks such as object recognition or segmentation. It is indeed an object property that is mostly independent of its geometry and can thus be efficiently employed in object indexing algorithms (Swain and Ballard 1991), for example. However, in order to be accurately representative of an object, image colors must be illuminant independent (Funt *et al.* 1998, Finlayson *et al.* 2001b).

Mathematically, color constancy is an ill-posed problem that cannot be solved exactly. Many approaches have been proposed over the last 30 years to reduce the dimension of this under-determined problem and return useful

illuminant descriptors, such as introducing assumptions on image statistics (Buchsbbaum 1980, Gershon *et al.* 1988), on surface properties such as specular highlights (Lee 1986), or by using redundancy in two, or more, images (D’Zmura and Iverson 1993a, DiCarlo *et al.* 2001, Finlayson *et al.* 2005, Lu and Drew 2006).

In this thesis, we will investigate how we can exploit redundant information across color images to retrieve reflectance and illuminant information. More precisely, we focus on two different problems, namely recovering illuminant and camera independent color values from standard RGB images and retrieving illuminant spectra in a set of images taken with fixed location cameras.

We know (Eq. 1.1) that the resulting pixel values in a color image do not only depend on the actual color of the objects in the photograph, but also depend on the illuminant and camera characteristics. Images are processed to compensate for these two factors, but this is never perfect, which is an issue for online shopping, for example. A consumer may wrongly pick a garment according to its color on the screen, which may differ significantly from its actual color. The first method we present offers a solution to overcome this issue in two specific cases: Makeup and Home Décor. In the first situation, the customer takes a self-portrait with her own camera and sends it via multimedia message (MMS) with her cell phone to the advisory service, which will assess the foundation that suits best her skin tone relying only on the incoming image. No additional information on the camera and on the illuminant is available. The subject is asked to hold a calibration target while taking the picture. This target is designed specifically for the application and helps correcting the image such that the face pixels are representative of the actual skin tone and can be employed to return appropriate makeup advice. We then adapt this method to Home Décor. Let us assume a consumer to have a green couch and to need advice on which wall color would complement its color. We suggest that the customer images the couch, or any decoration item, along with a specific calibration target and sends the picture to a cell phone or internet based service, which will take the target information to correct the couch color and return a set of wall color suggestions.

In the second part of this thesis, we consider a quite different framework. We are working with sets of images that have been captured in a given place with a fixed camera. This corresponds, for example, to the case of security cameras. In images taken with such devices, the lighting conditions will vary (at different times of the day or under changing weather) as most of the content (people or cars passing by). However, there are static objects, such as buildings, that will be present in all the images. While we may not know the reflectance properties of these elements, we know that they will not change from day to day. Formally, it means that, while the reflectance spectra  $S(\lambda)$  in Equation (1.1) are unknown, they remain constant in all the images, which allows us to retrieve the illuminant spectra  $E(\lambda)$  in all the images taken with a fixed location camera.

---

## 1.1 Contributions and overview

### 1.1.1 Color assessment of skin tones

We propose a method for the color correction of skin tones from standard RGB images taken with uncalibrated cameras and under uncontrolled illumination conditions. A subject is imaged with a calibration target of reference colors covering the range of human skin tones that we use to correct the subject's image by way of a linear transform. Once the image is color corrected, the face color is in a reference space. In other words, the pixel values are correlated with the actual face color and are independent from the lighting conditions and camera. This method is implemented in a makeup advisory service using cell phone cameras. The user takes a self-portrait holding a copy of the calibration target and sends the image via multimedia message (MMS). After color correction, the face color estimate is compared to an existing database of makeup recommendations built by an expert. The user's skin tone is classified and the system returns a foundation color advice according to the expert's opinion.

The specificity of the method lies in the design of the correction target, which is application dependent. It is not possible to accurately correct all colors in an image with a linear transform. However, we demonstrate that a small range of colors, skin tones in this particular example, can be properly corrected with this simple procedure. The values obtained correlate well with color values derived from in-vivo spectral measurements and are consistent across a range of cameras of different quality and resolutions.

### 1.1.2 Color assessment of object colors

Although initially designed for skin tones, our color correction procedure can be applied to any color by using a set of targets, as we demonstrate in the specific case of a Home Décor application. In this context, the user images a decoration object along with an appropriate color correction target and sends the picture to either a cell phone or internet based service that will return a set of colors complementing the object. As previously, the color of the object is corrected in order to discard the influences of the lighting conditions and of the camera characteristics and compared to an existing database of color recommendations built by a interior designer. As the color correction is accurate for limited ranges of color only, the system uses several color calibration targets to cover all hues.

We demonstrate that the accuracy of the method depends on the similarity of the object and reference target colors. We propose a method to quantify this similarity with volumes spanned by target patch values in the sRGB space, which can be used to give feedback when the color correction is not accurate or to help the user select the appropriate target.

While this color correction method is presented as integrated into two specific applications, these results are more fundamental. It shows that we can obtain an accurate colorimetric estimate of any color object imaged with an unknown consumer camera and illumination by imaging the object with an adapted correction target.

### 1.1.3 Illuminant retrieval for fixed location cameras

In Chapter 5, we present a method for retrieving illuminant spectra from a set of images taken with a fixed location camera, such as a surveillance or panorama camera. In such images, there will be significant changes in lighting conditions and scene content, but there will also be static elements in the background. As color constancy is an under-determined problem, we propose to exploit the redundancy and constancy offered by the static elements of the images to reduce the dimensionality of the problem. Specifically, we assume that the reflectance properties of these objects remain constant across the images taken with a given fixed camera. With respect to the other problem discussed above, the static elements of the scene become the “color target,” although with unknown reflectances. Here, we exploit the fact that we can take several images of the same scene.

We demonstrate that we can retrieve illuminant and reflectance spectra in this framework by modeling the redundant image elements as a set of synthetic RGB patches. We define an error function that takes the RGB patches and a set of test illuminant spectra as input and returns a similarity measure of the redundant surfaces reflectances. The test illuminants are then varied until the error function is minimized, returning the illuminants under which each image in the set was captured.

We suggest two function minimization methods: selecting a set of illuminant spectral power distributions that constitute scene illuminant candidates, we minimize the error function by testing all illuminant combinations. This approach avoids expressing the illuminant using linear models and can thus be applied to any illuminant type. However, it becomes computationally prohibitive if a large set of test illuminants or a large number of images is considered. We thus propose to minimize the error function by gradient descent. It is applied to illuminants that can reasonably be expressed as a linear combination of three basis functions. The error function is then defined with the illuminant descriptors as variables and minimized by gradient descent.

### 1.1.4 Overview

**Chapter 2** covers several notions of color science. We review a simple model for image formation, several important color spaces, a generic in-camera processing workflow, color transforms, and linear models for illuminant and reflectance spectra. We also review important color constancy algorithms and illuminant estimation error measures. **Chapter 3** presents a method allowing an accurate color assessment of skin tones from uncalibrated images. **Chapter 4** explains how the color assessment designed for skin tones can be modified and extended to correct any object color by using several color calibration targets. The method is applied to a Home Décor application. **Chapter 5** presents an approach for the retrieval of illuminants in a set of images containing several redundant surfaces, corresponding to the case of fixed location cameras. The method is tested on standard daylight illuminants, real daylight illuminants, and artificial illuminants. **Chapter 6** concludes the thesis.

**Appendices A** and **B** report the details of the color appearance model CIECAM02 and of the two-stage recovery of illuminant and reflectance spectra (D’Zmura and Iverson 1993*a*), respectively. **Appendix C.1** shows the im-

---

ages of all the subjects from the skin color assessment experiments reported in Chapter 3. **Appendix D** presents a historical review of color harmony theories. **Appendix E** presents the images of the decoration samples used in the experiments reported in Chapter 4. **Appendix F** contains supplementary material for Chapter 5, such as details on the algorithm’s code, plots of the test illuminants, and additional results.



## Chapter 2

# General background

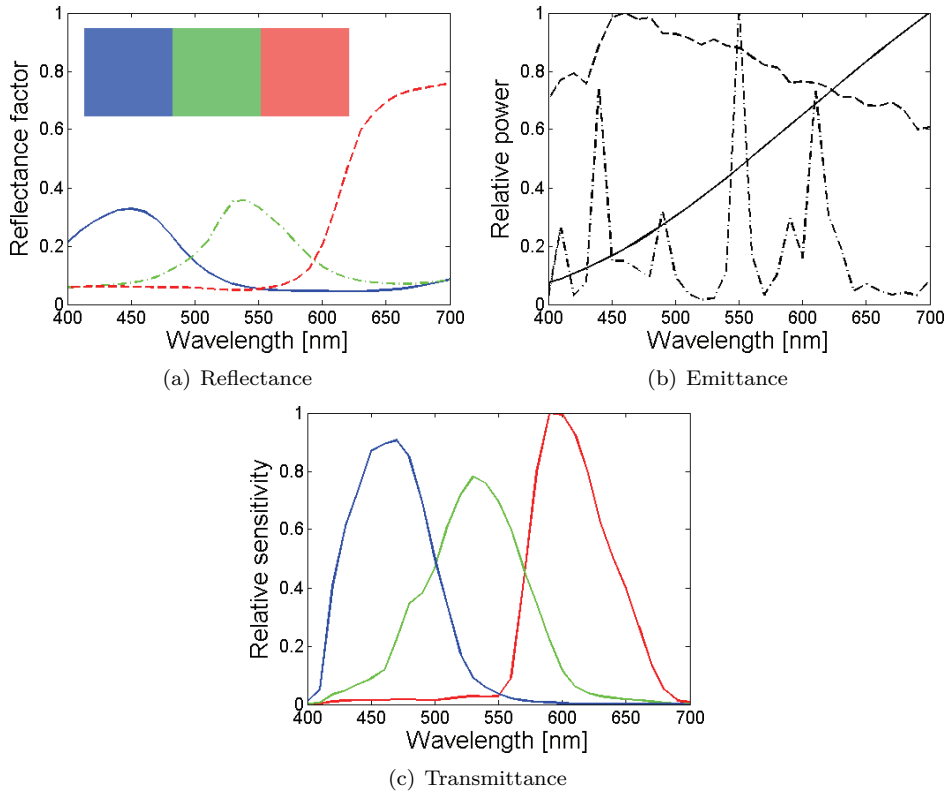
In this chapter, we review some important concepts of color imaging. Section 2.1 presents a Lambertian image formation model and Section 2.2 briefly discusses colorimetry and metamerism. Section 2.3 describes the generic workflow of a digital camera. Section 2.4 reviews several important color spaces, namely CIE XYZ, sRGB, CIELAB, and CIECAM02. Sections 2.5 to 2.7 present concepts about color correction, illuminant retrieval, and white-balancing. We review color correction transforms in Section 2.5 and linear models for illuminant and reflectance spectra in Section 2.6 and see how they can be used to reduce the dimensionality of the illuminant retrieval problem. Finally, Section 2.7 reviews a series of color constancy algorithms and explains how error in illuminant estimation can be quantified.

### 2.1 Image formation model

A color response, either captured by a camera or the human eye, depends on three factors. The *illuminant*, or surrounding light, at a spatial location  $\mathbf{x}$  is described by its *spectral power distribution* (SPD)  $E(\mathbf{x}, \lambda)$ , where  $\lambda$  is the wavelength. The *reflectance* spectrum  $S(\mathbf{x}, \lambda)$  represents the proportion of incident light reflected by an object at each wavelength between 400 and 700 nm. It characterizes the color of its surface. The *sensor sensitivities*  $R_k(\lambda)$  describe the response of a multi-channel imaging system to the incoming radiation, or *color signal*  $C(\mathbf{x}, \lambda) = S(\mathbf{x}, \lambda)E(\mathbf{x}, \lambda)$ , as a function of the wavelength (see Figure 2.1). The resulting *sensor response* or *color response*  $\rho_k(\mathbf{x})$  of a sensor  $k$  at a spatial position  $\mathbf{x}$  is given by the product of the illuminant, the reflectance, and the sensor sensitivities integrated over the visible range

$$\rho_k(\mathbf{x}) = \int_{\lambda} S(\mathbf{x}, \lambda)E(\mathbf{x}, \lambda)R_k(\lambda)d\lambda. \quad (2.1)$$

These continuous spectra can be represented by 31 discrete values sampled at 10 nm intervals over the visible range 400 – 700 nm. Smith *et al.* (1992) demonstrated that illuminant spectra can be represented using this sampling without significant loss of accuracy in terms of sensor responses, as long as they are relatively smooth, which is the case for daylight illuminants (Judd *et al.* 1964). This simplified image formation model assumes a “flat world” with



**Figure 2.1:** A color response depends on three elements: the reflection properties of the objects - the reflectance, the light, and the response of the imaging device - a camera or the eye. Plot (a) shows the reflectance spectra of the red, green, and blue patches from the MacBeth ColorChecker. Plot (b) shows three illuminant spectra, incandescent A (solid), fluorescent F11 (dash-dotted), and daylight D65 (dashed). Plot (c) shows the sensitivities of a Canon 350D camera.

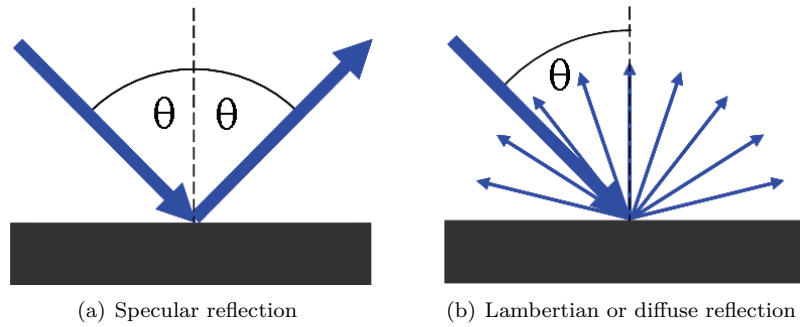
single uniform illuminants, Lambertian surfaces, i.e., reflecting light uniformly at each angle (see Figure 2.2), and no surface inter-reflections. The integral in (2.1) can be replaced by a sum and the sensor responses rewritten as

$$\rho_k(\mathbf{x}) = \mathbf{s}(\mathbf{x}, \lambda)^T \text{diag}(\mathbf{e}(\mathbf{x}, \lambda)) \mathbf{r}_k(\lambda), \quad (2.2)$$

where the vectors  $\mathbf{s}(\mathbf{x}, \lambda)$ ,  $\mathbf{e}(\mathbf{x}, \lambda)$ , and  $\mathbf{r}_k(\lambda)$  are, respectively, the sampled spectra  $S(\mathbf{x}, \lambda)$ ,  $E(\mathbf{x}, \lambda)$ , and  $R_k(\lambda)$ .  $\text{diag}(\mathbf{e}(\mathbf{x}, \lambda))$  is a  $31 \times 31$  matrix with the vector entries of  $\mathbf{e}(\mathbf{x}, \lambda)$  on its diagonal.

The sensor sensitivities  $R_k(\lambda)$  may represent any imaging system's sensitivity functions - real or modeled - such as the cone fundamentals or the CIE 1931 color matching functions (see Section 2.4.1). In this thesis,  $\rho_k(\mathbf{x})$  generally represents the sensor response of a standard RGB camera and  $k = \{R; G; B\}$  indexes the red, green, and blue channels.





**Figure 2.2:** A specular or mirror-like surface reflects an incoming ray of light, represented by the blue arrow, into a single outgoing direction, while a Lambertian surface reflects light equally at all angles.

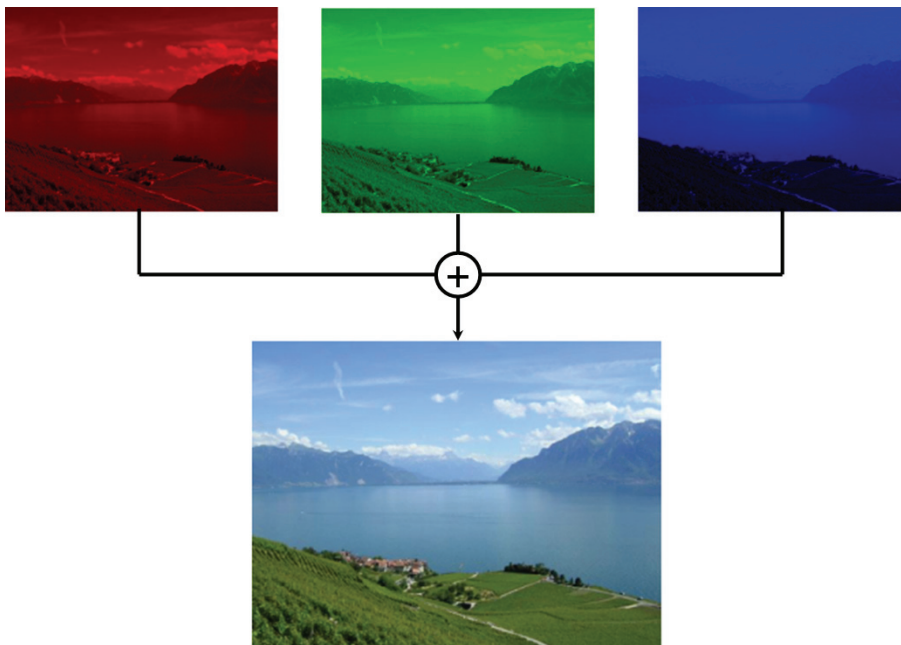
## 2.2 Trichromacy, colorimetry, and metamerism

We have seen that the color of a surface is characterized by its reflectance spectra. Once we know it, we can compute the corresponding sensor responses for any imaging device under any light. However, it is not always necessary to retrieve the full spectrum to characterize the color of an object, due to the integrative nature of the human visual system (HVS). The HVS is *trichromatic*, that is, the retina contains three types of receptors that are responsible for color vision and are sensitive to different parts of the visible spectrum (von Helmholtz 1867). Any color stimulus  $C(\lambda)$  can be matched by an additive mixture of three primaries  $C^r(\lambda)$ ,  $C^g(\lambda)$ , and  $C^b(\lambda)$  as

$$C(\lambda) = RC^r(\lambda) + GC^g(\lambda) + BC^b(\lambda), \quad (2.3)$$

where  $R$ ,  $G$ , and  $B$  are the *tristimulus values* of  $C(\lambda)$ . In other words, three values suffice to describe the color appearance of a surface under fixed lighting conditions. Consequently, the color of an object can be specified by a triplet of values under a canonical illuminant, known as the *tristimulus values*. Similarly, an illuminant can be described by the tristimulus values of a white surface  $S(\lambda) \equiv 1$ . In Chapters 3 and 4, we assess object colors from uncalibrated images not by retrieving full reflectance spectra, but as illuminant and camera independent sRGB values.

*Metamerism* is another important consequence of the integrative nature of the HVS. In daily life, it means that two objects having the same color under a given illuminant may appear different under another illuminant due to slightly different reflectance spectra. Most of us have experienced metamerism with clothing. A fabric that appears black at home under incandescent light may appear blueish or brownish in the office under fluorescent lighting. Mathematically, it means that for any tristimulus values there exists an infinity of ambiguous illuminant-reflectance combinations giving rise to them. A white surface under a reddish illuminant may produce the same color signal  $C(\lambda) = S(\lambda)E(\lambda)$  as a red surface viewed under a white light. These illuminant-reflectance pairs are named *metameric pairs*. For a more complete review of these phenomena see, for example, Ebner (2007).



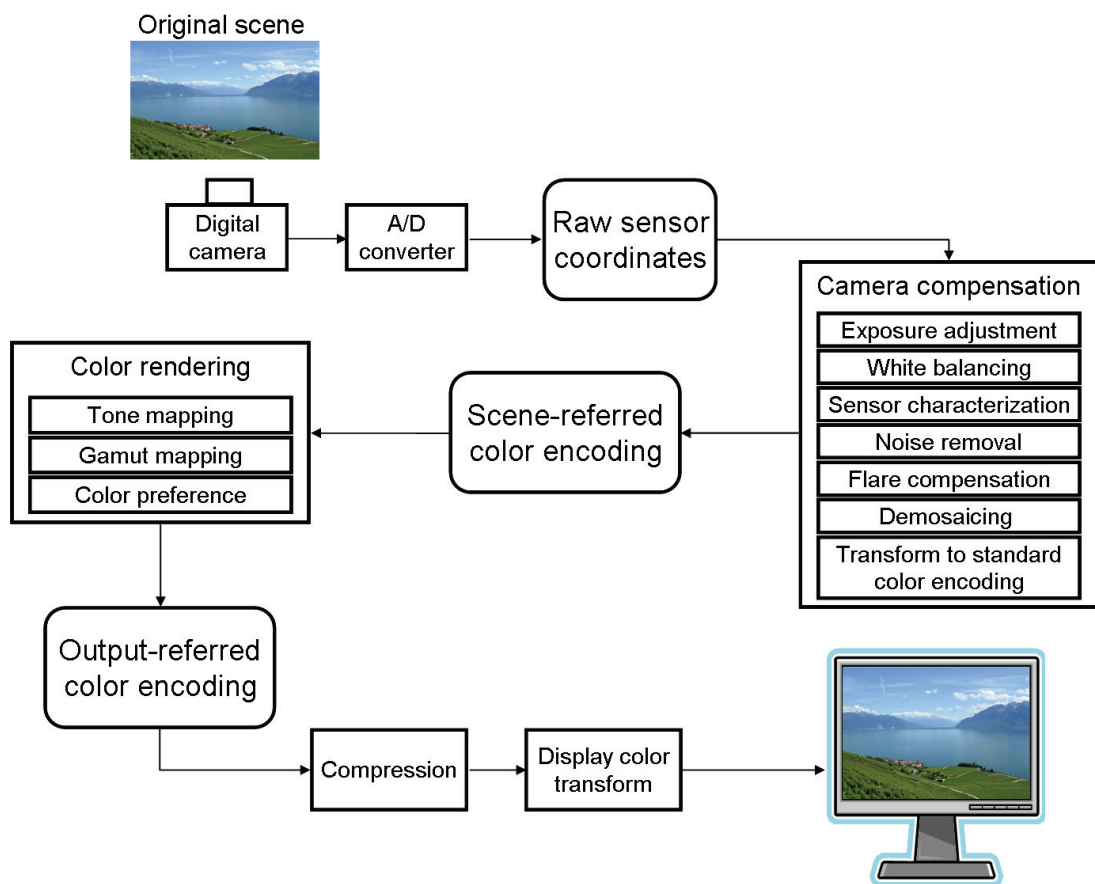
**Figure 2.3:** Any color can be matched by a mixture of red, green, and blue primaries. A monitor displays an image as a sum of a red, green, and blue components.

## 2.3 In-camera processing

Figure 2.4 shows a typical in-camera processing pipeline (ISO 2004). A scene is first captured by a sensor and stored in a raw format that is a device-dependent sensor space. The second step allows to compensate for the scene and camera characteristics. It includes white-balancing and a transform of image data into a standard color encoding. At this stage, the image is said to be in a *scene-referred* or *input-referred* state, which represents an estimate of the scene’s colorimetry. Such images are typically in a 12-bits per channel format, which maintains the scene’s original dynamic range and gamut. The image then undergoes a series of *color rendering* operations, resulting in an image that can be displayed or printed correctly. These steps depend on the *rendering intent*, which is generally to obtain a pleasing image on the desired output medium. The image is in an *output-referred* state, i.e., the image is represented by coordinates for some real or virtual output. Images are most often encoded in sRGB (Section 2.4.2), which describes images such that they can be displayed on a monitor.

## 2.4 Color spaces

A color space is a “geometric representation of colors in space, usually of three dimensions” (CIE 1987). This broad term covers many types of spaces. In additive RGB spaces, e.g., sRGB, color is expressed as a sum of three red, green,



**Figure 2.4:** Generic in-camera processing workflow. Modified from ISO (2004).

and blue primaries. In opponent color spaces, e.g.,  $YC_bC_r$ , color is represented by three components: lightness or luma - a sum of the RGB components - and two opponent components - differences of the RGB components corresponding to *red - green* and *yellow - blue*. In perceptual spaces, e.g., CIELAB, color is described by its perceptual attributes, namely hue, lightness, and chroma.

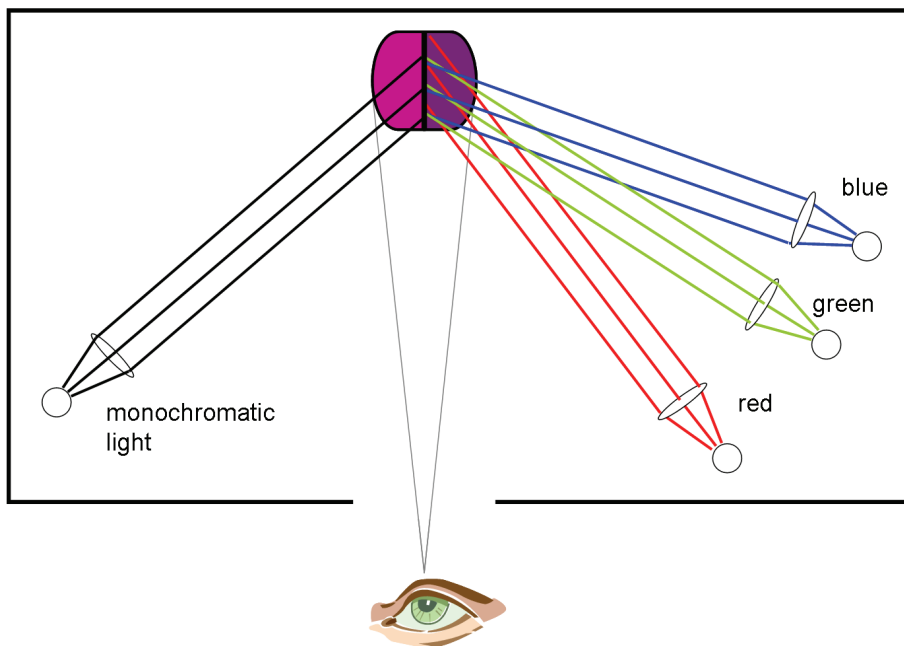
Color spaces, being accurately defined, allow to communicate color information without ambiguity. More precisely, a *colorimetric color space* is defined by three elements: a relationship between the space values and CIE colorimetry, an illuminant white-point - usually D50 or D65, and a color component transfer function (ISO 2004). This function accounts for the nonlinear response of a device or of the HVS; it is often a power-law or logarithmic function and is usually called a *gamma* function and denoted  $\gamma$ . The white-point is a set of tristimulus values specifying the reference white under a given illuminant.

A *color space encoding* is the digital encoding of a color space (ISO 2004), i.e., it includes an additional *color space range* and a digital encoding method. For example, colors specified in an RGB space can have values in  $[0, 1]$  and all values outside these bounds will be clipped. It also defines a *digital code value range*: if 8-bits per channel are used, color values will be encoded using integers in  $[0, 255]$ . A *color image encoding* is based on a color space encoding but also includes any information necessary to properly interpret the color values, such as the image state, the intended image viewing environment, and the reference medium (ISO 2004).

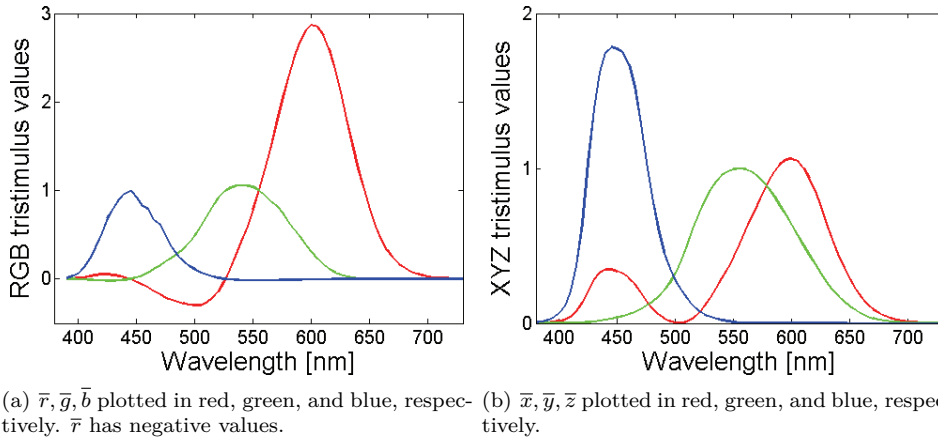
In this section, we review three color spaces: CIE XYZ, sRGB, and CIELAB. We also briefly discuss the color appearance model CIECAM02.

### 2.4.1 XYZ color space

The Commission Internationale de l'Eclairage (CIE) has defined a standard observer by *color matching experiment* (Wyszecki and Stiles 2000). Figure 2.5 illustrates the experiment principle. An observer is shown monochromatic stimuli activating a  $2^\circ$  area of the retina corresponding to the *fovea*, which has the highest cone density. The subject is asked to match each stimulus with an additive mixture (2.3) of three monochromatic primaries ( $700\text{ nm}$ ,  $563.1\text{ nm}$ , and  $435.8\text{ nm}$ ). The mixture weights are reported as a function of the monochromatic stimulus wavelength and define the color matching functions (CMF)  $\bar{r}$ ,  $\bar{g}$ , and  $\bar{b}$  shown in Figure 2.6. These  $\bar{r}$ ,  $\bar{g}$ ,  $\bar{b}$  values are then converted into the imaginary primaries  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{z}$  in order to eliminate the negative values of  $\bar{r}$  indicating that the monochromatic stimulus could not be matched without adding some red to it (CIE 1986).  $\bar{y}(\lambda)$  is forced to be equal to  $V(\lambda)$ , the eye response to luminance (Wyszecki and Stiles 2000). The *luminance* is the measure of the luminous intensity of a light source, as opposed to the *lightness*, that is, the nonlinear response of the eye to luminance. In other words, the luminance is a physical quantity, whereas lightness is a perceptual attribute. The CIE XYZ tristimulus values of a surface characterized by a reflectance spectrum  $S(\lambda)$  viewed under illuminant  $E(\lambda)$  are computed using the image formation model (2.1), where  $R_k(\lambda) = \bar{x}, \bar{y}, \bar{z}$ .  $Y$  represents the luminance of the stimulus. Tristimulus values are usually normalized such that the maximum of  $Y$  is 100. CIE XYZ are called *tristimulus values* and  $\bar{x}, \bar{y}, \bar{z}$  the *color matching functions of the CIE 1931 standard colorimetric observer* or simply the *CIE 1931 color matching functions*. They are shown in Figure 2.6.



**Figure 2.5:**  $\bar{r}, \bar{g}, \bar{b}$  color matching functions were determined by color matching experiments. The observer matches a monochromatic test light with a sum of red, green, and blue primaries by adjusting their relative intensity until it matches the test light.



**Figure 2.6:**  $\bar{r}, \bar{g}, \bar{b}$  and  $\bar{x}, \bar{y}, \bar{z}$  color matching functions.

The CIE XYZ tristimulus values are fundamental for colorimetry. Two stimuli with equal CIE XYZ values *color match*, i.e., have the same appearance when viewed under the same illuminant and under the same viewing conditions. CIE XYZ color space serves as the basis for all colorimetric color space definitions.

The intensity of XYZ can often be discarded and a stimulus conveniently represented by two dimensional *chromaticity coordinates*  $(x, y)$  computed as

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z}. \quad (2.4)$$

The normalization projects all colors onto the plane  $X + Y + Z = 1$  and the  $z$  coordinate becomes redundant.

## 2.4.2 sRGB

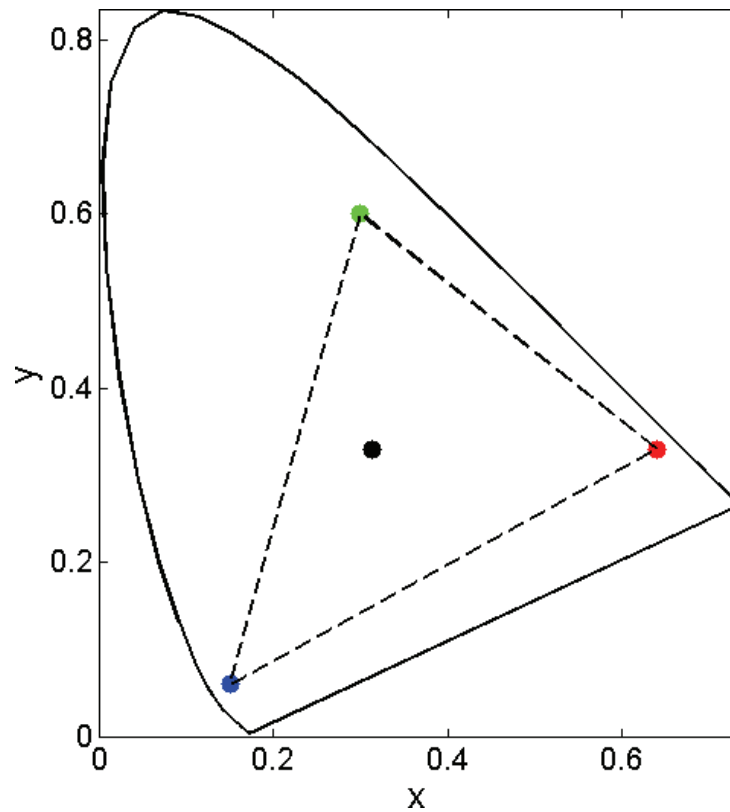
There are many RGB color spaces, each created for a particular purpose. They are defined by a set of additive RGB primaries, a white-point, and a color component transfer function (ISO 2004). The RGB sensors are a linear combination of the color matching functions  $\bar{x}, \bar{y}, \bar{z}$ .

The sRGB standard was created as a default color image encoding for the internet (Stokes *et al.* 1996, IEC 61966-2-1 1999). It is defined by ITU Rec. 709 primaries representative of a typical CRT monitor. sRGB values are defined by the following transformation of the CIE XYZ tristimulus values:

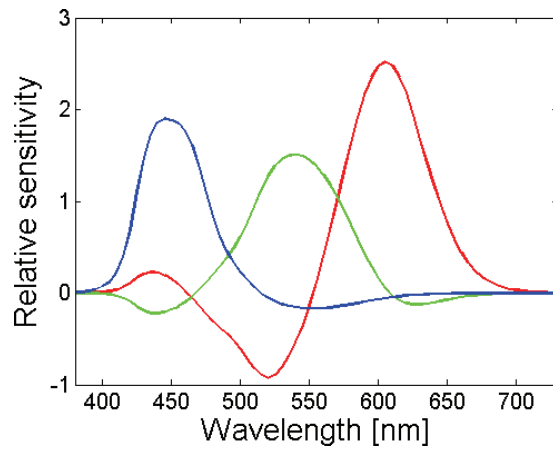
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{pmatrix} 3.2410 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (2.5)$$

Values falling outside the range  $[0, 1]$  are clipped. The color component transfer function, or gamma function, is given by

$$\gamma(x) = \begin{cases} 12.92x & x \leq 0.0031308 \\ 1.055x^{\frac{1}{2.4}} - 0.055 & x > 0.0031308 \end{cases} \quad (2.6)$$



**Figure 2.7:** The solid line shows the XYZ CMF in  $xy$  chromaticity space, i.e.,  $y(\lambda) = \bar{y}(\lambda)/(\bar{x}(\lambda) + \bar{y}(\lambda) + \bar{z}(\lambda))$  as a function of  $x(\lambda) = \bar{x}(\lambda)/(\bar{x}(\lambda) + \bar{y}(\lambda) + \bar{z}(\lambda))$ . It encloses the range of chromaticities perceptible by an average observer or, in other words, the gamut of human vision. Its boundary, named *spectral locus*, represents the monochromatic stimuli. The black dot represents the white point of D65. The red, green, and blue dots represent the sRGB primaries and the dashed triangle represents sRGB gamut, i.e., the colors that can be represented in sRGB by addition of the three primaries. The range of colors lying outside of the sRGB gamut and that cannot be represented in sRGB is quite large.



**Figure 2.8:** sRGB color matching functions.

and is applied to the RGB values after transformation (2.5). The white-point is the standard daylight illuminant D65. The transformation (2.5) from CIE XYZ to RGB, the gamma function (2.6), and the white-point define sRGB as a colorimetric color space, on which we can define an encoding method. The normalized values are converted to 8-bits per channel by multiplying the output of (2.6) by 255 and rounding it to the closest integer.

Colors in sRGB can be represented in a unit cube whose axes represent red, green, and blue. Gray values  $R = G = B$  lie on the line going from  $(0, 0, 0)$  to  $(1, 1, 1)$ , as illustrated in Figure 2.9.

sRGB is an output-referred image encoding that was designed to represent colors that can be displayed on a CRT monitor. It thus cannot encode all colors and its main disadvantage is the relatively small size of its gamut, as illustrated by Figure 2.7. sRGB remains the most common output-referred encoding: All consumer cameras still output images that are encoded in sRGB and we can assume that any digital image file is in sRGB format, unless specified otherwise.

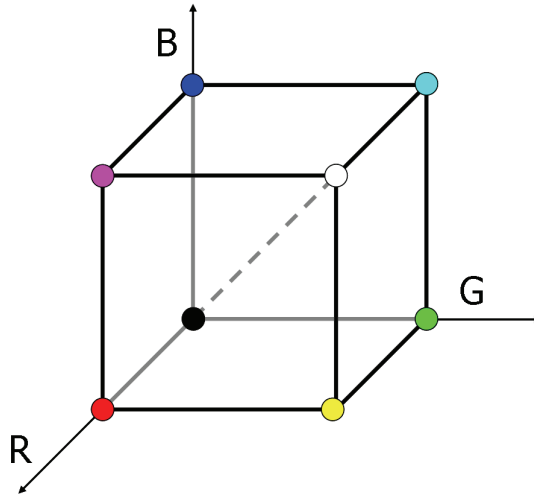
### 2.4.3 CIELAB

RGB representations of colors are adapted for additive devices, but are not intuitive descriptions.  $[250, 250, 0]$  is better conveyed by a “saturated light yellow.” Colors can be described by three perceptual attributes: hue - here “yellow,” saturation, and lightness.

The CIELAB color space (CIE 1978) was initially created to provide a single color difference measure that would replace the many color difference formulas used at that time. Its coordinates are the lightness  $L^*$  and the two opponent components  $a^* = \text{red} - \text{green}$  and  $b^* = \text{yellow} - \text{blue}$  computed from CIE XYZ tristimulus values using equations (2.7) to (2.10),

$$L^* = 116f(Y/Y_n) - 16, \quad (2.7)$$





**Figure 2.9:** Colors in sRGB can be represented in a unit cube. The encoding is additive; for example, red + green =  $R + G = (1, 0, 0) + (0, 0, 1) = (1, 0, 1) =$  magenta.

$$a^* = 500[f(X/X_n) - f(Y/Y_n)], \quad (2.8)$$

$$b^* = 500[f(Y/Y_n) - f(Z/Z_n)], \quad (2.9)$$

where

$$f(x) = \begin{cases} x^{1/3} & x > 0.008856 \\ 7.787x + 16/116 & x \leq 0.008856. \end{cases} \quad (2.10)$$

The CIE XYZ values of the stimulus are first normalized by the tristimulus values of the illuminant white-point  $X_n Y_n Z_n$ . A cube root  $f(x)$  (2.10) modeling the perceptual response to luminance is applied to this adapted stimulus. The resulting signals are combined into the three opponent responses *light – dark*, *red – green*, and *yellow – blue* channels. The multiplicative constants ensure the space’s perceptual uniformity.

*Chroma* is defined by

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}}. \quad (2.11)$$

It is a measure of what we call “saturation” in common language, but the two terms should not be confused. *Saturation is the colorfulness of a stimulus relative to its own brightness, while chroma is colorfulness relative to the brightness of a similarly illuminated area that appears white* (Fairchild 2005).

Hue is given by

$$h_{ab}^* = \tan^{-1}(b^*/a^*). \quad (2.12)$$

Hue angles can be represented on a wheel, even though the perceptual unique hues (red, green, yellow, blue) do not align with the CIELAB  $a^*b^*$  axes (Fairchild 2005). The “saturated light yellow” of our example expressed in sRGB by  $[250, 250, 0]$  is described by  $L^* = 96$ ,  $C_{ab}^* = 98$ , and  $h_{ab}^* = 102^\circ$ . Figure 2.10

represents Munsell colors (Munsell 1905) in CIELAB coordinates. Munsell colors are specified by the perceptual dimensions hue, lightness, and chroma distributed by roughly equal perceptual increments (see Appendix D). In an ideal perceptually uniform color space, colors having the same hue would lie on straight lines and colors having the same chroma would lie on circles. Their representation thus gives a geometrical estimation of the color space uniformity.

The color difference between two color stimuli 1 and 2 is given by the Euclidian distance between their CIELAB coordinates

$$\Delta E_{ab}^* = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2} = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}}. \quad (2.13)$$

$\Delta E_{ab}^* = 1$  is usually considered to represent a *just noticeable difference* (JND), i.e., to be the minimum distance between two distinguishable color stimuli. While this is a general rule of thumb, it is not always exact. This is due to CIELAB imperfect perceptual uniformity, but also to the dependency of JND on the sample size, on the viewing conditions, on the observer, etc.

CIELAB most important limitation is probably its lack of hue constancy, especially towards blue hues (Hung and Berns 1995, Moroney 2003), as illustrated by Figures 2.10(a) and 2.10(c).

#### 2.4.4 CIECAM02

CIECAM02 (Moroney *et al.* 2002) is the most recent color appearance model. It is more complex to implement than CIELAB and can predict a larger range of phenomena. In this thesis, we only use CIECAM02 as a “better CIELAB” when measuring color differences because of its improved overall perceptual uniformity. Figure 2.10 shows Munsell colors in both CIELAB and CIECAM02. Comparing the geometrical uniformity of the left and right plots of shows that CIECAM02 is more uniform than CIELAB.

Input data for the CIECAM02 includes the tristimulus values CIE XYZ of the color stimulus and of the white-point  $X_n Y_n Z_n$ , the adapting luminance  $L_A$ , and the relative luminance of the surround. It returns lightness, brightness, chroma, colorfulness, saturation, and hue.

We will need the lightness  $J$ , chroma  $C$ , hue  $h$ , and their cartesian coordinates

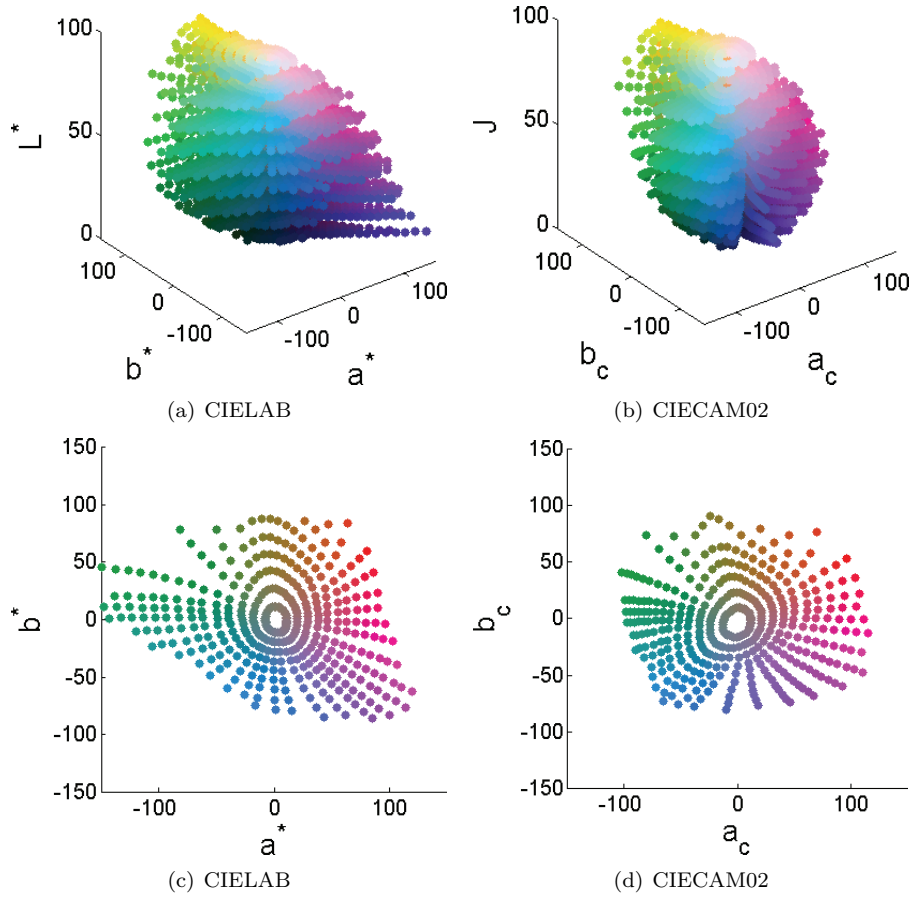
$$a_c = C \cos(h), \quad (2.14)$$

$$b_c = C \sin(h). \quad (2.15)$$

While CIECAM02 does not provide an official color difference formula, CIECAM02 coordinates  $(J, a_c, b_c)$  correlate well with CIELAB ones  $(L^*, a^*, b^*)$  and color differences can be computed as

$$\Delta E_{02}^c = \sqrt{\Delta J^2 + \Delta a_c^2 + \Delta b_c^2}. \quad (2.16)$$

The detail of the formulas is reported in Appendix A.



**Figure 2.10:** CIELAB (left) and CIECAM02 (right) are two perceptual color spaces. Plots (a) and (b) show the entire gamut of Munsell colors; plots (c) and (d) only show Munsell colors having a medium lightness ( $L^* = 50$ ). The geometrical representation of the uniformly perceptually spaced Munsell colors shows that CIECAM02 is more uniform, especially for darker blue and purple colors.

## 2.5 Color constancy and white-balancing

The HVS is said to be *color constant*, i.e., it adapts to the illuminant and sees colors roughly independently of the lighting conditions. It was modeled over a century ago by von Kries (1902). He assumed that the three types of color photoreceptors in the retina, named *cones*, adapt independently to light. There are three types of cones: L, M, and S. Each is sensitive to visible light at different wavelengths: long, medium, and short, as indicated by their name. Color constancy is achieved by individually adjusting the gains of the cone responses. Mathematically it translates into

$$L_a = k_L L \quad M_a = k_M M \quad S_a = k_S S, \quad (2.17)$$

where  $L$ ,  $M$ ,  $S$  and  $L_a$ ,  $M_a$ ,  $S_a$  represent the initial and adapted cones responses, respectively.  $k_L$ ,  $k_M$ ,  $k_S$  are the inverse cone responses to either a white ( $1/L_{white}$ ,  $1/M_{white}$ ,  $1/S_{white}$ ) or a maximum stimulus ( $1/L_{max}$ ,  $1/M_{max}$ ,  $1/S_{max}$ ). It can be written in the matrix form

$$\begin{bmatrix} L_a \\ M_a \\ S_a \end{bmatrix} = \begin{pmatrix} 1/L_{max} & 0 & 0 \\ 0 & 1/M_{max} & 0 \\ 0 & 0 & 1/S_{max} \end{pmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}. \quad (2.18)$$

This diagonal formulation of chromatic adaptation is named the *strong von Kries coefficient model* (von Kries 1902).

White-balancing in cameras follows the same formalism. It is performed early in the in-camera processing pipeline (see Section 2.3) and corresponds to the independent scaling of the  $R$ ,  $G$ ,  $B$  channels:

$$\begin{bmatrix} R_a \\ G_a \\ B_a \end{bmatrix} = \begin{pmatrix} \alpha_{11} & 0 & 0 \\ 0 & \alpha_{22} & 0 \\ 0 & 0 & \alpha_{33} \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.19)$$

The values of the color correction transform coefficients  $\alpha_{ii}$  are computed by a *color constancy algorithm* (see Section 2.7). This diagonal model holds as long as the sensors are not too correlated, which is the case for most cameras.

There are more general chromatic adaptation models, called *weak von Kries models*, in which off-diagonal terms in (2.19) are non-zeros (Brainard and Wandell 1992):

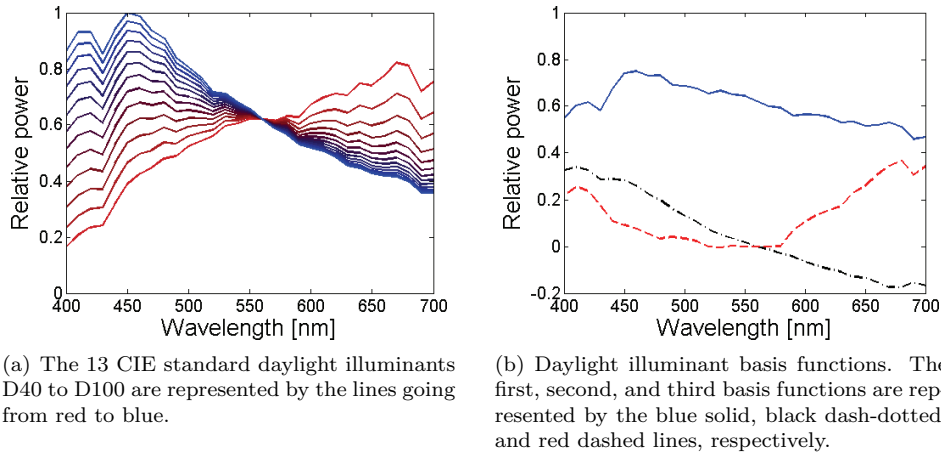
$$\begin{bmatrix} R_a \\ G_a \\ B_a \end{bmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.20)$$

It has been shown that generalized coefficients models (2.20) can be successfully applied in color constancy (Finlayson *et al.* 1994a, Finlayson *et al.* 1994b). A  $3 \times 3$  linear transform is applied to the sensors making them *sharper* - i.e., less correlated - prior to applying a diagonal transform.

Non-diagonal models can be interesting for images taken with low quality devices such as cameras integrated into cell phones, since they typically have broad-band sensors to increase the quantum efficiency, at the expense of color image quality (Baer *et al.* 1999). The off-diagonal terms  $\alpha_{ij}$  in the color transform (2.20) correspond to channel crossover. For example,  $\alpha_{21}$  represents how much of the green channel will be added to the red channel after correction.

## 2.6 Linear models for illuminants and reflectances

Color constancy algorithms can have many different outputs, illuminants or reflectances, spectra or tristimulus values. Either way, the image formation model presented in Section 2.1 must be inverted. Retrieving an illuminant spectrum  $E(\lambda)$  from (2.2) requires solving for 31 unknowns - its 31 samples - from only three equations - one for each RGB. One way of reducing the number of unknowns in the system of equations is to express the illuminant spectrum  $E(\lambda)$  and the reflectance spectrum  $S(\lambda)$  as linear sums of a limited number of



(a) The 13 CIE standard daylight illuminants D40 to D100 are represented by the lines going from red to blue. (b) Daylight illuminant basis functions. The first, second, and third basis functions are represented by the blue solid, black dash-dotted, and red dashed lines, respectively.

**Figure 2.11:** Plot (a) shows the CIE standard daylight illuminants. Plot (b) shows the three first basis vectors characteristic of daylight illuminants.

basis functions. Determining spectra reduces to determining the weights called illuminant and reflectance *descriptors*.

Most natural and man-made surfaces reflectance spectra are smooth enough to be represented by a linear combination of a low number of basis functions. Cohen (1964) demonstrated that Munsell surface reflectances could be approximated by the average reflectance plus two additional components, i.e., by a sum of three basis functions. However, later work showed that five to seven (Maloney 1986) or even eight (Parkkinen *et al.* 1989) basis functions were necessary to accurately represent real surface spectra.

These basis functions can be computed by principal component analysis (PCA) on a large set of reflectance spectra (Maloney 1986, Marimont and Wandell 1992, Westland *et al.* 2000). Reflectance spectra can be written as

$$S(\lambda) = \sum_{i=1}^{N_\sigma} \sigma_i \mathcal{S}_i(\lambda), \quad (2.21)$$

where  $\sigma_i$  are the weights or reflectance descriptors,  $\mathcal{S}_i(\lambda)$  the surface reflectance basis functions, and  $N_\sigma$  the number of basis functions.

Similarly, Judd *et al.* (1964) decomposed daylight illuminant spectra as sums of three basis functions  $\mathcal{E}_i$ , i.e.,

$$E(\lambda) = \sum_{i=1}^{N_\epsilon} \varepsilon_i \mathcal{E}_i(\lambda), \quad (2.22)$$

where  $\varepsilon_i$  are the illuminant descriptors,  $\mathcal{E}_1(\lambda)$  is the average daylight, and  $N_\epsilon = 3$  is the number of basis functions. The authors computed the illuminant basis functions by principal component analysis on a total of 622 spectra from three sets of daylight measurements. Using only three basis functions, i.e., the average daylight  $\mathcal{E}_1$  plus two orthogonal characteristic vectors, they

could reconstruct the illuminant spectra dataset with good accuracy, measured as the variance between measured and reconstructed spectra. They also proposed a method to compute the spectra of typical phases of daylight illuminants (CIE 2004) using these three basis functions. These *standard illuminants* are designated by the letter “D” followed by a number corresponding to the illuminant *correlated color temperature*. That is, the temperature to which an ideal black body radiator needs to be heated to match a certain light source color. For instance, D65 specifies a daylight illuminant with a correlated color temperature of 6500K. Figure 2.11 shows the standard daylight illuminant phases D40 to D100 and the three basis functions  $\mathcal{E}_1(\lambda)$ ,  $\mathcal{E}_2(\lambda)$ , and  $\mathcal{E}_3(\lambda)$  obtained by PCA.

Using this formalism, the image formation model (2.2) can be rewritten as

$$\boldsymbol{\rho} = \mathbf{R} \text{diag}(\mathbf{B}_e \boldsymbol{\varepsilon}) \mathbf{B}_s \boldsymbol{\sigma} = \boldsymbol{\Lambda}_\varepsilon \boldsymbol{\sigma}, \quad (2.23)$$

where  $\mathbf{R}$  is a  $3 \times 31$  matrix containing the sensor sensitivities. The matrix  $\mathbf{B}_e$  contains the  $N_\varepsilon$  illuminant basis functions and the matrix  $\mathbf{B}_s$  contains the  $N_\sigma$  reflectance basis functions.  $\boldsymbol{\varepsilon}$  and  $\boldsymbol{\sigma}$  contain the illuminant and reflectance descriptors  $\varepsilon_i$  and  $\sigma_j$ , respectively. The matrix  $\boldsymbol{\Lambda}_\varepsilon$  is called the *lighting matrix*. We will see this formulation in detail in Chapter 5.

## 2.7 Color constancy algorithms

Color constancy algorithms aim at retrieving illuminant descriptors or illuminant independent reflectance descriptors from RGB images. The dimension of this under-determined problem can be reduced by introducing assumptions on the image or scene content. The algorithms presented in this section are classified in two categories: *Single* light methods exploit the content of one scene or image to retrieve illuminant information, whereas *multiple* lights methods retrieve illuminant descriptors from a set of two, or more, images containing redundant information.

### 2.7.1 Single light methods

The illuminant can be estimated by introducing simple assumptions on the scene content. The *max-RGB* (Land and McCann 1971) algorithm relies on the presence of a white patch in the scene to retrieve illuminant descriptors. More precisely, it assumes the scene white-point to correspond to the maximum sensor response in each channel of an image

$$\boldsymbol{\rho}^E \mapsto [\max(\rho_R), \max(\rho_G), \max(\rho_B)], \quad (2.24)$$

where the three-dimensional vector  $\boldsymbol{\rho}^E$  designates the RGB values of a white surface, i.e., having a reflectance spectra  $S(\lambda) \equiv 1$ , and the symbol  $\mapsto$  means “is derived from.” This simple algorithm performs well if a “white-like” patch is present in the scene, but it may return erroneous results if one, or more, color channel is saturated or if no white patch is present. The *gray world* algorithm estimates the illuminant by assuming that the average of an image is gray, i.e., that the average reflectance of the objects present in the scene is constant (Buchsbbaum 1980). The scene white-point is estimated by calculating

the average value of each color channel

$$\boldsymbol{\rho}^E \mapsto [\bar{\rho}_R, \bar{\rho}_G, \bar{\rho}_B]. \quad (2.25)$$

The algorithm performs poorly if the variety of colors in the image is too low or if the influence of the larger surfaces bias the result. The latter can be prevented using a *weighted gray world* approach, where the importance of each surface is independent of its size. This can be done by segmenting the image prior to taking the image segments average (Gershon *et al.* 1988) or by averaging the image histogram. Gershon *et al.* (1988) also modified Buchsbaum’s (1980) method by adding information about the properties of surfaces likely to appear in a scene. The scene illuminant is found by mapping scene color values onto an *ideal material space* formed by the descriptors of “all” surface reflectances. Variations of the gray world algorithm include the *grey edge hypothesis* (van de Weijer and Gevers 2005) and *shades of gray* (Finlayson and Trezzi 2004).

Rather than retrieving a white-point, some algorithms retrieve spectral quantities. The dimension of the ill-posed illuminant retrieval problem must then be reduced, for example by expressing illuminant and reflectance spectra by linear models. In this framework, Maloney and Wandell (1986) assume the number of sensors to be superior to the number of degrees of freedom of the reflectance descriptors. They retrieve the illuminant by inverting a linear model (see Section 2.6). If we apply this approach to the case of an RGB camera, we have three sensors and can thus approximate the scene reflectances by a sum of only two functions, which is not enough to accurately represent reflectance spectra. This method can thus not be applied in practice.

A series of algorithms resulted from the observation that not all colors can arise under any illuminant. For example, a bright blue cannot be seen under an incandescent light. Formally, this is because the values of a color signal  $C(\lambda) = S(\lambda)E(\lambda)$  for all  $S(\lambda) \in [0, 1]$  are limited by the illuminant spectral power distribution  $E(\lambda)$ . Forsyth (1990) follows this idea and shows that color gamuts - the sets of RGB values that can be observed under a given light - are convex and bounded. The color gamut of an illuminant can be built by calculating the RGB responses resulting from “all” surfaces observed under that illuminant. The algorithm consists of two steps. First, it computes the set of possible solutions composed of the lights whose gamuts contain all the image RGB values. The best color correction transform is then chosen as the transform leading to the largest corrected gamut volume. Variations of this method take the mean (Barnard *et al.* 2000) or the median (Finlayson and Hordley 2000) over the set of possible solutions. Finlayson (1996) modified Forsyth’s (1990) approach by working in a two-dimensional chromaticity space and excluding illuminants that are unlikely to occur in real scenes, such as purple lights. *Color by correlation* (Finlayson *et al.* 2001a) presents a similar approach: The maximum likelihood of an illuminant occurrence is computed using the correlation between the image chromaticities and the chromaticities expected for each test illuminant. Gamut mapping approaches in general and *color by correlation* in particular perform better than simpler methods such as *max-RGB* or *gray world* (Barnard *et al.* 2002a, Hordley and Finlayson 2004). To a lesser extent, such methods may also fail if the range of colors present in the image is not large enough, in which case the RGB gamut derived from the image will not be representative of the gamut of the actual illuminant.

### 2.7.2 Multiple lights methods

There are other color constancy approaches that consider *multiple* lights or images. Two images taken under different illuminants or several surfaces viewed under different lights provide redundant information that can be exploited to solve for color constancy. The *flash/no flash* method takes two images of one scene captured with and without flash. Knowing the flash properties, information on the illuminant can be retrieved. DiCarlo *et al.* (2001) take an image pair to compute an estimate of the scene lit by the flash only and, knowing the flash spectral power distribution, use this last image to retrieve objects reflectance functions. They finally employ these reflectances to estimate the illuminant in the image taken without flash. Lu and Drew (2006) present a variation of the method.

The *chromagenic* approach takes two images of one scene captured with and without a colored filter placed in front of the camera (Finlayson *et al.* 2006). The filter is chosen such that the relationship between the filtered and unfiltered RGB values depends strongly on the illumination. The knowledge of its transmittance is used to solve for color constancy. The algorithm is tested on a set of precomputed illuminants. More precisely, transform matrices mapping unfiltered RGB onto filtered RGB are computed in a preprocessing step. The estimated illuminant minimizes

$$\| \mathcal{T}_i \mathcal{Q} - \mathcal{Q}^F \|, \quad (2.26)$$

where  $\mathcal{Q}^F$  and  $\mathcal{Q}$  represent the filtered and unfiltered RGB, respectively, and  $i$  runs over the test illuminants.  $\mathcal{T}_i = \mathbf{\Lambda}_i^F \mathbf{\Lambda}_i^{-1}$ , where  $\mathbf{\Lambda}_i^F$  and  $\mathbf{\Lambda}_i$  represent the filtered and unfiltered illuminants lighting matrices.

Finlayson (1994) presents a method to solve for color constancy taking two sets of three, or more, surfaces viewed under two different illuminants. The illuminant and reflectance spectra are modeled by tri-dimensional linear models. He considers the transforms  $\mathcal{M}^{1,2}$  mapping the sensor responses under the first illuminant to the ones under the second illuminant. He shows that the illuminant descriptors  $\boldsymbol{\varepsilon}^1 = (\varepsilon_1^1, \varepsilon_2^1, \varepsilon_3^1)$  and  $\boldsymbol{\varepsilon}^2 = (\varepsilon_1^2, \varepsilon_2^2, \varepsilon_3^2)$  can be found by solving

$$\mathcal{Q}\boldsymbol{\varepsilon}^1 = \mathcal{P}\boldsymbol{\varepsilon}^2, \quad (2.27)$$

where  $\mathcal{P}$  is a  $9 \times 3$  matrix containing the stretched out  $3 \times 3$  basis lighting matrices  $\mathbf{\Lambda}^i$ ,  $i = 1, 2, 3$ . The three columns of  $\mathcal{Q}$  are the matrices  $\mathcal{M}^{1,2} \mathbf{\Lambda}^i$  in the form of  $9 \times 1$  vectors. The solution of (2.27) is the intersection of the two tri-dimensional spaces spanned by the columns of  $\mathcal{P}$  and  $\mathcal{Q}$  and is found by the method of the principal angles, which chooses  $\boldsymbol{\varepsilon}^1$  and  $\boldsymbol{\varepsilon}^2$  such that the angle between  $\mathcal{Q}\boldsymbol{\varepsilon}^1$  and  $\mathcal{P}\boldsymbol{\varepsilon}^2$  is minimized.

In (Finlayson *et al.* 1995), the authors note that all the possible transforms mapping the chromaticity coordinates of a surface viewed under a unknown illuminant to a canonical one roughly lie on a straight line and that these transforms for different unknown illuminants lie on different lines. By interesting them, they can recover illuminant independent chromaticities for the surface. Kawakami *et al.* (2004) present a variation the method that is more robust to noise.

D'Zmura and Iverson (1993a, 1993b, 1994) present a solution for illuminant and reflectance retrieval when  $N_p$  surfaces are viewed under  $N$  illuminants.



Illuminant and reflectance spectra are expressed as sums of a small number of basis functions and the image formation model is expressed as (2.23). This form is called *bilinear*: if the illuminant descriptors  $\varepsilon_i$  are fixed, the problem is linear in the reflectance descriptors  $\sigma_j$  and inversely. The authors write (2.23) for all the surfaces and illuminants and combine the resulting equations in the compact form

$$\mathbf{F}\tilde{\boldsymbol{\sigma}} = 0, \quad (2.28)$$

where  $\tilde{\boldsymbol{\sigma}}$  is built using exclusively the reflectance descriptors  $\sigma_j$  and  $\mathbf{F}$  depends on the illuminant descriptors  $\varepsilon_i$ , the illuminant and reflectance basis functions  $\mathcal{E}_i(\lambda)$  and  $\mathcal{S}_j(\lambda)$ , the sensors responses  $R_k(\lambda)$ , and the color responses  $\boldsymbol{\rho}$ . More precisely,  $\mathbf{F}$  is of the form

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & -\mathbf{F}_2 & 0 & \dots & 0 \\ \mathbf{F}_1 & 0 & -\mathbf{F}_3 & & 0 \\ \vdots & & & \ddots & \\ \mathbf{F}_1 & 0 & & & -\mathbf{F}_{N_\sigma} \end{pmatrix}, \quad (2.29)$$

where the  $N_\sigma$  matrices  $\mathbf{F}_j = \mathbf{C}_j\mathbf{D}$ .  $\mathbf{C}_j$  are block-diagonal matrices whose blocks  $\mathbf{B}_j$  are built using the sensor sensitivities  $R_k(\lambda)$ , the illuminant basis functions  $\mathcal{E}_i(\lambda)$ , and the reflectance basis functions  $\mathcal{S}_j(\lambda)$  as

$$(\mathbf{B}_j)_{ki} = \int_{\lambda} R_k(\lambda)\mathcal{E}_i(\lambda)\mathcal{S}_j(\lambda)d\lambda, \quad j = 1, \dots, N_\sigma. \quad (2.30)$$

The matrix  $\mathbf{D}$  contains the sensor responses  $\boldsymbol{\rho}$  of the  $N_p$  surfaces viewed under the  $N$  illuminants. The details of the derivation can be found in Appendix B. The first step of the method is to solve (2.28) by singular value decomposition and retrieve the reflectance descriptors  $\sigma_j$  from  $\tilde{\boldsymbol{\sigma}}$ . At this stage, the reflectances are known and the illuminant descriptors can be retrieved in a second step. Indeed, the problem is now reduced to a simple linear system in  $\varepsilon_i$  and we have for each surface  $n_p$  and each illuminant  $n$

$$\boldsymbol{\rho}_{n_p,n} = \sum_{j=1}^{N_\sigma} \sigma_{n_p,j} \mathbf{B}_j \boldsymbol{\varepsilon}_n, \quad (2.31)$$

or, equivalently,

$$\boldsymbol{\varepsilon}_n = \left[ \sum_{j=1}^{N_\sigma} \sigma_{n_p,j} \mathbf{B}_j \right]^{-1} \boldsymbol{\rho}_{n_p,n}. \quad (2.32)$$

This formulation assumes that the  $\mathbf{B}_j$  are square, i.e., that the number of reflectance basis functions  $N_\sigma$  is equal to the number of channels  $N_\rho$  of the imaging device. It can be extended to the cases where  $N_\sigma \leq N_\rho$  by adding entries to  $\mathbf{B}_j$  to make it square. The supplementary entries must be linearly dependent to the existing columns. The case  $N_\sigma > N_\rho$  is not generally solvable. D'Zmura and Iverson (1994) report the conditions on the number of color channels  $N_\rho$ , the number of surfaces  $N_p$ , the number of illuminant descriptors  $N_\varepsilon$ , the number of reflectance descriptors  $N_\sigma > N_\rho$ , and the number of illuminants  $N$  under which the illuminants and reflectances can be simultaneously retrieved. The conditions are of the form  $N_p = N_\varepsilon = N_\sigma = N > N_\rho$ . The

authors also suggest that conditions of the form  $N = N_p > N_\varepsilon = N_\sigma > N_\rho$  may lead to perfect recovery. These conditions on the number of parameters for the problem to be solvable are restrictive.

In Chapter 5, we present a method allowing to retrieve illuminant spectra using a small number of redundant reflectances present in a set of  $N$  images. Using the assumption that the reflectance of these surfaces, while unknown, remains equal in all the images, we can deduce the illuminant in each image in the set. This approach can be classified as a *multiple lights* method.

The main difference between our method and approaches using pairs of redundant images such as the “flash/no flash” or “chromagenic” approaches is that we only use the redundancy of reflectances rather than information on the difference between two images (the flash spectral power distribution in the “flash/no flash” approach or the colored filter transmittance in the chromagenic approach). Another difference is the number of input images required to estimate the illuminants: these methods, when considering uniform illuminants, retrieve one illuminant using two images, an original and a modified image; whereas our method allows retrieving  $N$  illuminants in  $N$  input images. Also, our input images do not need to represent the same scene, but only to contain several objects present in every scene.

The approach described in Chapter 5 presents most similarities with D’Zmura and Iverson’s (1993a, 1993b, 1994) approach. By combining the information on all surfaces viewed under all illuminants into one system of linear equations, they first recover the reflectance descriptors by inverting this linear system and then use the resulting descriptors to compute the illuminants. In our approach, we solve in parallel several linear systems, one for each individual surface viewed under each test illuminant. We force the resulting reflectance descriptors to be equal, which allows us to deduce the illuminants.

### 2.7.3 Illuminant estimation error measure

The error in white-point estimation is often computed as the angle between the vectors  $\rho^E$  and  $\hat{\rho}^E$  formed by the real and retrieved white-points and is called the *angular error* (Hordley and Finlayson 2004, Barnard *et al.* 2002a, Barnard *et al.* 2002b)

$$e_{ang} = \arccos\left(\frac{\rho^E \circ \hat{\rho}^E}{\|\rho^E\| \|\hat{\rho}^E\|}\right). \quad (2.33)$$

The space in which the angular error is computed is often not specified. We can assume that it is either the camera RGB or, in case the camera sensor responses are not accessible, CIE XYZ. This measure has the advantage of being intensity independent, but RGB or even CIE XYZ spaces are not perceptually uniform and two different real and retrieved illuminant pairs  $(\rho_1^E, \hat{\rho}_1^E)$  and  $(\rho_2^E, \hat{\rho}_2^E)$  may give the same angular error (2.33), while corresponding to perceptually different ones. Also, the angular error is space dependent.

The illuminant estimation error can also be computed in a perceptual space, such as CIELAB. As opposed to the angular error (2.33),  $\Delta E_{ab}^*$  is perceptually meaningful, but requires more computation and is not intensity independent. Hordley (2006) computed an approximate equivalence between both measures. For example, an angular error of about  $3^\circ$  in XYZ corresponds to an error in CIELAB of about  $\Delta E_{ab}^* = 6.8$ . An error  $\Delta E_{ab}^* \leq 6$  seems to be acceptable

---

for real images (Hordley 2006). Moreover, an error in white-balancing will be more or less noticeable depending on the image content and on the direction of the color shift (Nikkanen *et al.* 2008), aspects that cannot be accounted by a simple illuminant error measure.

Barnard *et al.* (2002a) compute the angular error on a set of test images with a root mean square (RMS) estimation. Hordley and Finlayson (2004) argue that the RMS is not adapted to illuminant estimation error, whose underlying distribution is not normal, and suggest to use the median error instead.

The algorithms presented in this section assume a “flat world” as presented in Section 2.1 on the image formation model. Real scenes contain multiple illuminants, shadows, and specularities. Real surfaces are not Lambertian and create inter-reflections, i.e., the light reflected from an object strikes other surrounding objects, illuminating them with a colored light that is the combination of the incident light and object’s reflectance. These factors may reduce the performances of the color constancy algorithms.

## 2.8 Summary

We reviewed that captured color depends on the product of three main factors, namely the illuminant spectral power distribution, the object’s reflectance properties, and the imaging device response to incoming light, integrated over the visible spectrum (Section 2.1). As a consequence of the integrative nature of image formation, a color stimulus can be specified by tristimulus values under a canonical illuminant, i.e., a point in a three-dimensional color space such as the additive sRGB or the perceptual CIELAB (Section 2.4). We also presented the generic workflow of in-camera processing (Section 2.3) and focused on color correction and illuminant compensation (Section 2.5). White-balancing is generally performed using  $3 \times 3$  diagonal linear transforms whose coefficients are computed by a color constancy algorithm (Section 2.7). More precisely, we have seen that the retrieval of reflectance or illuminant spectra from triplets of RGB values is an ill-posed problem that cannot be solved without additional information or assumption on image content. For example, expressing spectra as a sum of a small number of basis functions allows reducing the dimension of the illuminant retrieval problem (Section 2.6). Finally, we have seen that the error in illuminant estimation can be quantified as the angle between the vectors representing the real and retrieved white-points, respectively (Section 2.7.3).



## Chapter 3

# Skin color assessment from uncalibrated images

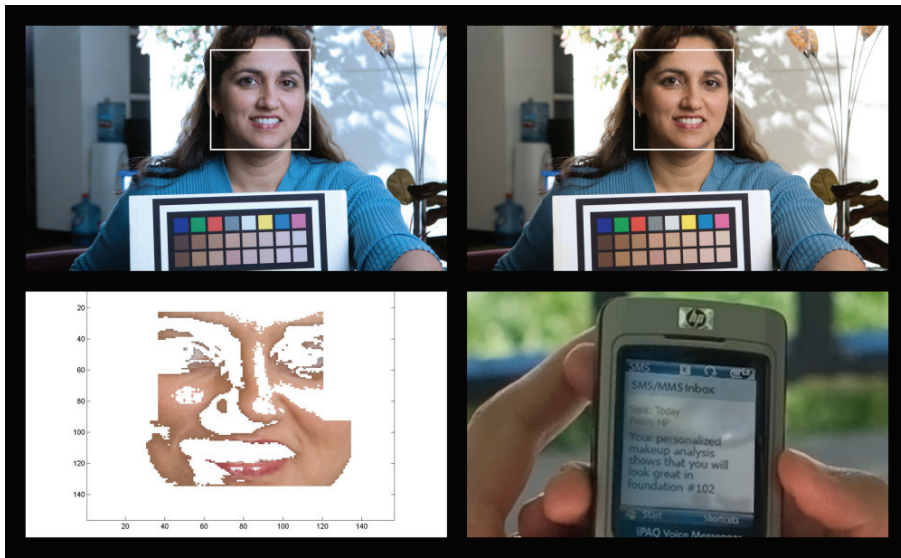
### 3.1 Introduction

We present an application designed to provide women with cosmetic advice through their mobile phones (Green 2007) and focus on the color correction method targeted towards skin tones. The consumer takes a self-portrait and sends the resulting picture via multimedia messaging service (MMS) to an advisory service, which returns a makeup recommendation. The image can be captured in any environment; the system is designed to work equally well under fluorescent, daylight, or incandescent lights. Our algorithm color corrects the image, extracts the face pixels, compares them with an existing database associating makeup recommendations with skin tones, and returns a foundation shade suggestion in a short text message (SMS). Figure 3.1 illustrates the system.

A foundation is a tinted cream, or liquid, used as a base for facial makeup. It is the second most popular product in the annual \$160 billion worldwide market of cosmetics. Despite the popularity of this product, studies showed that choosing the right foundation is a difficult task and that about 94% of women were wearing the wrong shade (Jain *et al.* 2008). The system does not simply return the foundation matching the skin in color but uses the color corrected face image to replicate the opinion of an expert stored in a database.

For the system to work properly, an accurate color correction of skin tones is necessary. As we have seen in Chapter 2, the colors in an image depend on the camera characteristics, object properties, and lighting conditions. Different cameras will most likely return different pixel values when capturing the same scene in the exact same conditions. The in-camera processing varies among manufacturers and camera models resulting in important variations across images. As we are working with images that are already rendered, we cannot access raw data or bypass the in-camera processing. Moreover, the growth of low quality devices such as cell phone integrated cameras results in larger rendering variations.

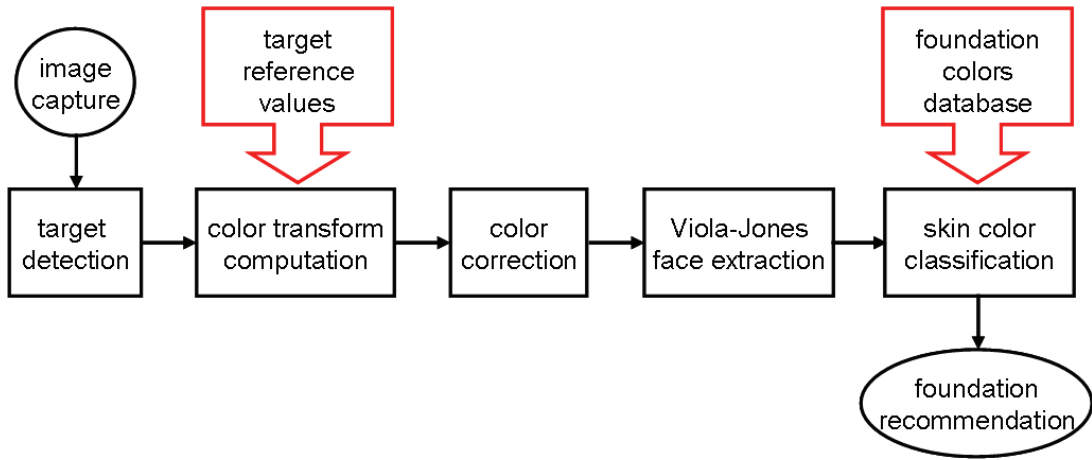
The color of a surface is fully characterized by its reflectance spectrum, which can be measured by spectrometry, but which can also be specified by tris-



**Figure 3.1:** The user sends an image of herself via MMS. The system color corrects the image, extracts face values, compares them to a database built with the help of a makeup expert and returns foundation color advice in the form of a short text message.

stimulus values (Section 2.2) measured by a colorimeter whose response mimics the CIE 1931 color matching functions  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{z}$  (CIE 2004). However, our application requires to assess color from a standard RGB image and such devices are thus not adapted. Still, standard RGB cameras can be used as colorimeters (Farrell *et al.* 1994, Wu *et al.* 2000), provided that they have been previously calibrated and are employed under controlled illumination conditions. Nevertheless, it requires access to the raw data of the camera. Such approaches are too restrictive for the present application, as the color correction must be fast. Also, the consumer must be able to use her own camera under any lighting conditions without requiring specific equipment.

Because our application only requires the assessment of the color of skin tones, images can be corrected using a limited number of reference colors present in the scene in the form of a color calibration target. This is a simple and inexpensive alternative to the use of calibrated devices. The only requirement is that the image contains, in addition to the subject's face, the calibration target designed specifically for the correction of skin tones. The target patches are automatically detected and their values compared with reference ones. A color correction matrix is computed and applied to the entire image, prior to face pixels extraction. Figure 3.2 shows the system's pipeline. The experiment is run on photographs of faces taken with uncalibrated cameras under uncontrolled light conditions. The images were captured with consumer cameras employed in fully automatic mode, the images are thus already rendered.



**Figure 3.2:** Overview of the image processing pipeline for skin color classification.

## 3.2 Background

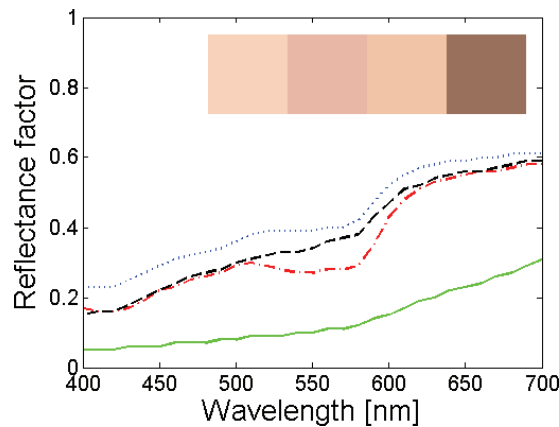
### 3.2.1 Cameras as colorimeters

Retrieving reflectances from camera responses is not trivial. For many applications, however, it is sufficient to retrieve colorimetric values rather than the full reflectance spectrum. The human visual system is indeed unable to recover spectral information (Section 2.2). A three channel imaging device is thus sufficient to retrieve tristimulus values under known illuminant conditions. Several approaches using standard RGB cameras as colorimeters have been proposed. Farrell *et al.* (1994) present a method to turn a scanner into a colorimeter. Wu *et al.* (2000) use an RGB camera to compute calibration matrices mapping camera RGB to CIE XYZ. Color transforms are computed by minimizing a cost function in CIELAB space or the mean square error in CIE XYZ color space under several selected illuminant conditions. The application was the colorimetry of human teeth. Hubel *et al.* (1997) present a method to compute  $3 \times 3$  color transform matrices intended for camera calibration in digital photography by simple least squares regression, white-point preserving least squares regression, and weighted white-point preserving least squares regression.

Camera calibration is generally performed using a standard calibration target, such as the widely used MacBeth ColorChecker (Barnard and Funt 2002). The transform is usually applied prior to the image rendering implemented in the camera. Such *calibration* methods require the access to the raw data of the sensor. We present a *post-calibration* method, i.e., we apply a color transform on rendered images captured under unknown illuminants with unknown cameras. Cai (2002) shows a similar approach to determine tongue color from images containing a MacBeth ColorChecker.

### 3.2.2 Skin color measurement

Skin appearance has been studied in computer graphics (for skin rendering), computer vision (for detection and tracking of faces), medicine (for diagnostic



**Figure 3.3:** Example of skin spectra. The four colored patches correspond, from left to right, to examples of North Asian (blue dotted), Caucasian (red dash-dotted), South Asian (dashed), and Africanoid (green solid) skin spectra.

purposes), and cosmetology (for makeup and skin care). In particular, it has been shown that, across the range of possible human skin tones, lightness varies more than chromaticity (Störting *et al.* 2001). The relative constancy of the skin chromaticity is often used as a cue in face detection algorithms (Hsu *et al.* 2002).

Skin color is usually measured with reflectance spectrometry (for a review see Igarashi *et al.* 2005). Figure 3.3 shows examples of typical skin spectra. The perceived color of skin depends on its pigmentation, blood microcirculation, roughness, sebum, and perspiration (Barel *et al.* 2001). The multiple layers of the skin make accurate color measurement difficult as its reflectance properties are not Lambertian (Igarashi *et al.* 2005). In other words, the perceived color and brightness of the skin depend upon its viewing angle. Even though they are not optimal for skin measurement, traditional spectrometers are inexpensive and simple to use and hence still widely employed for skin colorimetry (Clarys *et al.* 2000).

Narrow band spectrometers were developed specifically for skin measurement after observing that the color of skin has two main components, which selectively absorb wavelengths, melanin and hemoglobin, located in two different layers of the skin (Taylor *et al.* 2006). The use of spectrometers has significant drawbacks: the area measured is about  $0.05 \text{ cm}^2$  while the skin is not homogeneous (Barel *et al.* 2001). Indeed, uneven tan, blemish, rosacea, and shine are responsible for skin unevenness. Furthermore, the pressure of the probe on the skin can be an important source of bias (Piérard 1998). Caisey *et al.* (2006) developed a proprietary device composed of an integrating sphere, a spectrometer, and a 3-CCD camera allowing non-contact spectroscopy of different parts of the face and simultaneous imaging for estimation of the skin unevenness.

The melanin and hemoglobin components have been modeled as independent and extracted using in-vivo measurements (Nakai *et al.* 1998). Tsumura *et al.* (1999) applied the independent component analysis method to the pixel



---

values of color images of faces. They extracted two independent color components without any *a priori* assumption on skin properties. The two components are identified *a posteriori* with melanin and hemoglobin. The results were used for skin color reproduction and rendering and are thus not applicable to our problem.

### 3.3 Our approach

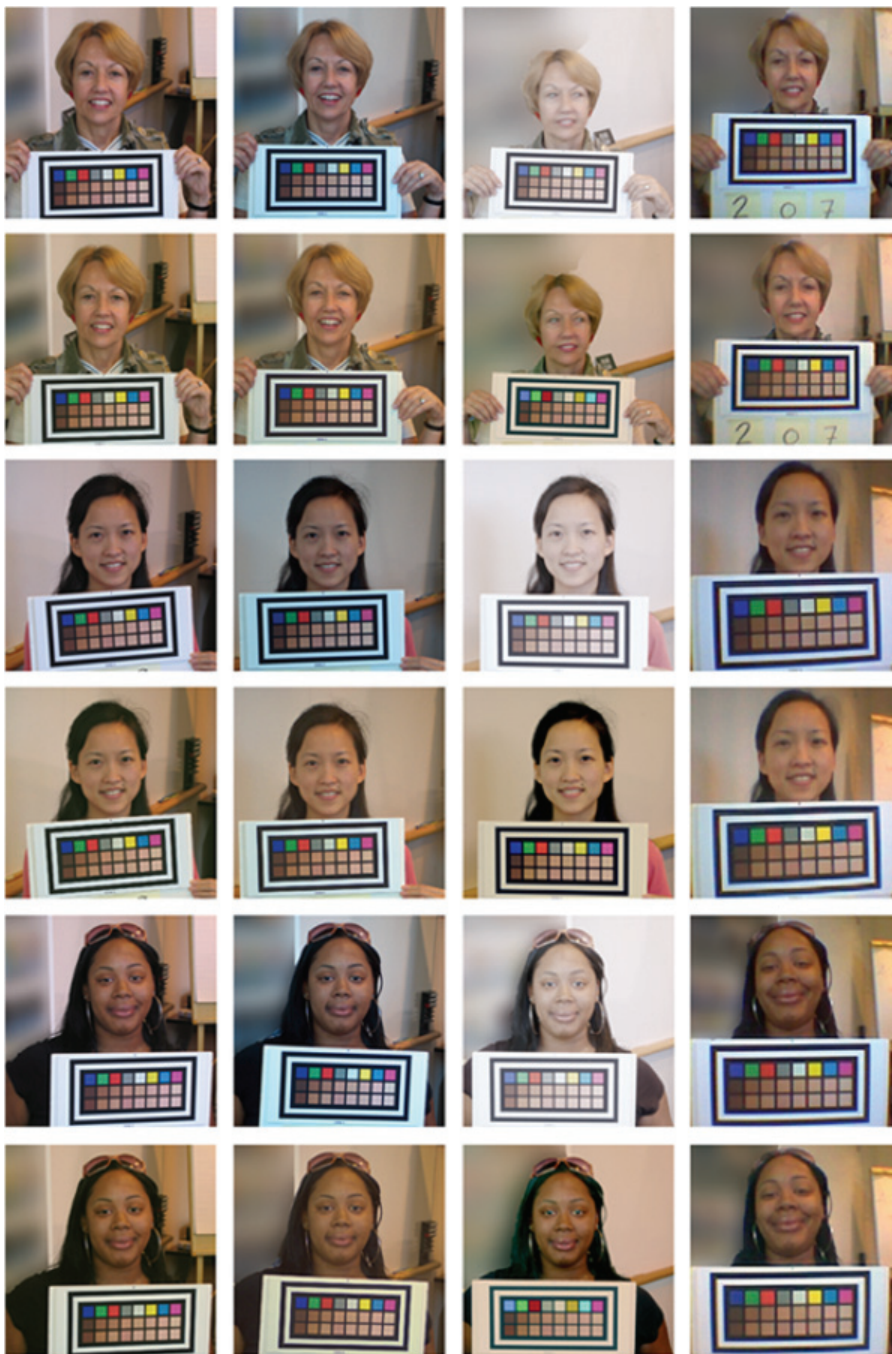
Skin tone is assessed from a single uncalibrated image taken in fully automatic mode with a consumer camera under uncontrolled illumination conditions. We have no access to raw data and have no information on the in-camera processing; the color correction is performed on already rendered images. We suppose that the camera performs white-balancing and encodes images in sRGB, which has a defined white-point of D65. Image rendering implemented in consumer cameras aims at producing pleasing images resembling our memory of the scene. Saturation and contrast are generally boosted and such output-referred images are not colorimetrically accurate. Moreover, due to imperfect illuminant compensation, different sensor sensitivities, and variations in image processing and in quality across devices, uniform color rendering is never achieved. The resulting image colors of a given scene imaged with different uncalibrated cameras will have important variations. Figure 3.4 shows examples of possible variations in color across devices. It displays the images of three subjects imaged with four different cameras used in fully automatic mode before and after color correction. To be able to classify skin tones, we need face pixels to have the same sRGB values independently of the illuminant and camera. In other words, we need the image sRGB to correlate with the actual colorimetry of the skin rather than its appearance after an “ideal” image rendering. We use a known reference target present in an image to compute a color correction transform.

The target patches are extracted automatically. Their detection is based on the segmentation of the image into regions whose contours are located at zero crossings of the Laplacian of a smoothed luminance version of the image (Harville *et al.* 2005). Their values are averaged and compared against a set of pre-computed reference sRGB triplets allowing the computation of an image dependent color transform. The correction is applied to the entire image prior to a Viola-Jones (2001) face extraction. The entire workflow is illustrated in Figure 3.2.

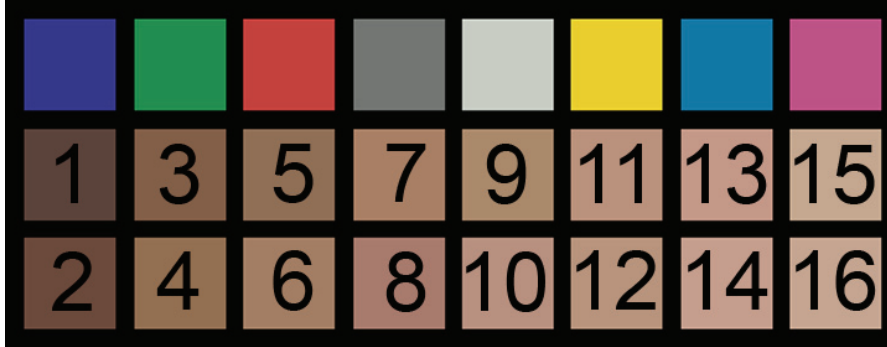
#### 3.3.1 Reference target

The target contains three rows of eight patches set against a black background and surrounded by a frame used for its automatic detection (see Figure 3.5). The first row contains primary and secondary colors and two shades of gray. The two last rows contain 16 patches characteristic of the range of human skin ordered by uniformly increasing lightness alternating on two rows.

We printed the target on matte photopaper medium and measured the reflectance spectrum of each patch. Figure 3.6 shows the desired patch spectra and the actual printed target spectra. The solid lines represent spectra typical of human skin. The dotted lines represent spectra of the printed skin color patches. These patches must have reflectances as smooth and as close to actual



**Figure 3.4:** The image color values of a given scene imaged with different uncalibrated cameras can have important variations. The subjects were imaged using four cameras: a HP850 (3.9 megapixels), a Canon S400 (4.0 megapixels), a Nikon D1 (2.7 megapixels), and a Nokia 6820 cell phone camera (0.1 megapixels). Rows 1, 3, and 5 show the uncorrected images and row 2, 4, and 6 show the corrected images.



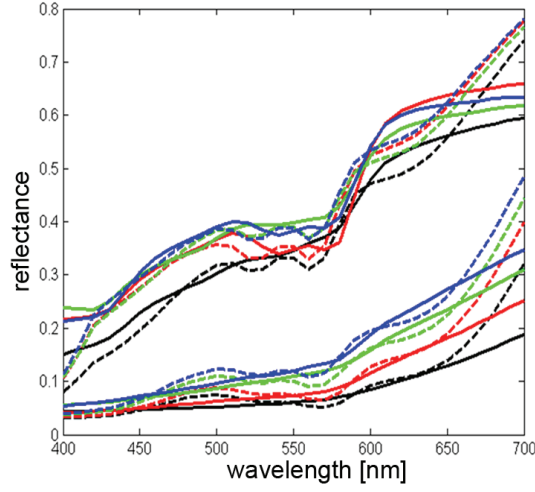
**Figure 3.5:** The target contains three rows of eight patches. The two last rows contain 16 patches characteristic of the range of human skin. The patches of the actual target are not numbered.

human skin spectra as possible to avoid metamerism. The measured spectra allow computing first CIE XYZ values under illuminant D65 using the image formation model (2.23), where  $\mathbf{s}(\lambda)$  are the target reflectances,  $\mathbf{e}(\lambda)$  is the standard illuminant D65, and  $\mathbf{r}_k(\lambda)$  are the CIE  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{z}$  color matching functions. The CIE XYZ values are converted to sRGB according to (2.5) and (2.6).

### 3.3.2 Color transform

We are computing a linear transform that maps the pixel values of an incoming image, referred as *image extracted* or *uncorrected*, onto *reference* ones. After correction, the image pixels are referred as *corrected*.

While incomplete illuminant compensation and variations in in-camera processing may result in different pixel values, the image encoding (sRGB) is well defined. However, sRGB is not perceptually uniform and it seems natural to compute a transform that will minimize a perceptually meaningful error, such as CIELAB  $\Delta E_{ab}^*$  (2.16) (Wu *et al.* 2000). That is, sRGB image extracted target values are converted into CIE XYZ and then CIELAB ones, that are then compared with CIELAB reference values. The color transform can minimize, for example, a color distance  $\overline{\Delta E_{ab}^*}$  averaged over the target values. The resulting transform is applied to the face pixels, which are then used for the skin tone classification. This approach requires several color space conversions that will slow down the transform computation. As we need the color correction to be fast, we are rather computing a transform that minimizes a Euclidian distance between image extracted sRGB target values and sRGB reference ones. Furthermore, the image values are not linearized prior to the color transform computation, which brings several advantages, besides allowing a faster computation. First, the target skin colored patches were chosen such that they are uniformly distributed in lightness and keeping the nonlinearity in sRGB allows keeping “perceptual” intervals between target patches. Second, the in-camera processing includes nonlinear rendering steps, such as tone mapping, that may introduce errors if linearizing the image extracted sRGB values.



**Figure 3.6:** Printing accuracy of the reference target: The solid and dashed lines represent spectra of the desired and printed skin colored patches, respectively.

Let us now briefly go back to the general case of linear transform computations for color space conversions and consider a transform  $\mathbf{A}_{1 \rightarrow 2}$  that maps values between two arbitrary color spaces. We have

$$\mathbf{S}_2 = \mathbf{S}_1 \mathbf{A}_{1 \rightarrow 2}, \quad (3.1)$$

where  $\mathbf{S}_1 = [R_1 G_1 B_1]$  and  $\mathbf{S}_2 = [R_2 G_2 B_2]$  are  $31 \times 3$  matrices containing the respective primaries of the color spaces 1 and 2 in vector form. In other words, we are looking for a transform  $\mathbf{A}_{1 \rightarrow 2}$  that maps  $\mathbf{S}_1$  onto  $\mathbf{S}_2$ . This transform must minimize *some* error function, such as the square error  $\|\mathbf{S}_1 \mathbf{A}_{1 \rightarrow 2} - \mathbf{S}_2\|_2$ . It can be computed using the Moore-Penrose pseudo-inverse denoted  $^+$

$$\mathbf{A}_{1 \rightarrow 2} = \mathbf{S}_1^+ \mathbf{S}_2 = (\mathbf{S}_1 \mathbf{S}_1^T)^{-1} \mathbf{S}_1^T \mathbf{S}_2. \quad (3.2)$$

We are applying the same formalism in our approach. We need to find a transform that maps sRGB values extracted from images to sRGB values that correlate with the skin colorimetry, i.e., that are independent of the camera characteristics and of the lighting conditions. We do not use a traditional  $3 \times 3$  color transform, but rather use a  $3 \times 4$  transform. It acts as a  $3 \times 3$  color transform, plus an offset accounting for the variations in exposure across images. The matrix  $\mathbf{A}$  maps the target patches mean color values  $\mathbf{M}$  extracted from the image onto reference target values  $\mathbf{T}$

$$\mathbf{T}_{\{3 \times n\}} = \mathbf{A}_{\{3 \times 4\}} \cdot \mathbf{M}_{\{4 \times n\}}, \quad (3.3)$$

where  $\mathbf{T}$  is a matrix whose  $i$ th column contains the  $i$ th value of the  $n$  reference patches  $\mathbf{t}_i = (t_i^{red}, t_i^{green}, t_i^{blue})^T$  and  $\mathbf{M}$  is a matrix whose  $i$ th column contains the  $i$ th value of the  $n$  mean camera patch color  $\mathbf{m}_i = (m_i^{red}, m_i^{green}, m_i^{blue}, 1)^T$ .

We want to find  $\mathbf{A}$  that minimizes  $\|\mathbf{T} - \mathbf{A} \mathbf{M}\|_2$ , i.e., a least mean square

(LMS) error in sRGB color space.  $\mathbf{A}$  is computed as

$$\mathbf{A} = \mathbf{T}\mathbf{M}^+ . \quad (3.4)$$

The pseudo-inverse of  $\mathbf{M}$  is computed by singular value decomposition. For each image, the target patches are extracted and the color transform  $\mathbf{A}$  (3.4) is calculated. This transform is then applied to the entire image prior to face pixels extraction and color classification (Harville *et al.* 2005). The linear transform  $\mathbf{A}$  will differ depending on the camera characteristics and lighting conditions.

The least mean square transform computation (3.4) is fast, but gives no control on the error repartition. While the color of the target patches will be properly corrected, the colors located “in between” the target patches cannot be exactly controlled. To ensure the correction accuracy, the colors considered must be limited to a small range, such as skin tones. For example, if the reference colors  $\mathbf{t}_i$  span a large gamut, such as a MacBeth ColorChecker, the sampling of the color space is coarse and, while the overall appearance of the image will be good, the correction will not be accurate enough to obtain a colorimetric estimate of the scene. If we limit the range of colors to be corrected, it gives some control on the error amplitude and we can assume that the colors falling in the gamut formed by the reference values will be accurately corrected. Also, the overall appearance of the image after correction may be poor, which is unimportant as long as the skin pixels are properly corrected.

### 3.3.3 Validity of the color correction approximation by a linear transform

In this section, we demonstrate that the color correction can be approximated by a linear transform. Let us assume that there exists a function  $f(\mathbf{x})$  which exactly maps the uncorrected normalized sRGB image values onto ideal sRGB ones

$$\begin{aligned} f : [0, 1]^3 &\longrightarrow [0, 1]^3 \\ \mathbf{x} &\longmapsto \mathbf{y} = f(\mathbf{x}) . \end{aligned} \quad (3.5)$$

Let us also assume that this function is continuous and  $n$ -times continuously derivable over  $[0, 1]^3$ . Moreover, let us assume  $f''(\mathbf{x})$  to be bounded. Under these assumptions,  $f(\mathbf{x})$  can be expanded using Taylor’s formula

$$f(\mathbf{x}) = f(\mathbf{a}) + \nabla f(\mathbf{a})(\mathbf{x} - \mathbf{a}) + (\mathbf{x} - \mathbf{a})^T \mathbf{H}(\mathbf{a})(\mathbf{x} - \mathbf{a}) + R_2(\mathbf{x}) , \quad (3.6)$$

where  $\nabla f(\mathbf{a})$  and  $\mathbf{H}(\mathbf{a})$  are, respectively, the gradient and the Hessian matrix of  $f$  evaluated at  $\mathbf{x} = \mathbf{a}$  and  $R_2(\mathbf{x})$  is the remainder. In this case,  $\mathbf{x} = (R, G, B)$  represents a face color and  $\mathbf{a} = (R_{\text{ref}}, G_{\text{ref}}, B_{\text{ref}})$  represents the center of mass of the reference colors or, in other words, the average reference value. As we are considering a limited range of colors and assuming that all face colors are close to the reference ones,  $|\mathbf{x} - \mathbf{a}|$  remains small and the second order term can be neglected. We thus have

$$f(\mathbf{x}) \cong f(\mathbf{a}) + \nabla f(\mathbf{a})(\mathbf{x} - \mathbf{a}) + R_2(\mathbf{x}) . \quad (3.7)$$

The remainder  $R_2$  can be evaluated using Cauchy's estimate

$$|R_2| \leq \sup_{\mathbf{y} \in [0,1]^3} \left| \frac{1}{2!} \frac{\partial^2 f(\mathbf{y})}{\partial y_1 \partial y_2} \right| = B \quad (3.8)$$

and we then have that

$$f(\mathbf{x}) \cong f(\mathbf{a}) + \nabla f(\mathbf{a})(\mathbf{x} - \mathbf{a}) + B. \quad (3.9)$$

If we regroup the constant terms into one constant  $B_o$ , we finally obtain that

$$f(\mathbf{x}) = \nabla f(\mathbf{a})\mathbf{x} + B_o. \quad (3.10)$$

The function  $f(\mathbf{x})$  maps uncorrected sRGB values onto corrected ones: If the images were ideally rendered, this function would tend towards the identity. The assumptions on  $f(\mathbf{x})$ 's continuity are thus reasonable and justified *a posteriori* by experiment. Equation (3.10) holds and the color correction can indeed be approximated by a linear term plus an offset, as presented in Section 3.3.2.

## 3.4 Experiments and results

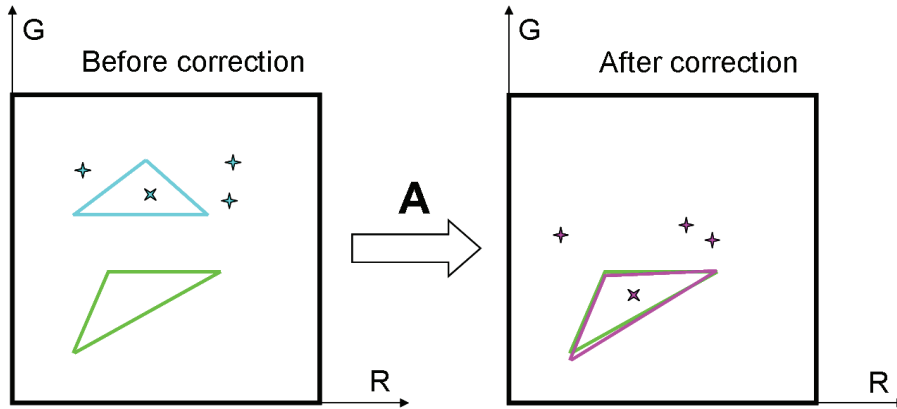
Fifty-three subjects holding a copy of the color calibration target were imaged with four cameras: a HP850 (3.9 megapixels), a Canon S400 (4.0 megapixels), a Nikon D1 (2.7 megapixels), and a Nokia 6820 cell phone camera (0.1 megapixel). The HP850 images are reported in Appendix C.1. The illuminant is a mixture of a side daylight illumination through a window and an overhead fluorescent lighting. The skin reflectance of the 53 subjects was also measured on a uniform area of the cheek with a portable Microflash spectrometer using a  $0^\circ/45^\circ$  geometry.

We determine which patches should be used for the best color correction accuracy in Section 3.4.1. We show in Section 3.4.2 that computing the color correction transform in nonlinear or linear sRGB results in an equivalent correction accuracy. We compare face colors values extracted from the images with sRGB ones derived from the skin spectral measurements in Section 3.4.3 and finally look at face colors obtained with the different imaging devices in Section 3.4.4.

### 3.4.1 Color correction accuracy

The color correction transform  $\mathbf{A}$  depends on the target patches used to compute (3.3). Harville *et al.* (2005) demonstrated that the primary and secondary colors should be left out. It was not established whether the achromatic patches should be considered in the color correction transform computation. The black and white background could be employed in (3.3), but as these colors are the most likely to be clipped, they are not taken into account.

The matrix  $\mathbf{A}$  maps the target patch values extracted from an image onto reference values. We can visualize this color correction by looking at the convex hulls of the reference, image extracted, and color corrected target values in normalized sRGB, as illustrated by Figure 3.7. The experiment was run on

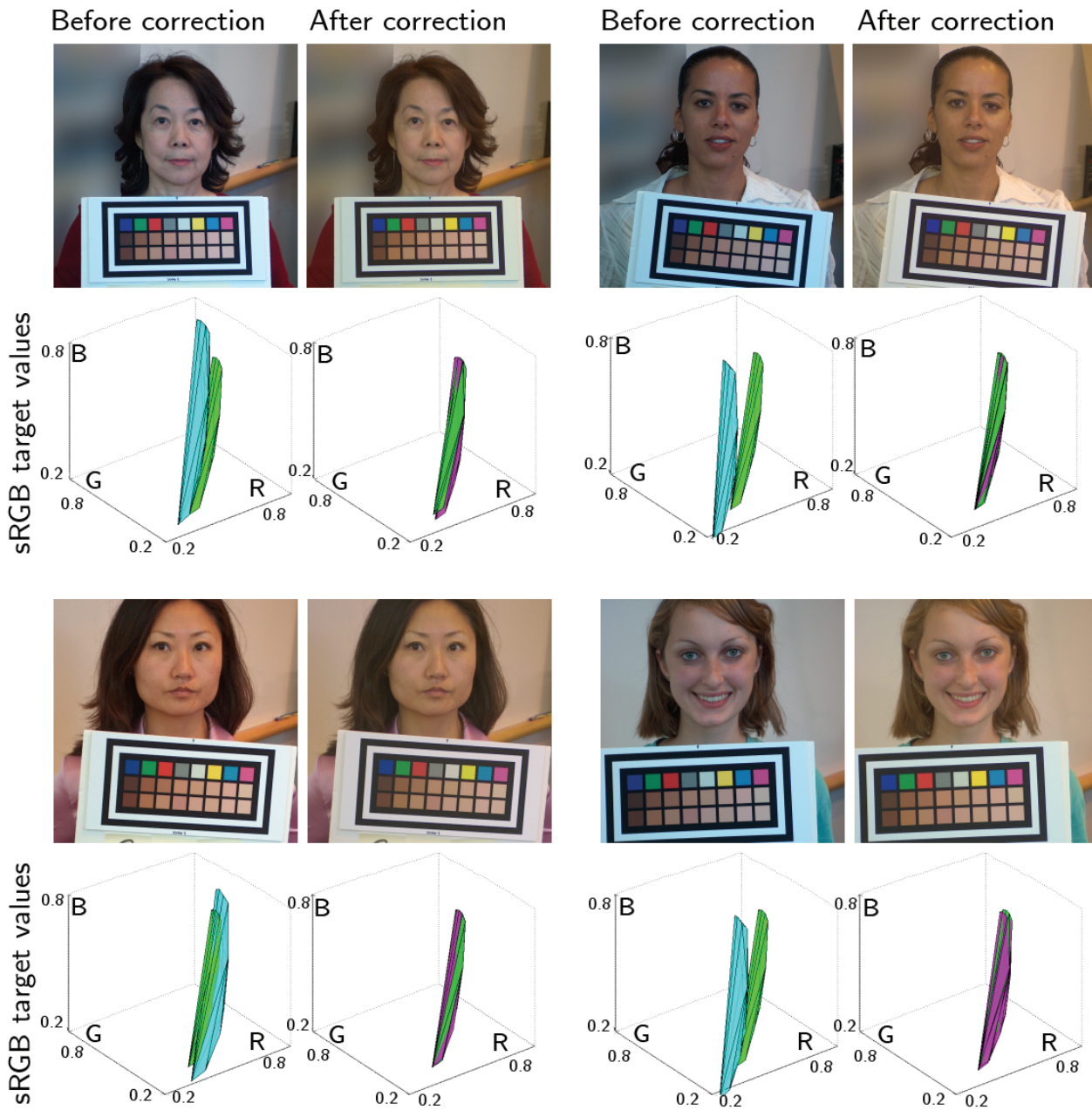


**Figure 3.7:** Two-dimensional illustration of the color correction. The gamuts of the uncorrected, reference, and corrected target values are represented respectively by the cyan, green, and magenta triangles. Colors located within the reference gamut, represented by the 'x', are accurately corrected, as opposed to colors located outside of the reference gamut, represented by the '+'.

the 53 images taken with the HP850 camera. Figure 3.8 displays four images from this set: the images before correction have variations in color, despite being captured with the same camera. The gamuts of uncorrected, reference, and corrected skin colored target patches illustrate how accurately the color correction maps the skin tones.

In order to assess the usefulness of the gray patches in determining the color correction transform, we compare the outcome of a set of color correction transforms  $\mathbf{A}_s^{\text{test}}$  computed using solely skin colored patches versus a set of color correction transforms  $\mathbf{A}_{s+g}^{\text{test}}$  computed with the two additional gray patches. The target is the only well defined element in the image, we thus use it to estimate the color transform performance with a leave-one-out method on its patches. Each of the 16 skin colored patches is successively corrected by a color transform computed from the remaining patches - 15 skin colored and the two optional gray patches. The color correction error is computed as the Euclidian distance between the normalized sRGB and CIELAB coordinates of the image extracted test patch value after color correction and its reference. The error is averaged over the 16 skin colored patches extracted from 53 images, i.e., a total of 848 patches. Table 3.1 reports the results. We see that leaving out the gray patches in the computation of  $\mathbf{A}^{\text{test}}$  allows a color prediction of an average  $\Delta E_{ab}^* \approx 0.86$  as opposed to  $\Delta E_{ab}^* \approx 1$  with the middle grays. The average error in normalized sRGB are respectively 1% and 1.2%. Skin tones are thus better mapped with a color correction transform computed from skin tones without additional gray patches. Indeed, leaving out the gray patches limits the size of the target gamut, which results in a more accurate correction of skin tones. The standard deviation is quite large: The error  $\Delta E_{ab}^*$  averaged over the 53 images ranges from 0.59 for patch No. 13 (numbered according to Figure 3.5) to 2.72 for patch No. 1. This is explained by the “leave-one-out” method: The corrected target test patch may lie outside of the gamut formed





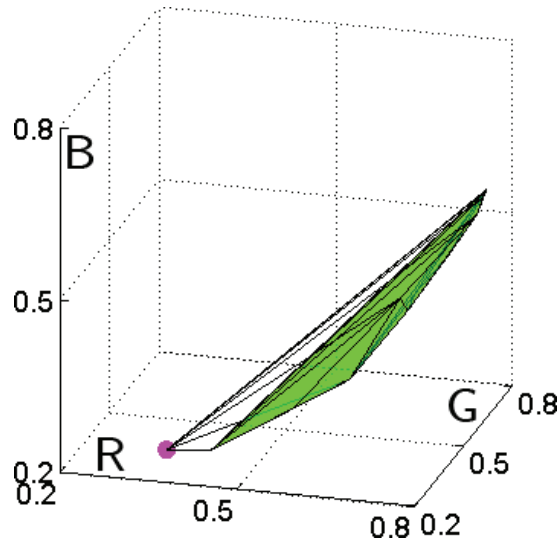
**Figure 3.8:** Four subjects were imaged in the same conditions with the HP850 camera used in automatic mode. Under each pair of images, we show the convex hulls of the image uncorrected skin colored target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.

by the remaining 15 skin colored patches used to compute the color transform  $\mathbf{A}^{\text{test}}$  and cannot be corrected accurately. Figure 3.9 shows test patch No. 1



	norm $sRGB$			$\Delta E$		
	mean	median	std	mean	median	std
skin	0.010	0.009	0.005	1.03	0.86	0.69
skin + gray	0.012	0.011	0.004	1.15	1.00	0.57

**Table 3.1:** Color differences in normalized  $\Delta sRGB$  and  $\Delta E_{ab}^*$  for transforms using skin colored patches only vs. transform using two additional middle gray patches. The error is computed on the 16 patches using a leave-one-out method and is averaged over the 53 images.



**Figure 3.9:** If the skin colored patch values (magenta dot) are too different from the 15 patches (green surface) used to compute the color correction transform, it cannot be accurately color corrected.

(magenta dot), corrected with an error  $\Delta E_{ab}^* = 2.72$ , and the convex hull formed by the 15 remaining skin colored patches (green surface). Patch No. 1 indeed lies outside of the gamut formed by the remaining patches. We use the difference in volume between the gamuts formed by the 15 patches with and without the test patch as an estimate of how “far out” the test patch lies from the reference colors by computing

$$\Delta V = \frac{V^{\text{skin}} - V^{\text{test}}}{V^{\text{test}}}, \quad (3.11)$$

where  $V^{\text{test}}$  and  $V^{\text{skin}}$  are the volumes of the convex hulls formed by the 15 patches and the 16 skin colored patches, respectively.  $\Delta V$  correlates well with  $\Delta E_{ab}^*$ .  $\Delta V$  ranges from zero if the test patch is in the reference gamut to 0.68 for patch No. 1. While this points out a shortcoming of the color correction accuracy estimation method, it illustrates how important it is that the color of

	<i>sRGB</i>		$\Delta E$	
	mean	std	mean	std
16 patches	0.007	0.003	0.71	0.39
8 patches: row 1	0.009	0.003	0.82	0.42
8 patches: row 2	0.009	0.004	0.81	0.47
8 patches CIELAB	0.008	0.003	0.76	0.41
4 patches CIELAB	0.008	0.005	0.82	0.85

**Table 3.2:** Color differences in normalized sRGB and  $\Delta E_{ab}^*$  for transforms using 16, 8, and four patches, respectively. The error is computed on the 16 extracted skin patches averaged over the 53 images.

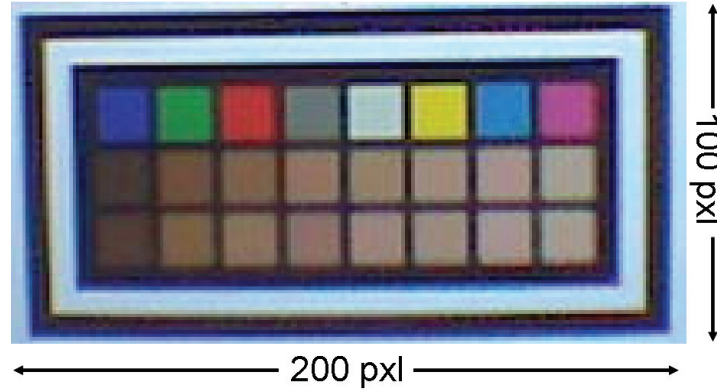
the object to be corrected lies close to the gamut formed by the target colors used in the transform computation (Eq. 3.3).

The number and size of the patches have an influence on the quality of the color correction. To compute (3.3),  $\mathbf{T}_{\{3 \times n\}}$  and  $\mathbf{M}_{\{4 \times n\}}$  must have at least four columns and thus the minimal set contains four patches; four points are necessary to span a volume in a three-dimensional space. We estimate the influence of the number of patches by computing five transforms computed from

1. 16 skin colored patches
2. 8 patches: the first row of skin tones (odd numbers)
3. 8 patches: the second row of skin tones (even numbers)
4. 8 patches (1, 2, 5, 8, 9, 10, 12, and 15) selected manually to form a convex hull of all skin colored patches in CIELAB, including two patches in the center
5. 4 patches (1, 8, 10, and 12) selected manually to form a convex hull of all skin colored patches in CIELAB.

Table 3.2 reports the color correction errors when 4, 8, and 16 patches  $\mathbf{t}_i^{\text{test}}$  are used for the color correction. There is no significant difference between the average  $\Delta E_{ab}^*$  errors resulting from color transform computed with 4, 8, or 16 patches, but the standard deviation  $\sigma$  is larger when only using 4 patches, indicating that more patches allow a more uniform color correction of skin tones.

For a better color correction accuracy, a larger number of patches is more desirable, but reducing the number of patches allows one to increase their size. Larger color patches are better detected, especially for low resolution cameras. In the images taken with the Nokia 6820 cell phone, patches are as small as  $15 \times 15$  pixels (see Figure 3.10). At this scale, JPEG artifacts may cause the background to leak into the target patches and introduce important errors in the estimation of the patches' color. Even though skin tones *can* be color corrected using only four patches, the color correction is less uniform and less robust. If, for example, shadows are projected on just one patch, the resulting color transform becomes inconsistent. A higher number of patches



**Figure 3.10:** JPEG artifacts can introduce error in the color correction transform estimation for low resolution cameras.

allows a finer sampling of skin tones and in general more accurate transforms, while larger patches are easier to detect. The trade-off depends mainly on the camera resolution. Despite the recent increase in cell phone cameras resolution, the file size of the MMS message is limited by the carrier: the image will be downsampled and resolution remains an issue.

### 3.4.2 Computation of the transform in linear sRGB

In this section, we are comparing the correction accuracy of the transform computed in nonlinear sRGB as presented in Section 3.3.2 (referred to as NL for “nonlinear”) with the accuracy that we can obtain if computing it in linear sRGB. We are testing two new types of color transforms: we compute  $\mathbf{A}$  by comparing linear reference values with first image values linearized by applying a fixed gamma  $\gamma = 2.4$  (referred to as L1) corresponding to the nonlinearity of the sRGB color encoding (Section 2.4.2) and then with image values linearized by applying an image dependent gamma  $\gamma_{\text{var}}$  (referred to as L2). In the latter case, we deduce  $\gamma_{\text{var}}$  by comparing the logarithm of linear reference values against the logarithm of the image extracted target values. Let us consider the equation

$$x_{\text{ref}} = y_{\text{lin}} = y_{\text{im}}^{\gamma_{\text{var}}}, \quad (3.12)$$

where  $x_{\text{ref}}$ ,  $y_{\text{lin}}$ , and  $y_{\text{im}}$  represent the red, green, and blue components of, respectively, the linear reference, the linearized image (which we are looking for), and the image patch values. If we take the logarithm of this expression, we have

$$\log x_{\text{ref}} = \log y_{\text{lin}} = \log y_{\text{im}}^{\gamma_{\text{var}}} = \gamma_{\text{var}} \log y_{\text{im}} \quad (3.13)$$

or equivalently

$$\gamma_{\text{var}} = \frac{\log x_{\text{ref}}}{\log y_{\text{im}}}. \quad (3.14)$$

The slope of the linear regression of  $\log x_{\text{ref}}$  against  $\log y_{\text{im}}$  thus gives us the varying gamma  $\gamma_{\text{var}}$ . The distributions of the resulting  $\gamma_{\text{var}}$  are displayed in

	mean	median	std	mean	median	std
	HP			Canon		
NL	1.03	0.86	1.15	<b>1.04</b>	<b>0.89</b>	<b>0.64</b>
L1	<b>1.00</b>	<b>0.84</b>	<b>1.31</b>	1.10	0.95	0.66
L2	<i>1.06</i>	<i>0.88</i>	<i>1.38</i>	1.17	0.99	0.75
	Nikon			Nokia		
NL	1.64	1.29	1.40	2.04	1.81	1.18
L1	1.45	1.17	1.06	<i>2.03</i>	<i>1.83</i>	<i>1.10</i>
L2	<b>1.19</b>	<b>0.96</b>	<b>0.80</b>	<b>2.01</b>	<b>1.81</b>	<b>1.08</b>

**Table 3.3:** Color differences in  $\Delta E_{ab}^*$  for transforms computed in nonlinear sRGB (NL), in sRGB linearized with  $\gamma = 2.4$  (L1), and in sRGB linearized with an image dependent  $\gamma_{\text{var}}$  (L2). The numbers in italic denote that the  $\Delta E_{ab}^*$  difference of L1 or L2 compared to NL is not statistically significant (with  $\alpha = 0.05$ ).

Figure C.1.

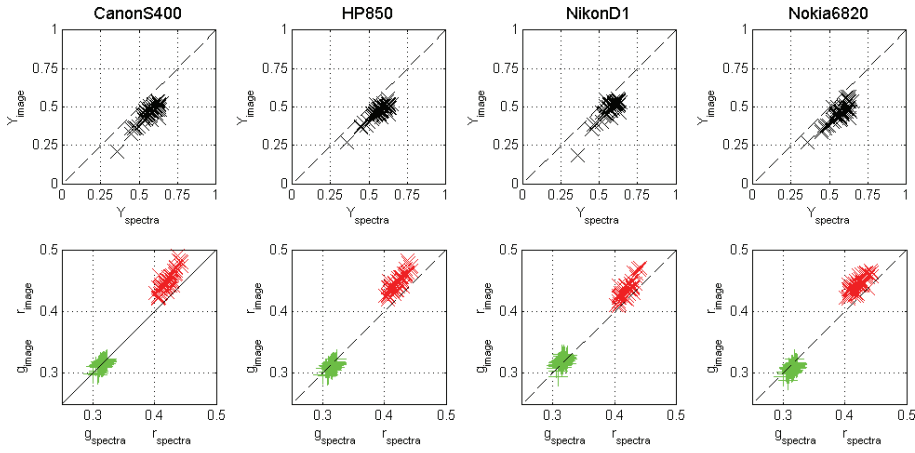
The color correction accuracy is expressed in terms of CIELAB  $\Delta E_{ab}^*$  differences computed using the leave-one-out method on the target patches as reported in Section 4.3.2. The statistical significance of these differences is tested by applying the Kruskal-Wallis nonparametric one-way analysis of variance (Whitley and Ball 2002), which returns the  $p$ -value (between 0 and 1) for the null hypothesis that the two, or more, data samples are drawn from the same population, i.e., have the same median and distribution. More precisely, if the  $p$ -value is smaller than the confidence level  $\alpha$ , the null hypothesis is rejected with a probability  $1 - \alpha$ , that is, the samples are from different populations, which in our case means that the  $\Delta E_{ab}^*$  differences are statistically significant. A  $p$ -value larger than  $\alpha$  indicates a failure to reject the hypothesis, that is, the samples probably have the same distribution and the variations of  $\Delta E_{ab}^*$  are caused by fluctuations without statistical significance. The choice of  $\alpha$  depends on the desired confidence level; it is common to choose  $\alpha$  equal to 0.01 or 0.05, corresponding to a reliability of the test result of, respectively, 99% and 95%.

Table 3.3 reports the correction error in terms of CIELAB differences for the three color transform computations. The smallest  $\Delta E_{ab}^*$  differences are reported in bold for each of the four cameras. We see that the transform correction leading to the best accuracy depends on the camera and that none of them systematically yields to a more precise correction. Moreover, the differences in  $\Delta E_{ab}^*$  across cameras are larger than the ones across transforms for a given camera.

We extend the analysis by comparing each of the three transforms for all pairs of cameras and report which transform performs the best in each case. Table 3.4 reports, respectively, the performance of NL vs. L1, N1 vs. L2, and L2 vs. L1. We also compute the statistical significance of these results with the Kruskal-Wallis test, whose  $p$ -values are reported in Section C.3, and only keep the statistically significant results. These tables must be read as follows: the diagonal terms correspond to the comparisons between two different transforms applied on the same images, while the non-diagonal terms represent the

a)	L1-HP	L1-Canon	L1-Nikon	L1-Nokia
NL-HP	<i>L1</i>	NL	NL	NL
NL-Canon	L1	NL	NL	NL
NL-Nikon	L1	L1	<i>L1</i>	NL
NL-Nokia	L1	L1	L1	$\emptyset$
b)	L2-HP	L2-Canon	L2-Nikon	L2-Nokia
NL-HP	$\emptyset$	NL	NL	NL
NL-Canon	L2	NL	NL	NL
NL-Nikon	L2	L2	L2	NL
NL-Nokia	L2	L2	L2	$\emptyset$
c)	L1-HP	L1-Canon	L1-Nikon	L1-Nokia
L2-HP	<i>L1</i>	L2	L2	L2
L2-Canon	L1	L1	L2	L2
L2-Nikon	L1	L1	L2	L2
L2-Nokia	L1	L1	L1	$\emptyset$

**Table 3.4:** Comparisons of a) the transform computed with nonlinear sRGB values (NL) against the one computed with linear sRGB (L1,  $\gamma = 2.4$ ), b) the transform computed with nonlinear sRGB values (NL) against the one computed with linear sRGB (L2,  $\gamma_{\text{var}}$ ), and c) the transform computed with linear sRGB values (L2,  $\gamma_{\text{var}}$ ) against the one computed with linear sRGB (L1,  $\gamma = 2.4$ ) for each pair of cameras. The table reports the transform leading to the smallest median error  $\Delta E_{ab}^*$ . The symbol  $\emptyset$  indicates that the result is not statistically significant (with  $\alpha = 0.01$ ), while a number in italic indicates that  $0.01 \leq p\text{-value} \leq 0.05$ .



**Figure 3.11:** Spectrally derived values (x-axis) vs. image extracted values (y-axis) for Canon S400, HP850, Nikon D1, and Nokia 6820. The top row shows luma values  $Y = (R + G + B)/3$ . The bottom row shows normalized color coordinates  $r = R/(R + G + B)$  and  $g = G/(R + G + B)$ . The black dotted line indicates the identity relation.

comparisons between images captured with two different cameras and corrected with two different transforms. All off-diagonal results are statistically significant with a probability  $1 - \alpha = 99\%$ , which indicates that the camera model is of greater influence than the choice of sRGB for the transform computation. We can conclude from this analysis that computing the color correction transform in nonlinear sRGB results in an equivalent color correction accuracy, but requires less computation and is thus preferable to a computation in linear sRGB.

### 3.4.3 Face color derived from spectral measurements

The faces are extracted using the Viola-Jones face detection (Harville *et al.* 2005, Viola and Jones 2001) and only pixels having a luma between 10% and 90% - computed as  $Y = (R + G + B)/3$  - are considered in order to remove outliers due to hair, eyebrows, eyes, lips, and specularities. The mean color values of the remaining pixels after color correction are then taken as the skin color estimate. The skin reflectance of each face was also measured. We computed sRGB skin values for each of the 53 people from the measured skin reflectance spectra. sRGB values are converted into normalized color coordinates  $Y = (R + G + B)/3$ ,  $r = R/(R + G + B)$ , and  $g = G/(R + G + B)$ . These values were compared with face colors extracted from the HP850 images color corrected with the transform computed over the 16 skin colored patches. The correlation between extracted  $(Y_{\text{image}}, r_{\text{image}}, g_{\text{image}})$  face colors and spectrally derived  $(Y_{\text{spectra}}, r_{\text{spectra}}, g_{\text{spectra}})$  values is high (see Figure 3.11). However, spectrally derived values have systematically smaller  $r = R/(R + G + B)$  components and larger  $Y = (R + G + B)/3$  components than the ones extracted from the images. These differences can be caused by several factors: direct

measurement of skin reflectance by spectrometry is not optimal, as there are often important differences of shades within one face; the pressure of the probe can be a source of error; the skin is not Lambertian nor flat. The higher image  $r = R/(R + G + B)$  component is consistent with the pressure of the probe pushing the blood away and making skin appear whiter to the spectrometer. The lower image  $Y = (R + G + B)/3$  may be caused by a significant amount of shadows remaining in the estimated skin pixels despite the luma bounds applied to remove non-skin pixels. Skin color sRGB values extracted from color corrected images cannot be directly matched with the ones derived from spectra. Nevertheless, the systematic discrepancies indicate that the color correction of skin tones is coherent across cameras.

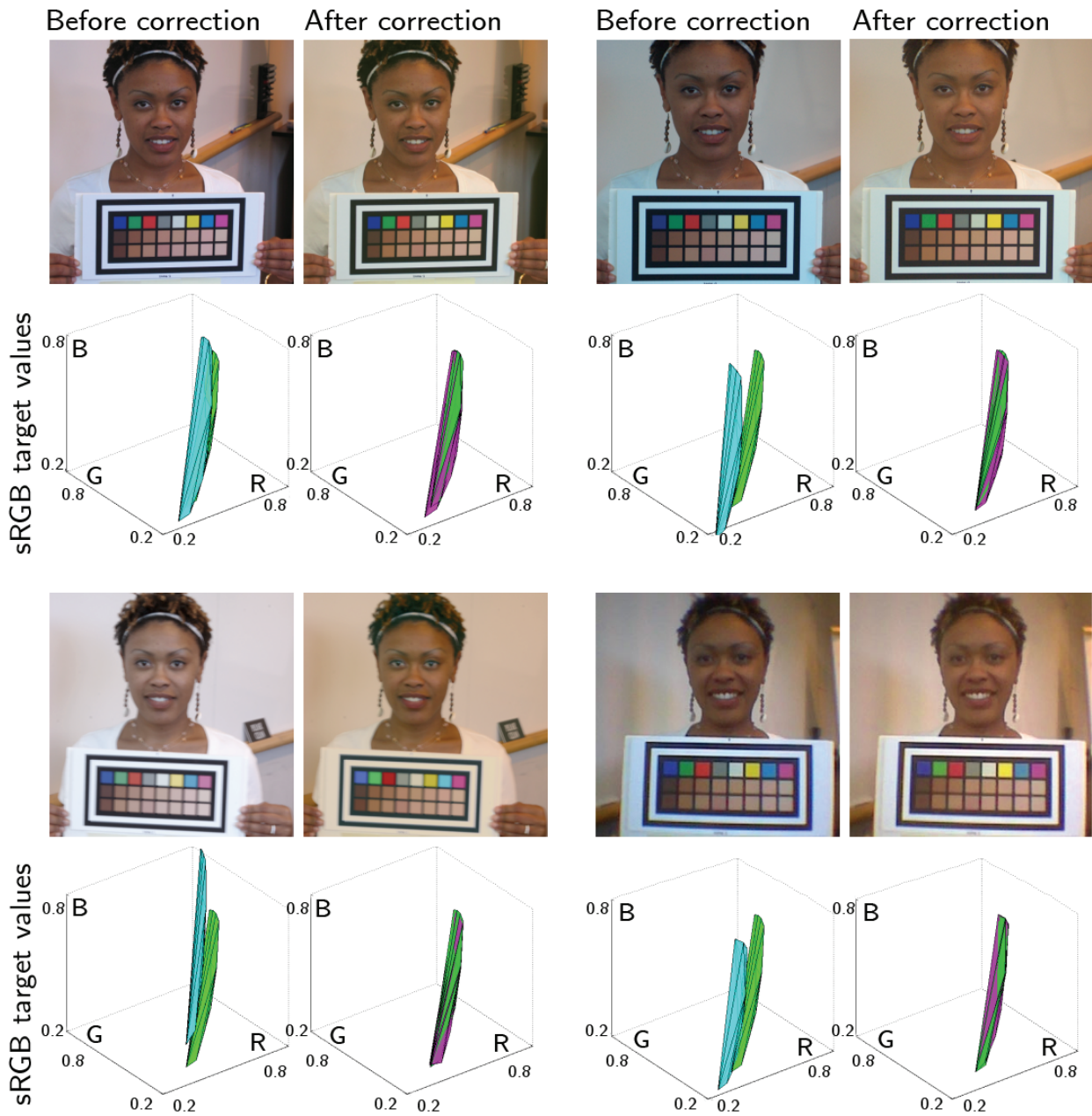
### 3.4.4 Color correction across cameras

Figure 3.12 shows examples of one subject imaged with four different cameras. Despite variations in color before correction, skin tones after correction are close. The figure also shows the gamuts of the corresponding target values in normalized sRGB space. The skin colored patch values extracted from the four images before correction (cyan surface) are different but overlap with the reference values (green) after correction (magenta). The method relies on the assumption that the subject face color lies in or close to the target gamut.

Although we could not directly link image corrected sRGB values with spectrally derived ones, the recommendation system only needs skin tones to be consistently assessed independently of the camera. The cross-camera correlation is estimated by comparing the luma  $Y = (R + G + B)/3$  and the color coordinates  $r = R/(R + G + B)$  and  $g = G/(R + G + B)$  of the face colors across cameras. Figures 3.13 and 3.14 show the luma  $Y$  and normalized chromaticity coordinates  $(r, g)$  extracted from images taken with the four cameras. Values are displayed for each pair of cameras. A black dotted line indicates the identity relation.

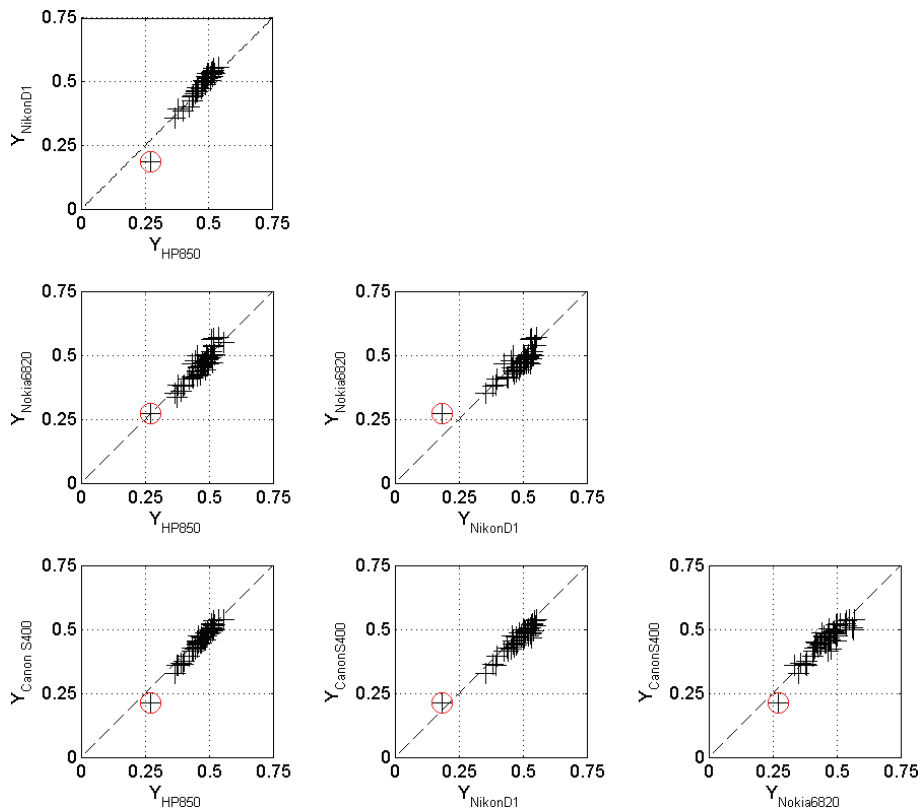
The correlation is high across all pairs of cameras and the relation is linear. Table 3.5 reports the correlation coefficients for the corresponding  $(Y, r, g)$  values for each pair of cameras and the equivalent CIELAB  $\Delta E_{ab}^*$  color difference averaged over all images. Correlation coefficients range from 0.90 to 0.98 for  $Y$ , from 0.57 to 0.96 for  $r$ , and from 0.82 to 0.91 for  $g$ . The correlation coefficients are lower for  $(r, g)$  compared to  $Y$ . This is due to the smaller range of values that  $(r, g)$  can take compared to  $Y$  and not to a lower estimation accuracy. We have indeed seen in Section 3.2.2 that lightness varies more than chromaticity across the range of possible human skin colors. The average absolute difference in normalized luma  $\Delta Y = |Y_1 - Y_2|$  ranges from 1.57% for the HP-Canon pair to 2.93% for the Nikon-Nokia pair, while the normalized chromaticity coordinates  $\Delta r = |r_1 - r_2|$  and  $\Delta g = |g_1 - g_2|$  range from, respectively, 0.42% and 0.27% for the HP-Canon pair to 1.16% and 1.09% for the Nikon-Nokia pair. The error in normalized sRGB  $= \|sRGB_1 - sRGB_2\|_2$  ranges from 0.035 for the HP-Canon pair to 0.064 for the Nikon-Nokia pair (see Table 3.6). The average  $\Delta E_{ab}^*$  ranges from  $\simeq 1.8$  to  $\simeq 5$ .

The quality of the estimation depends clearly on the quality of the camera. The low resolution of the Nokia images and the systematical overexposure of the Nikon dataset explain the less favorable result obtained for this pair. The resulting error  $\Delta E_{ab}^* \simeq 5$  is quite large, but the extracted face color still allows

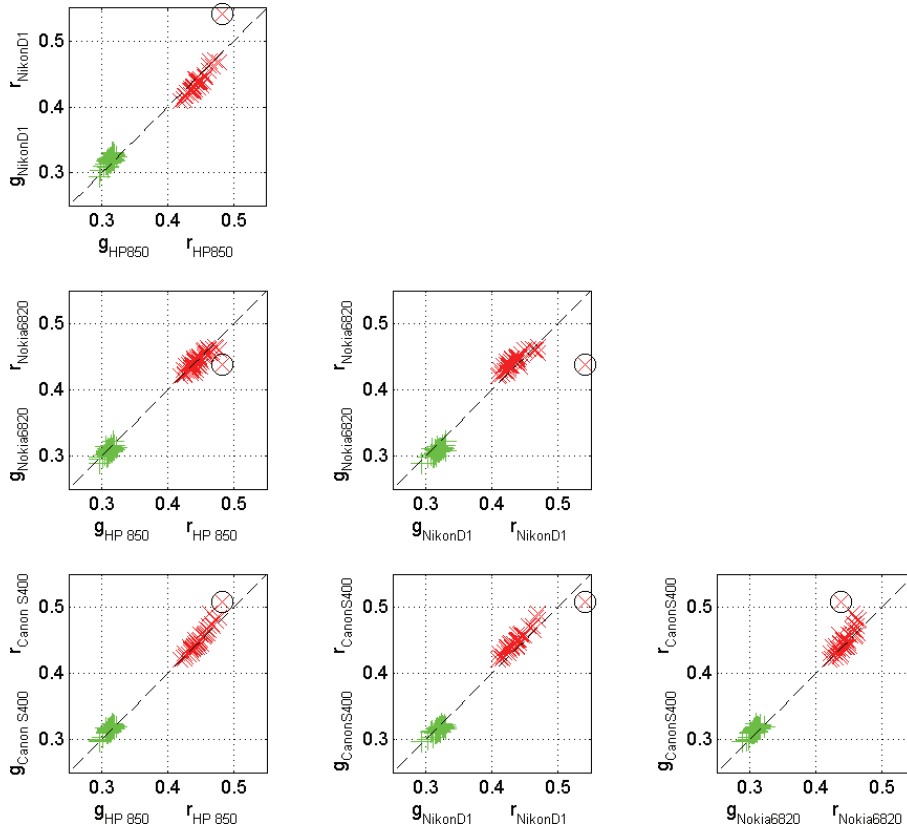


**Figure 3.12:** The subject was imaged with four cameras in the same conditions, but automatic white-balancing shows important variations. The cameras are, from left to right and top to bottom, a Canon S400, a HP850, a Nikon D1, and a cell phone camera Nokia 6820. Under each pair of images, we show the convex hulls of the image's uncorrected skin colored target values (cyan), reference target values (green), and image's corrected target values (magenta). The corrected and reference values overlap.





**Figure 3.13:** Luma  $Y = (R+G+B)/3$  values compared for each pair of cameras. The x-axis shows (from left to right) cameras HP850, Nikon D1, and Nokia 6820 and the y-axis show (from top to bottom) cameras Nikon D1, Nokia6820, and Canon S400. The black dotted line indicates the identity relation. The circle corresponds to the subject imaged in Figure 3.15.



**Figure 3.14:** Normalized color coordinates  $r = R/(R + G + B)$  and  $g = G/(R + G + B)$  compared for each pair of cameras. The x-axis shows (from left to right) cameras HP850, Nikon D1, and Nokia 6820 and the y-axis shows (from top to bottom) cameras Nikon D1, Nokia6820, and Canon S400. The black dotted line indicates the identity relation. The circle corresponds to the subject presented in Figure 3.15.

	HP	Nokia	Nikon	Canon	HP	Nokia	Nikon	Canon
	$Y$				$g$			
HP	1				1			
Nokia	0.93	1			0.80	1		
Nikon	0.98	0.90	1		0.86	0.82	1	
Canon	0.98	0.91	0.96	1	0.91	0.84	0.85	1
	$r$				$\Delta E_{ab}^*$			
HP	1				0			
Nokia	0.80	1			3.59	0		
Nikon	0.89	0.57	1		3.28	5.32	0	
Canon	0.96	0.71	0.94	1	1.90	4.00	3.40	0

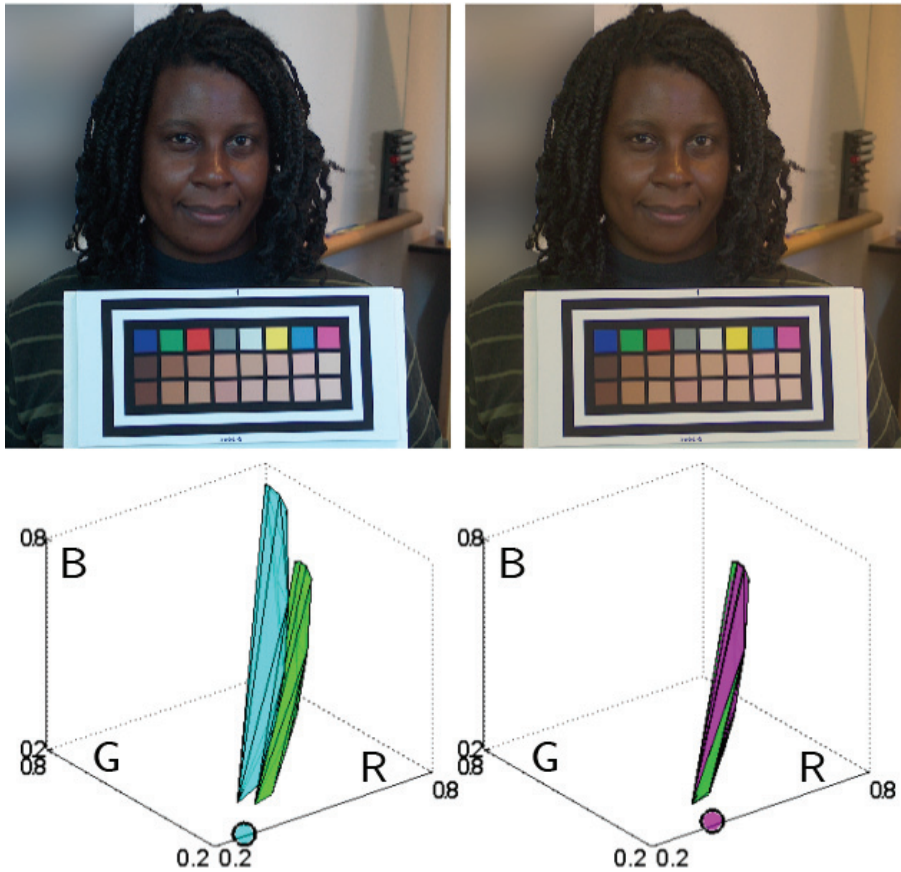
**Table 3.5:** Correlation coefficients between the  $(Y, r, g)$  values of extracted skin color and corresponding average  $\Delta E$  for each pair of cameras.

	HP	Nokia	Nikon	Canon	HP	Nokia	Nikon	Canon
	$\Delta Y =  Y_1 - Y_2 $				$\Delta g =  g_1 - g_2 $			
HP	0				0			
Nokia	0.020	0			0.005	0		
Nikon	0.018	0.029	0		0.007	0.011	0	
Canon	0.016	0.019	0.026	0	0.003	0.006	0.005	0
	$\Delta r =  r_1 - r_2 $				$\Delta sRGB = \sqrt{\Delta Y^2 + \Delta r^2 + \Delta b^2}$			
HP	0				0			
Nokia	0.006	0			0.047	0		
Nikon	0.011	0.012	0		0.042	0.064	0	
Canon	0.004	0.009	0.012	0	0.035	0.051	0.054	0

**Table 3.6:** Error between the  $(Y, r, g)$  and sRGB values of extracted skin color for each pair of cameras averaged over the 53 images.

for a good classification of skin tones. An accuracy of  $\Delta E_{ab}^* = 1$ , considered as the distance between two distinguishable color stimuli, may not be required for all applications.

A closer look at Figures 3.13 and 3.14 shows that the clouds of  $(Y, r, g)$  points are compact except for one outlier, indicated by the circles. It corresponds to a subject whose skin color lies outside of the gamut of target skin tones with  $\Delta V = 0.16$  (Eq. 3.11). Figure 3.15 shows the subject before and after color correction and the corresponding target gamuts. The cyan and magenta circles indicate the subject's mean face color before and after correction, respectively. Despite the target values being accurately mapped, the face color is too far from the reference colors to be properly corrected.



**Figure 3.15:** The outlier indicated by a circle in Figures 3.13 and 3.14 corresponds to a subject whose skin color lies outside of the target skin colored patches gamut. We see in Figure 3.14 that her skin tone does not get consistently corrected across the images.

---

## 3.5 Discussion

Colors cannot be accurately retrieved from images taken with uncalibrated cameras. However, a limited range of colors can be estimated by using appropriate information in the form of a target present in the scene. A transform mapping the scene target color values onto pre-computed target reference ones is calculated by least mean square estimation in sRGB and applied to the entire image. The least mean square estimation of the color correction matrix in sRGB allows an accurate yet fast and computationally inexpensive color correction. Nevertheless, the sRGB values of the image element to be corrected must lie close to the reference sRGB values used to compute the correction matrix.

With this method, skin tones can be consistently classified from uncalibrated images taken with a large variety of cameras and color corrected using a target consisting of patches characteristic of the range of human skin tones. With this mapping, skin tones can theoretically be estimated with an error of about 1% in sRGB and  $\Delta E_{ab}^* < 1$ . This bound is obtained by estimating the error on the known target patches, which are Lambertian and flat.

There is noise in the measurement of face colors using real images, due to the estimation of the face pixels, the shadows, and the texture and unevenness of skin. However, results correlate well with spectral data and across a variety of devices, with a resolution as low as 0.1 megapixel and poor image quality. These results were obtained using four cameras from four different manufacturers and without bypassing any of the image processing implemented in the camera.

The high correlation across devices indicates that with a solid ground truth, the system can be correctly trained to give consistent results. The assessment of color is not perfect, but it does not require any expensive calibrated imaging devices or controlled illuminant and can be performed with any consumer camera.

The method assumes a uniform illuminant across the image; shadows and mixed illuminants can be an important source of error. A comparison between the two halves of the face allows eliminating some failing cases. Also, the number and size of the patches have an influence. A greater number of patches gives a finer sampling of skin tones and more robustness. Reducing the number of patches allows to increase their size and thus improves the extraction and estimation of the target color values, especially for low resolution cameras. The trade-off depends on the application and on the type of cameras used.

## 3.6 Summary

A scene imaged under the same lighting conditions but using different cameras will generally result in different pixel values. This is due to incomplete illuminant compensation and to variations in image rendering. Consequently, colors cannot be accurately assessed from standard RGB images, unless the camera is calibrated.

In this chapter, we presented a post-calibration method for the correction of skin tones to be employed in a consumer makeup advice application, which excludes the use of calibrated devices. We demonstrated that accurate skin tone assessment can be achieved by imaging a subject's face along with a color

---

calibration target characteristic of human skin. The target pixels are extracted and their values compared to reference ones in order to compute a linear color correction transform that is applied to the entire image prior to the face pixels extraction.

We showed that skin tones can be corrected with an accuracy of  $\Delta E_{ab}^* < 1$ . When considering real images though, the error is larger and the average cross-camera color differences range from  $\Delta E_{ab}^* \sim 2$  to  $\Delta E_{ab}^* \sim 5$  for various camera pairs. However, image extracted face colors exhibit a high correlation with values derived from spectrometric measurements, despite a systematic discrepancy in lightness. More generally, we have demonstrated that skin tones can be accurately corrected provided that their values are close to those from the reference colors. In the next chapter, we will establish that a similar method can be applied to any range of colors.

## Chapter 4

# Color assessment for a Home Décor application

### 4.1 Introduction

In this chapter, we present how to extend the skin color assessment method presented in Chapter 3 to any object color in the framework of an advisory service for Home Décor. Instead of providing users with a foundation matching their skin tone, we want to give them advice on colors complementing their furniture or interior.

A color harmony is a pleasing combination of colors and many theories exist on how to obtain such combinations. Appendix D presents a short historical review of color harmony theories. Most commonly accepted rules concern combinations of hues. They are simple to express in terms of hue angles, but introducing rules for the influence of lightness and chroma or the sample size is a difficult task. Moreover, these theories only take into account the objective aspects of color, such as hue angles, but they do not take into account the more subjective aspects of color, such as color emotions or symbolism; subjective perception of colors varies among cultures. It is thus not possible to create universal rules of color harmonies. The simplest method to find sets of harmonious colors is to use existing palettes created by interior designers or artists, who remain the most qualified at applying the sometimes subjective laws of color harmony. For these reasons, we do not attempt to generate color harmonies, but are rather interested in providing a color assessment method allowing the use of any existing color palettes and mimic the opinion of an expert, the same way the foundation advice application employed a database built by a makeup expert.

In Chapter 3, we assessed skin color from a single digital picture taken with an unknown consumer camera under unknown lighting conditions, using solely a calibrated reference target representative of skin tones present in the scene. In this chapter, we demonstrate that this method can be extended to assess the color of virtually any object. As only a limited range of colors can be corrected with our method, we design a set of correction targets covering all hues. The system is fed with an image of an interior color to be coordinated, such as a piece of fabric or a portion of a wall. The object is imaged with an adapted target

allowing an accurate color correction of the object. The corrected object pixels are extracted, their values converted to CIECAM02 and compared against a database of paint samples to find the closest match. The system then returns a set of paints complementing that match. We demonstrate that our method can be applied to Home Décor. An expert, a professional interior designer, graded the results as very good. Her opinion was also used to optimize the matching metric.

We present our approach in detail in Section 4.2. Section 4.3 discusses the accuracy of the color correction and the grading of the results by a Home Décor expert. Section 4.4 concludes the chapter. A more complete review of color harmony theories can be found in Appendix D.

## 4.2 Our approach

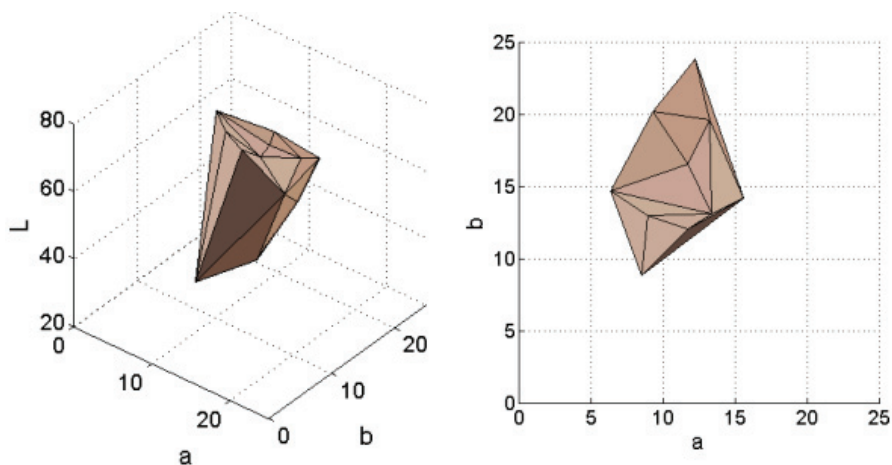
In Chapter 3, we demonstrated that we could assess skin tones from uncalibrated images with good accuracy ( $\Delta E_{ab}^* < 1$ ) using a color target consisting of patches covering the range of possible skin colors. Considering that we could color correct skin tones with a single target and that they span a hue angle of about  $20^\circ$  in CIELAB (see Figure 4.1), we can roughly estimate that  $360^\circ/20^\circ = 18$  targets would be necessary to correct all hues with a similar accuracy. However, it is not practical for users to choose among that many targets. We thus decided to employ nine targets covering all hues, at the cost of a lesser accuracy. We assume that high accuracy in color correction is more critical for skin tones than for the present application.

It must be noted that we consider *object colors*, i.e., colors stimuli arising from the reflection of incident radiant power by objects (Wyszecki and Stiles 2000). For a given spectral power distribution, we can compute, e.g., CIE XYZ tristimulus values for all possible reflectance spectra. The points taken for all object color stimuli define an *object color solid* (Wyszecki and Stiles 2000), whose shape and volume depend on the SPD. These solids are subsets of the CIE XYZ color space, i.e., we are not actually considering “any” color stimuli, but the more limited range of actual object colors. However, we will encounter a variety of object textures and shapes, which may modify the surface’s color appearance. Additionally, we may come across dyed objects, whose reflectances can be non-smooth, which causes metamerism issues, or may even be fluorescent (absorbing light at a certain wavelength and re-emitting it at another wavelength). The variety of textures and materials can be a source of error.

The system returns a set of paint colors coordinated with the one of the imaged object. The paint database consists of colors grouped by palettes of four (see Figure 4.3). We do not need exact color matching, rather, we are interested in selecting the closest shade match in a color palette. The palette is the basis of our color coordination recommendation. In other words, we are not *creating* a palette complementing the object’s color, but are looking for the best match in an existing database of coordinated paint colors created independently by an artist or an interior designer. The actual composition of the paint database and the choice of color palettes are not relevant *per se*; the database can be replaced by any other color palettes collection.

The results were graded by a Home Décor expert, hereafter referred as “the expert.” Our metric of success is the agreement between the algorithm and





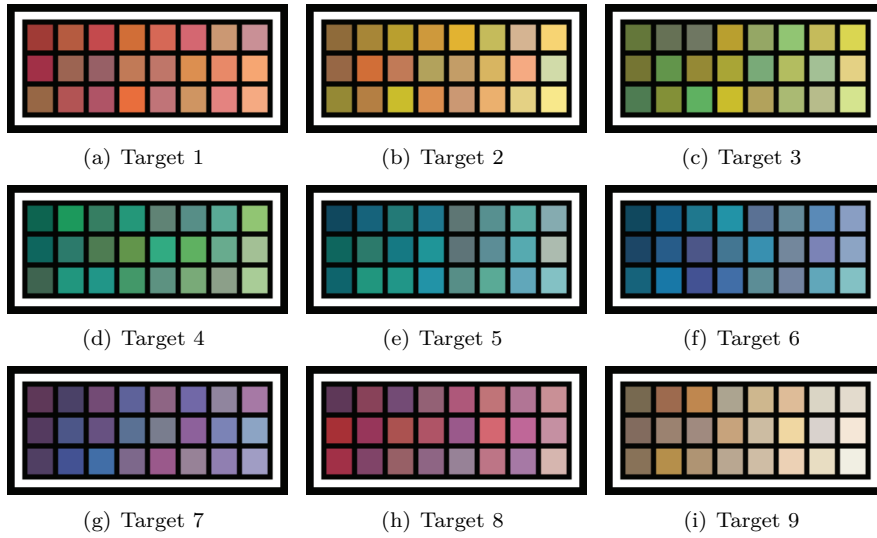
**Figure 4.1:** Skin tones only represent a limited color range and span a hue angle of about  $20^\circ$  in CIELAB. The surfaces represent the color values of the target used for skin color correction covering the range of human skin tones in sRGB and CIELAB.

the expert in the choice of a corresponding palette. The quality of the color assessment is based on the expert’s opinion and our metric is optimized after these results. That is, we are testing if the advice can give solid guidelines to users in real conditions such as what colors to add to a room while keeping an existing design element. The system has to work with a relatively low number of calibration targets and under many real conditions, such as limited palettes and non-uniformity of the samples.

In the application presented in Chapter 3, the target patches were extracted automatically. More precisely, they were detected as regions whose contours correspond to the zero crossing of a second order derivative of the image (Harville *et al.* 2005). The same extraction method can be applied to the present approach; however, it has not yet been implemented and the object and patch pixels were extracted manually.

### 4.2.1 Reference targets

Eight targets consist of a selection of Munsell Colors (Munsell 1905) covering a hue angle of roughly  $60^\circ$ . A target contains 24 patches, 21 come from 7 secondary hues distributed to cover a good range of chroma and lightness, and three are paints extracted from our database. The range of colors of two adjacent targets overlap. Targets were ordered by similar hues to facilitate their use. Hue is the most natural attribute of color, it makes the choice of the target by visual matching much simpler for users. Also, overlapping hues avoid having an object whose color may be in-between charts. We added a ninth chart made of 24 paint samples from the paint database covering a variety of beiges and browns, colors that are common in Home Décor and may require a finer sampling. Figure 4.2 shows the color correction targets. The target reference values were computed as reported in Section 3.3.1.



**Figure 4.2:** The color correction uses 8 targets covering all hues, plus a target consisting of brown and beige paint samples.

### 4.2.2 Color transform

The color correction follows the same computation as presented in Section 3.3.2. A linear transform is computed by comparing image extracted patch values with reference ones and by minimizing a least square error in nonlinear sRGB. The only difference is that users must choose one of the nine color charts. They image the object and chart together, the target patches are extracted, their color pixel values are averaged and compared to reference values given by sRGB triplets.

### 4.2.3 Matching metric

In order to find colors complementing the object, we first need to find which paint in the database constitutes its best color match. An immediate way of matching these colors is to compare the sRGB values of the 252 paint samples in the database with the corrected image object value and minimize a distance in sRGB

$$\Delta sRGB = \sqrt{(R_o - R_p)^2 + (G_o - G_p)^2 + (B_o - B_p)^2}, \quad (4.1)$$

where ‘*o*’ stands for ‘object’ and ‘*p*’ for ‘paint’. For each object from our set of decoration items samples (see Appendix E), we retrieved the paint minimizing this distance and visually compared the objects with the paints returned by the system. The results were poor, which can be explained by the lack of perceptual uniformity of the sRGB color space. The range of colors considered in the Home Décor application is not constrained to a small range of colors, such as skin tones, and the minimization of a distance in sRGB is not adapted. The color matching must be done in a perceptually uniform color space, such

as CIELAB or CIECAM02. We chose CIECAM02 over CIELAB for its better perceptual uniformity. Moreover, CIELAB’s lack of uniformity in bluish hues (Moroney 2000) may introduce errors when considering blue or purple samples. The object and paint samples are thus matched by minimizing their color difference in CIECAM02. The object pixel values are converted from sRGB to CIE XYZ and then to CIECAM02 (see Section 2.4.4). Input data for CIECAM02 includes the tristimulus values CIE XYZ of the object and of the white point  $X_w Y_w Z_w$ , the adapting luminance  $L_A$ , and the relative luminance of the surround. Viewing conditions parameters are chosen as advised for the previous model CIECAM97s when considering sRGB (Moroney 2000), i.e., the white-point is D65, the adaptive luminance  $L_A = 4 \text{ cd/m}^2$ , and the relative luminance of the surround is “average.”

The Euclidian distance between the object’s and paint’s CIECAM02 values

$$\Delta E_{02} = \sqrt{(J_o - J_p)^2 + 100(a_o - a_p)^2 + 100(b_o - b_p)^2}, \quad (4.2)$$

where ‘ $o$ ’ stands for ‘object’ and ‘ $p$ ’ for ‘paint’, is minimized.  $J$  represents the CIECAM02 lightness and  $a$  and  $b$  are the *red – green* and *yellow – blue* components, respectively. These components were chosen for computational simplicity over the hue  $h$  and chroma  $C$ .  $a$  and  $b$  are multiplied by a factor 100 to adjust them to the range of lightness  $J$ . Using the actual cartesian coordinates  $(a_c, b_c)$  (2.14 and 2.15) would probably have given slightly better results, as we will see in Section 4.3.4. Color differences will be expressed as  $\Delta E_{02}$  when the metric (4.2) used by our system is considered, but will also be expressed if needed as  $\Delta E_{02}^c$  (2.16) for a better perceptual estimation or  $\Delta E_{ab}^*$  (2.13) for comparison with the results from Chapter 3.  $\Delta E_{02}$ ,  $\Delta E_{02}^c$ , and  $\Delta E_{ab}^*$  values are generally very close.

#### 4.2.4 The algorithm

The algorithm consists of the following steps:

1. Users choose a target whose hue is similar to the hue of the object. They image the object and target together.
2. The target patches are extracted. The color correction transform  $\mathbf{A}$  (3.3) mapping the image target values  $\mathbf{M}$  onto reference values  $\mathbf{T}$  is computed and applied to the entire image.
3. The object pixels values are extracted, averaged, and converted to CIECAM02.
4. The object color is compared against a collection of 63 coordinated paint palettes, each consisting of four coordinated colors, i.e., a total of 252 paints. The system chooses the paint closest to the color of the object in CIECAM02 (Eq. 4.2).
5. The system finds the palette containing the best match and the three paints complementing its color, hence complementing the object (see Figure 4.3).



**Figure 4.3:** Paint samples are grouped in palettes of four colors. The system matches the object color to one of the paints and returns the corresponding palette. The figure shows, from left to right, the uncorrected image, the corrected image, and the corresponding paint palette.

### 4.3 Experiments and results

Sixty-three decoration objects were imaged with three cameras, a HP R-967 camera (10 megapixels) and two 2 megapixels cell phone integrated cameras, a Nokia 7310 and a Sony W580i, under uniform fluorescent light along with a properly chosen color calibration target. Images of the sixty-three samples can be seen in Appendix E, as the corrected images and the resulting color palettes. Fluorescent lighting conditions are common in office environments and stores, as well as in homes with the increasing use of energy-saving light bulbs (the European Union, Canada, the United States of America, and many other countries have agreed on banning the use of standard incandescent light bulbs by 2012). Despite fluorescent light sources having peaky spectral power distributions and creating more metamerism problems, they will progressively become the main artificial light source and were thus considered in our application.

The object database consists of 63 samples of various colors and materials. The samples were chosen such that their colors cover most hues. The choice of beige, brown, and wood-like samples is larger (26 samples) as these colors are very common in Home Décor. The samples are: wood (7 samples), linoleum (8 samples, including some mimicking wood), tiles (10 samples, including two semi-transparent glass tiles), kitchen tops (6 samples), and fabrics (32 samples). We intentionally chose different, textured, non uniformly colored, and non Lambertian samples to test the system in “real” conditions. However, the database only contains flat and relatively smooth samples, our method may thus have to be adapted depending on the geometry of the considered decoration object. The database and the targets were created independently, i.e., we did not use the targets as references while collecting the samples.

The object CIECAM02 values are compared against the CIECAM02 values of the paint samples by minimizing the Euclidian distance (4.2). The color database consists of 252 paint samples from a paint manufacturer grouped in 63 palettes of four colors. The system picks the paint closest to the object among the 252 paint samples and returns the palette containing the one matching the object's color. The system thus returns four colors, one matching and three complementing the object. Examples of color palettes can be seen in Figure 4.3.

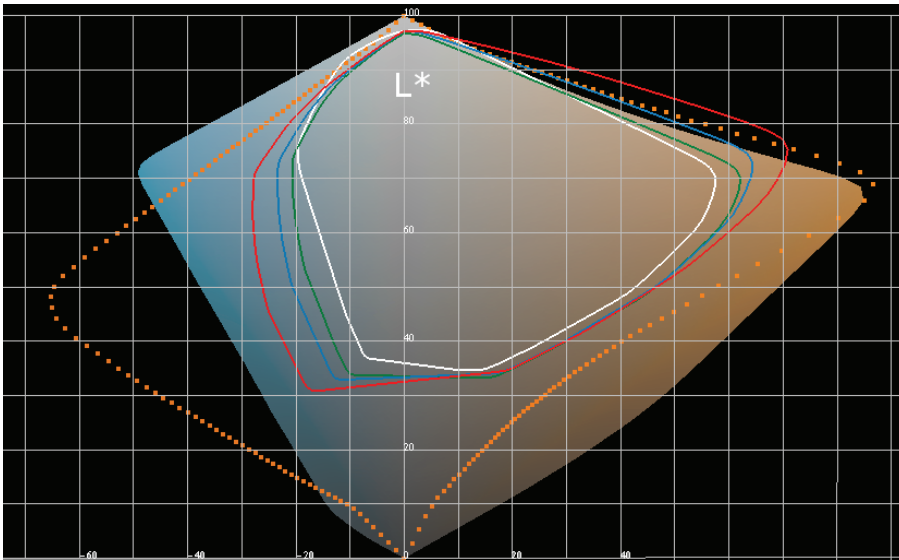
We determine if light colors risk being clipped in sRGB in Section 4.3.1, characterize the color correction accuracy in Sections 4.3.2 and 4.3.3, and discuss how the results correlate with the expert's opinion in Section 4.3.4.

### 4.3.1 sRGB gamut size and white-balance

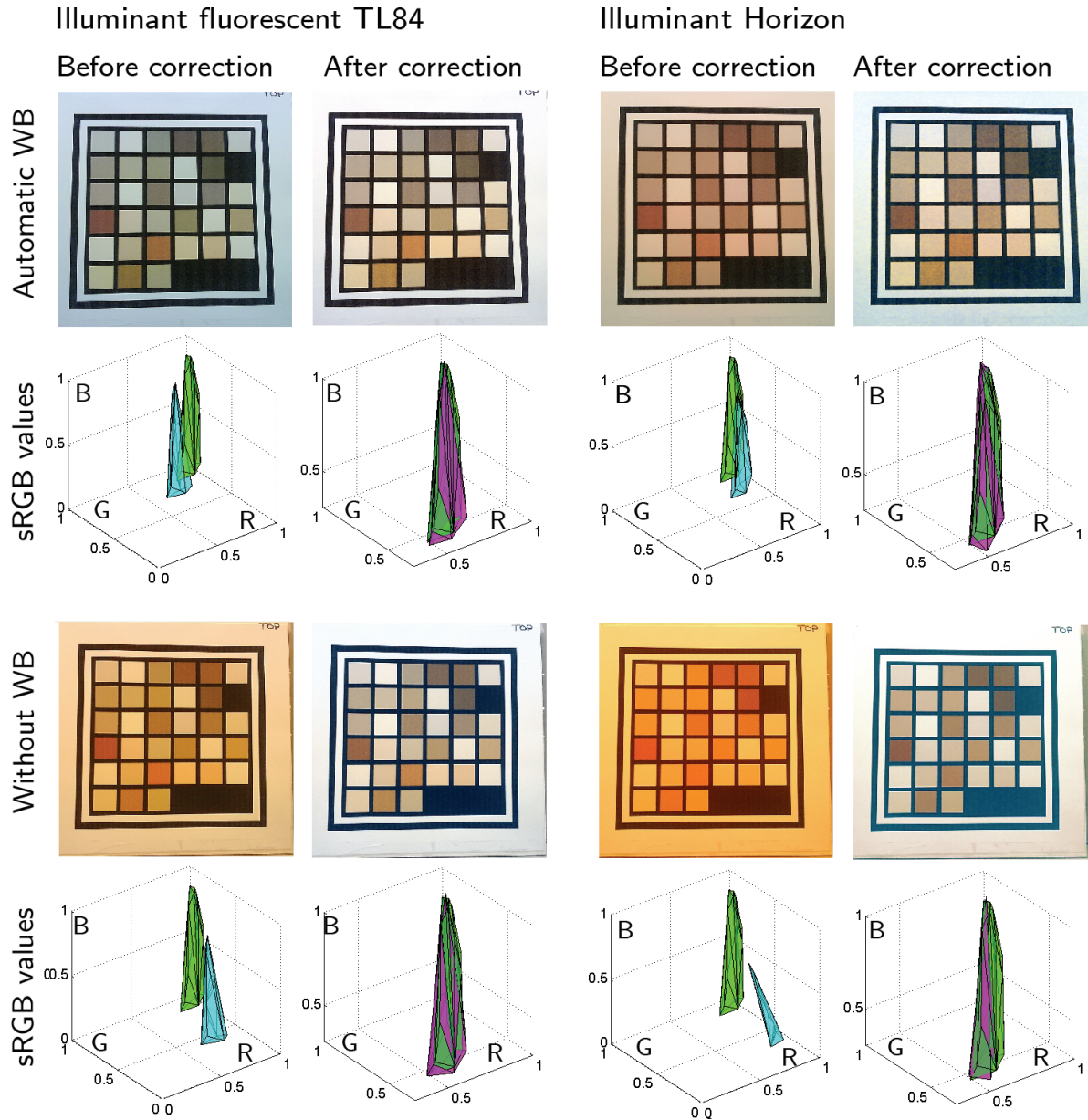
We have seen in Section 2.4.2 that the sRGB color space's main disadvantage is the small size of its gamut. Light colors and variations of beige are common in decoration and are precisely the colors most likely to be clipped in sRGB.

We measured the reflectance spectra of all the paint samples in the database using an Eye-One Pro spectrometer with  $45^\circ/0^\circ$  measurement geometry, used them to generate CIELAB values (Section 2.4.3) as viewed under several illuminants, and represented them along with the sRGB gamut in a 3D plot. Figure 4.4 shows the gamuts of sRGB (colored surface) and the gamuts of all paint samples under the illuminants incandescent A (red curve), daylight D50 (blue curve), daylight D65 (green curve), and fluorescent F7 (white curve). The orange dotted line represents the gamut of the printer used to produce the color correction targets. CIELAB values for beige colors under incandescent illuminant A lie outside of the sRGB gamut. In order to estimate if this is an issue in practice, we conducted the following experiment: we built a target using the brown and beige paint samples reflectance spectra and imaged it with two cameras, the HP R-967 and the Sony W580i, under several light booth illuminants (daylight D65, cool white, Horizon, fluorescent TL84, and incandescent A). A set of images was taken in automatic white-balancing mode and another set was taken under two fixed white-balancing modes, named "sunny" and "daylight" for the HP and Sony cameras, respectively. We assumed that these modes compensate for an illuminant close to D65, which is the defined illuminant of sRGB. We thus considered that these modes performed no or little illuminant compensation and refer to them as "no white-balancing." We can visualize this color correction by looking at the convex hulls of the reference, image extracted, and color corrected target values in normalized sRGB (Figure 3.7). Figures 4.5 and 4.6 show the resulting images for each respective camera and two illuminants, the fluorescent TL84 and "Horizon" matching a reddish sunset light. The top images are obtained using automatic white-balancing.

When no white-balancing is applied, most of the patches imaged with the Sony cell phone camera are clipped in the red channel and the red information cannot be recovered (see Figure 4.5). The reference values (green surface) and the image corrected ones (magenta) do not overlap and present large deviations. The error has less of an impact when using the HP camera, but the phenomenon is still present (see Figure 4.6). Table 4.1 reports the corresponding color correction errors in  $(Y, r, g)$ . We see that the use of automatic

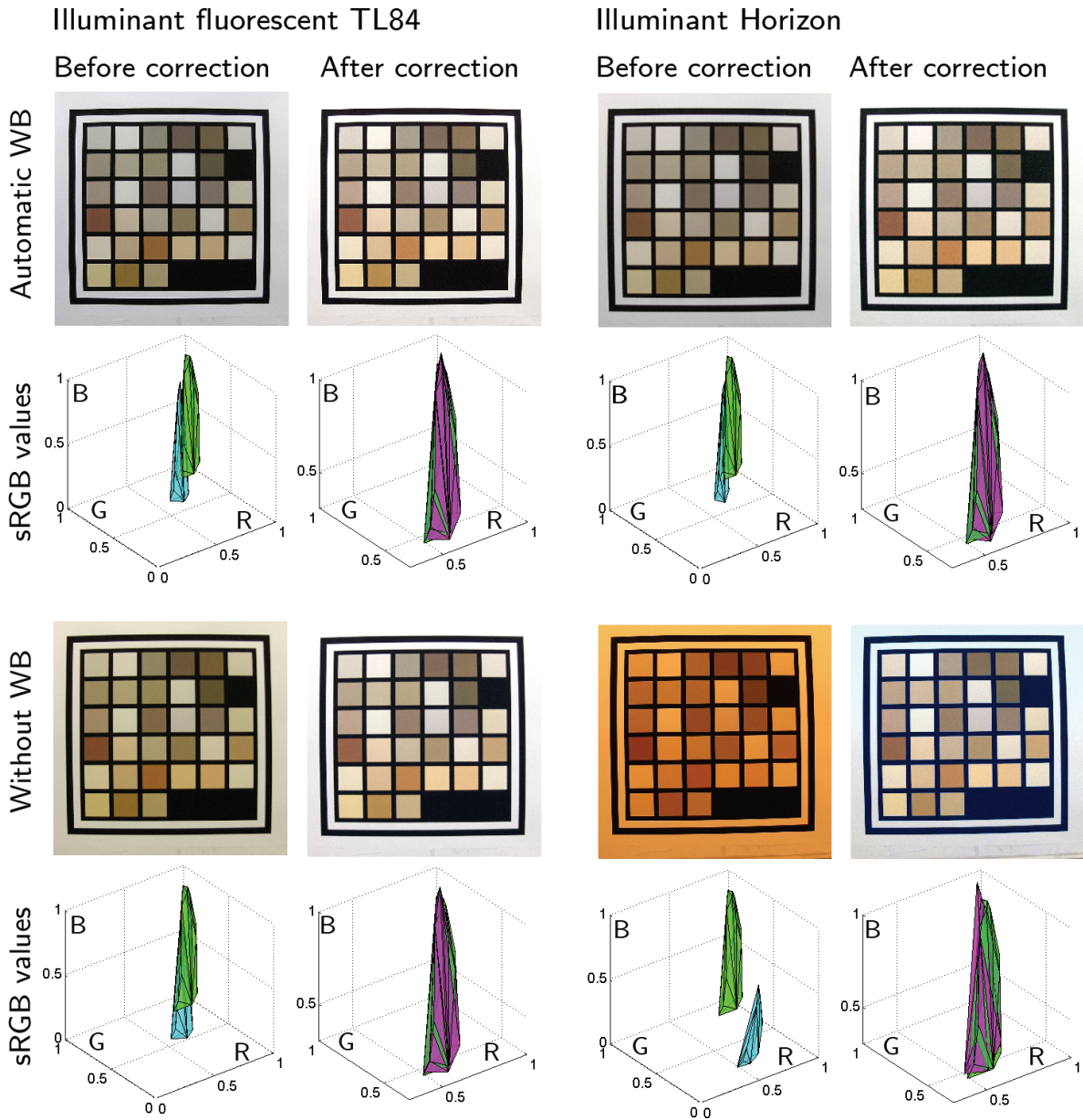


**Figure 4.4:** Color values of beige paints viewed under a reddish illuminant (red curve) fall outside of sRGB gamut and may not be correctly represented in rendered images. The colored surface represents sRGB gamut in CIELAB. The solid curves represent CIELAB values generated from paint samples reflectance spectra under four illuminants: incandescent A (solid red), daylight D50 (solid blue) and D65 (solid green), and fluorescent F7 (solid white). The orange dotted line represents the gamut of the printer used to produce the correction targets.



**Figure 4.5:** Will beige be clipped because of the relatively small size of sRGB gamut? A set of beige values were imaged with the Sony W580i camera under illuminants TL84 and the reddish “Horizon”. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). We see that beige colors are clipped under a reddish illuminant and cannot be recovered after correction. Automatic white-balancing avoids this issue.





**Figure 4.6:** Will beige be clipped because of the relatively small size of sRGB gamut? A set of beige values were imaged with the HP R-967 camera under illuminant TL84 and the reddish “Horizon”. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). We see that beige colors are clipped under a reddish illuminant and cannot be recovered after correction. Automatic white-balancing avoids this issue.



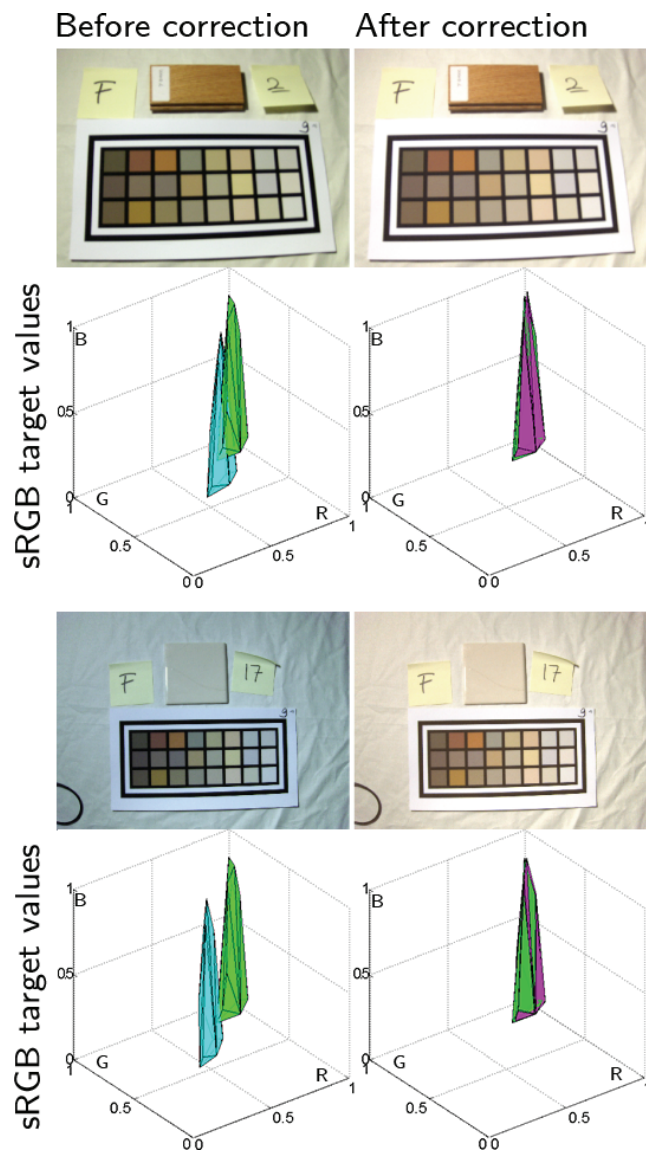
		TL84	Horizon			TL84	Horizon
		$\Delta Y =  Y_1 - Y_2 $				$\Delta g =  g_1 - g_2 $	
HP	auto WB	0.015	0.018	auto WB	0.002	0.002	
	no WB	0.015	0.020	no WB	0.002	0.004	
Sony	auto WB	0.015	0.017	auto WB	0.004	0.004	
	no WB	0.012	0.016	no WB	0.004	0.005	
		$\Delta r =  r_1 - r_2 $				$\Delta sRGB = \sqrt{\Delta Y + \Delta r + \Delta g}$	
HP	auto WB	0.002	0.002	auto WB	0.015	0.019	
	no WB	0.002	0.006	no WB	0.015	0.022	
Sony	auto WB	0.005	0.007	auto WB	0.016	0.018	
	no WB	0.005	0.015	no WB	0.014	0.022	

**Table 4.1:** Color correction accuracy of brown and beige colors under two illuminants: fluorescent TL84 and the reddish “Horizon.” The error  $\Delta r$  is larger when the samples are imaged under the illuminant “Horizon” without white-balance, but not when automatic white-balance is used. It indicates that clipping of beige colors should not an issue whenever consumers use their camera in fully automatic mode.

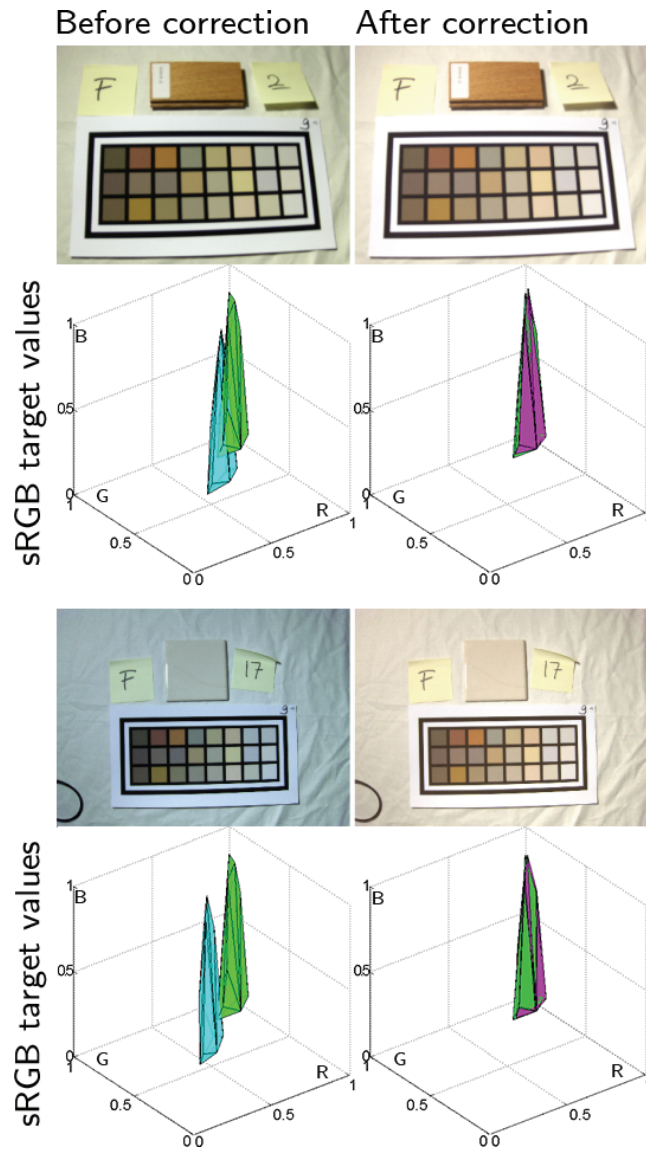
white-balancing does not have an influence on the final correction error amplitude under illuminant TL84, but has an important influence for the illuminant “Horizon,” especially on the  $r$  component. That is, clipping seems to be an issue under reddish illuminants. On the other hand, this problem does not occur in our two examples when automatic white-balancing is used. White-balancing is not complete, yet it allows all reference values to be correctly recovered after color correction. Consequently, we can assume that if a consumer uses the camera in fully automatic mode, even under a reddish light, clipping of beige colors should not be an issue. White-balancing actually happens early in the in-camera processing pipeline, while sRGB encoding happens rather at the end of it (see Section 2.3). This would explain our observations, with illuminant compensation being performed in a “camera raw” color space having a larger gamut than sRGB and taking place prior to clipping.

### 4.3.2 Color correction accuracy

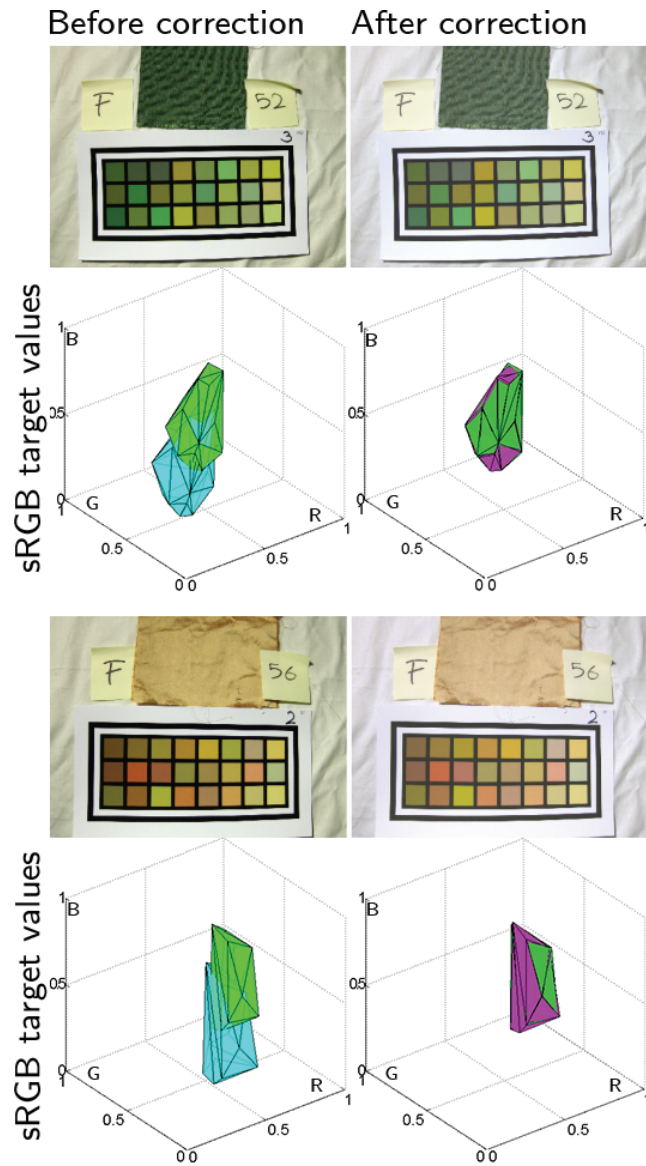
In this section, we estimate the color correction accuracy using the 63 images taken with the HP R-967. Figure 4.7 shows six samples before and after color correction and the corresponding target values in normalized sRGB. The accuracy of the color correction cannot be estimated using the samples reflectance spectra due to the variety of non Lambertian and textured materials in the set. However, it can be estimated using the target patches, which are known, and a leave-one-out method, such as presented in Section 3.4.1. Each target patch is successively corrected using a color correction transform computed from the 23 remaining patches. The error in color correction is computed as the difference between the patch image extracted normalized sRGB values after color correction and the corresponding reference ones. This is done for the 24 patches extracted from the 63 images, i.e., a total of 1512 patches. The average error



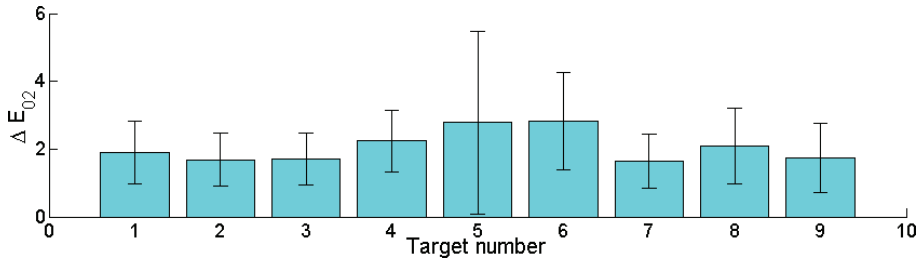
**Figure 4.7:** Examples of color correction. All images were taken with the HP R-967 camera. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.



**Figure 4.7:** Examples of color correction. All images were taken with the HP R-967 camera. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.



**Figure 4.7:** Examples of color correction. All images were taken with the HP R-967 camera. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.



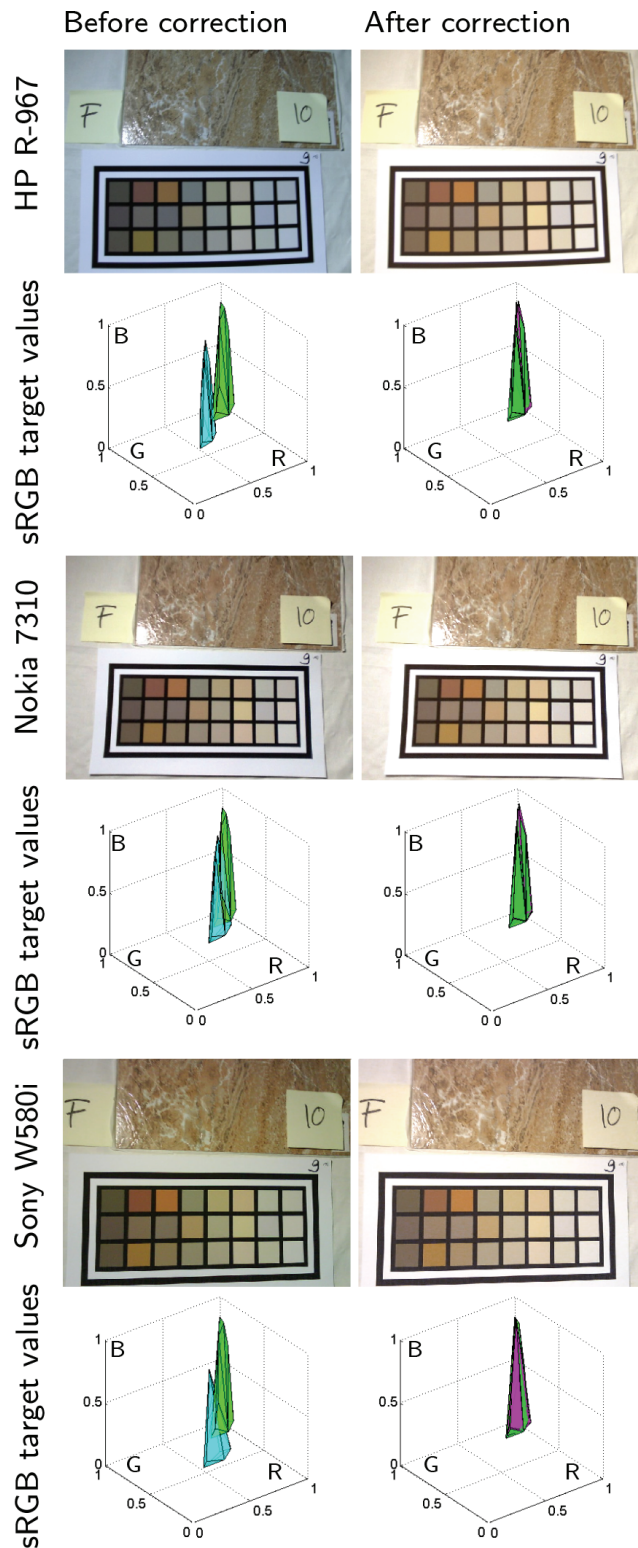
**Figure 4.8:**  $\Delta E_{02}$  (4.2) error amplitude in color correction for each of the 9 targets. The numbering refers to Figure 4.2. The results are obtained using a leave-one-out approach on the target patches. The bars show the standard deviation  $\sigma$ .

in normalized sRGB target values is 2.02% ( $\sigma = 1.02\%$ ). The equivalent errors in CIECAM02 (4.2) is  $\Delta E_{02} = 2.19$  ( $\sigma = 1.17$ ). Figure 4.8 shows the color correction errors  $\Delta E_{02}$  for each of the nine targets. The error is target dependent. It can also be expressed as  $\Delta E_{02}^c = 2.02$  ( $\sigma = 1.09$ ) or  $\Delta E_{ab}^* = 1.98$  ( $\sigma = 1.02$ ). The equivalent error for the similar estimation of the color correction accuracy for skin color assessment (see Section 3.4.1) was  $\Delta E_{ab}^* < 1$  ( $\sigma < 1$ ). The error in color correction is larger than the error obtained in Chapter 3 when correcting skin tones, which is unsurprising given that each target covers a hue angle of about  $60^\circ$  as opposed to about  $20^\circ$  previously.

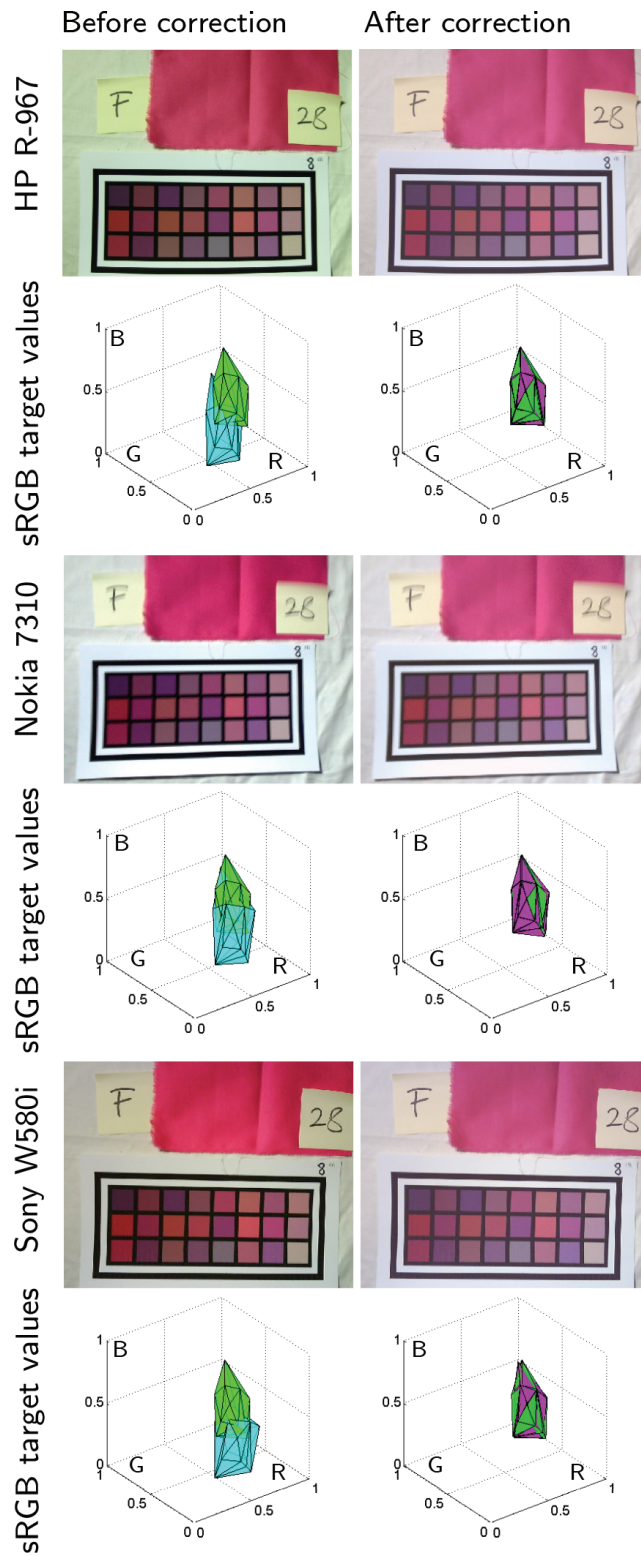
### 4.3.3 Color correction across cameras

The color correction accuracy estimated on the target patches remains within 2% in sRGB, but real decoration samples are not as smooth and uniform as the target patches used to compute the correction errors. Figure 4.9 shows a total of four samples imaged with the three cameras HP R-967, Nokia 7310, and Sony W580i under fluorescent illuminant. Variations in color are strongly reduced after color correction. The figures also show the gamuts of the corresponding target values in sRGB. The reference patch values extracted from the images before correction (cyan surfaces) differ from the reference values (green), but overlap after color correction (magenta) for all 12 examples. We assume that the color of the object of interest lies in or close to the gamut of the corresponding correction target and is accurately corrected, which is the case for about two thirds of the samples, as we will verify shortly.

The resulting object color needs to be independent of the camera. We estimate the color correction accuracy across cameras by comparing the luma values  $Y = (R + G + B)/3$  and the normalized color coordinates  $r = R/(R + G + B)$  and  $g = G/(R + G + B)$  of the object colors before and after correction. Figure 4.10 shows the luma  $Y$  and the normalized chromaticity coordinates  $(r, g)$  extracted from the images of the 63 samples captured with the three cameras. Values are displayed for each pair of cameras. A black dotted line indicates the identity relation. This figure should be compared with Figures 3.13 and 3.14. The correlation is high across the three pairs of cameras. Table 4.2 reports the correlation coefficients for the corresponding  $(Y, r, g)$  values and the equivalent CIECAM02  $\Delta E_{02}^c$  color differences averaged over all images. Correlation coefficients range from 0.97 to 0.98 for  $Y$ , from 0.96 to 0.98 for  $r$ ,

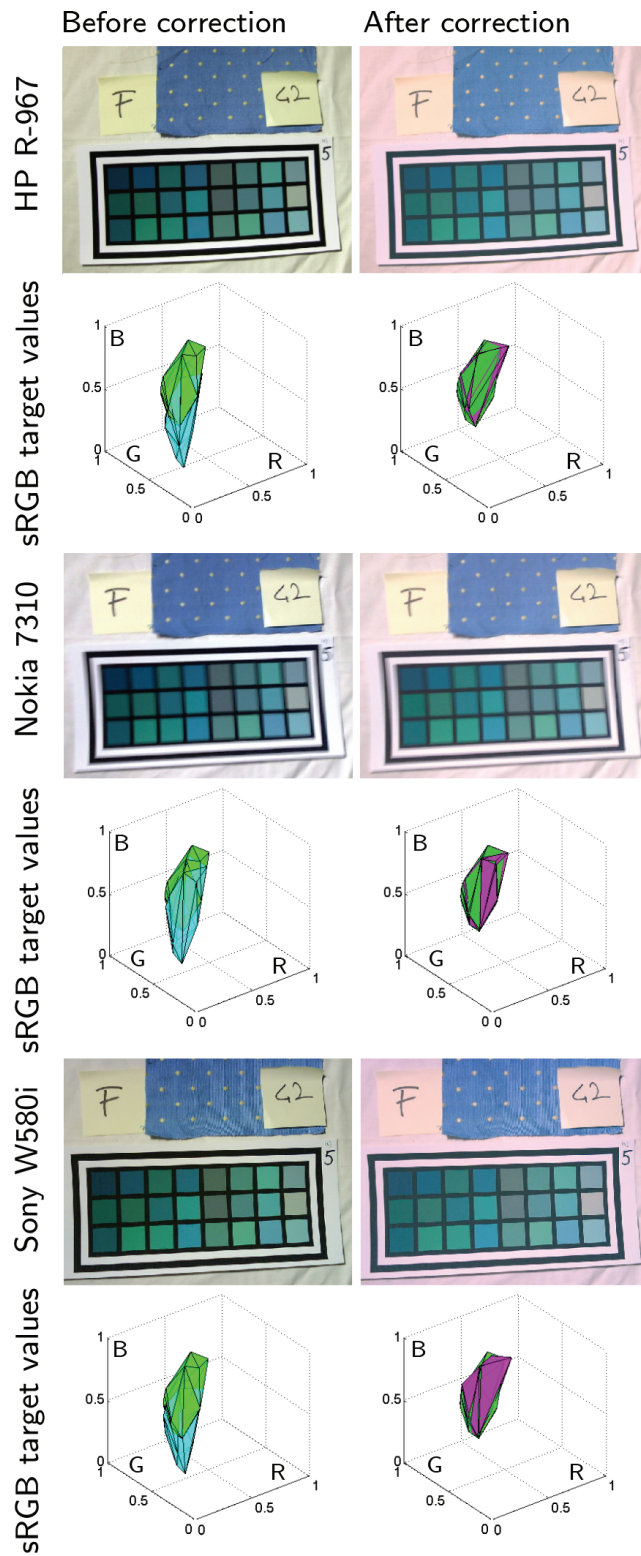


**Figure 4.9:** Cross-camera correction. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.



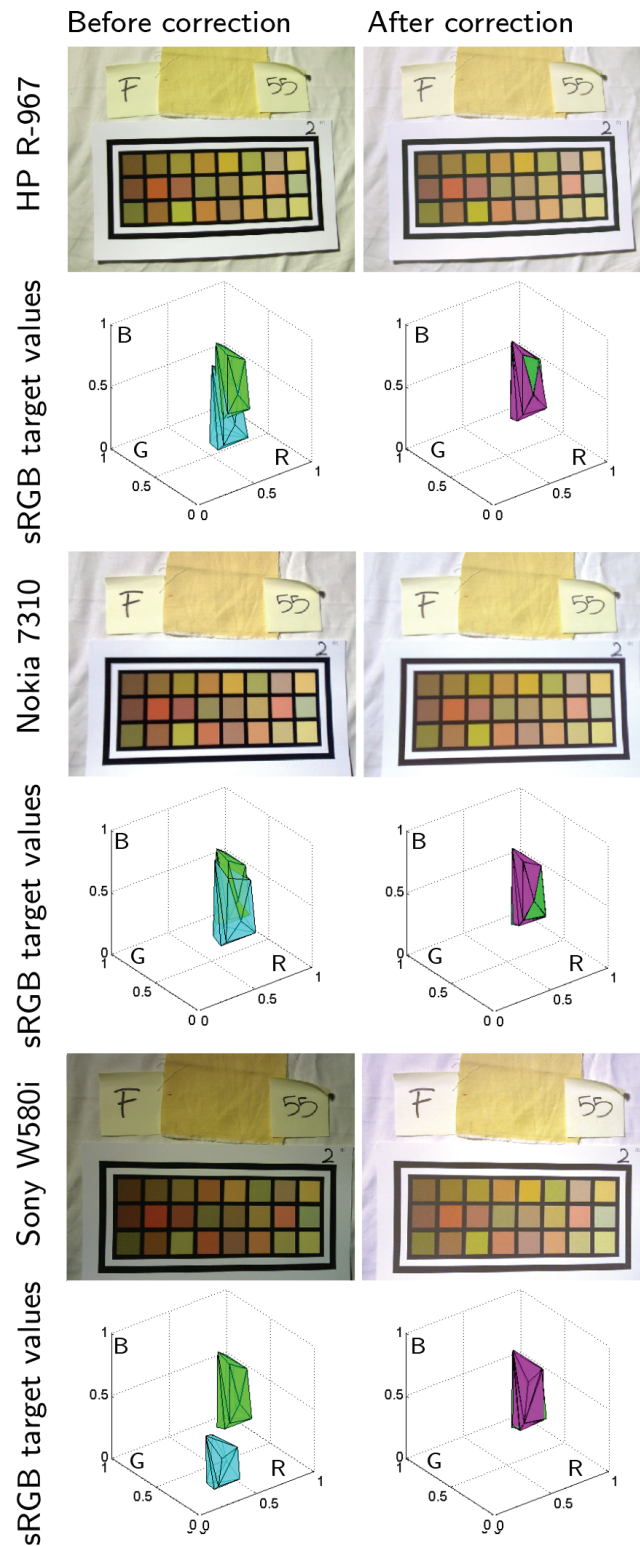
**Figure 4.9:** Cross-camera correction. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.





**Figure 4.9:** Cross-camera correction. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.





**Figure 4.9:** Cross-camera correction. Under each pair of images, we show the convex hulls of the image uncorrected target values (cyan), reference target values (green), and image corrected target values (magenta). The corrected values and reference values overlap.

	HP	Nokia	Sony	HP	Nokia	Sony
	Y			g		
HP	1			1		
Nokia	0.98	1		0.93	1	
Sony	0.97	0.97	1	0.94	0.93	1
	r			$\Delta E_{02}^c$		
HP	1			0		
Nokia	0.97	1		3.49	0	
Sony	0.96	0.98	1	5.22	6.18	0

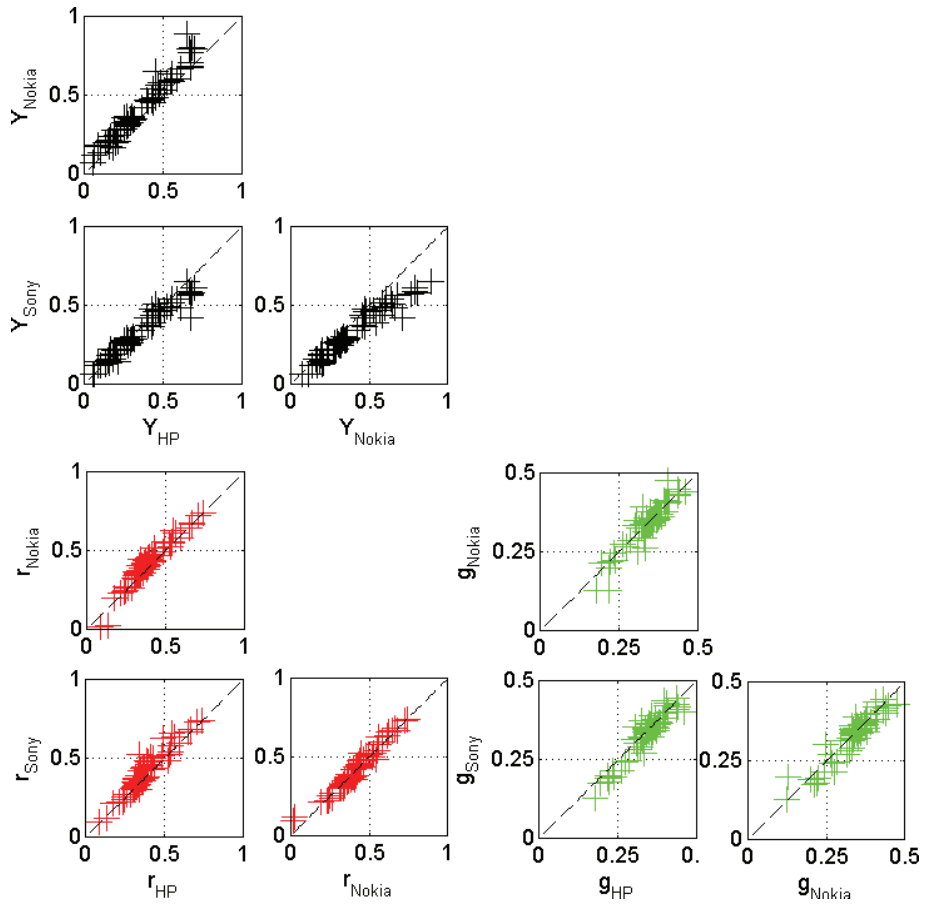
**Table 4.2:** Correlation coefficients between the  $(Y, r, g)$  values of extracted object color and corresponding average  $\Delta E_{02}$  for each pair of cameras.

	HP	Nokia	Sony	HP	Nokia	Sony
	$\Delta Y =  Y_1 - Y_2 $			$\Delta g =  g_1 - g_2 $		
HP	0			0		
Nokia	0.058	0		0.017	0	
Sony	0.037	0.078	0	0.021	0.020	0
	$\Delta r =  r_1 - r_2 $			$\Delta sRGB = \sqrt{\Delta Y^2 + \Delta r^2 + \Delta g^2}$		
HP	0			0		
Nokia	0.033	0		0.13	0	
Sony	0.034	0.022	0	0.10	0.15	0

**Table 4.3:** Average error between the  $(Y, r, g)$  and sRGB values of extracted object color each pair of cameras.

and from 0.93 to 0.94 for  $g$ . The average absolute difference in normalized luma  $\Delta Y = |Y_1 - Y_2|$  ranges from 3.7% to 7.8%, while the normalized chromaticity coordinate  $\Delta r = |r_1 - r_2|$  ranges from 2.2% to 3.4% and  $\Delta g = |g_1 - g_2|$  ranges from 1.7% to 2.1%. The error in normalized sRGB  $\|sRGB_1 - sRGB_2\|_2$  ranges from 0.10 to 0.15 (see Table 4.3).

This color correction error is about three times larger than the cross-camera error obtained in the skin color assessment experiments (Section 3.4.4). We have indeed seen that the angle spanned by skin tones in CIELAB is about  $20^\circ$  and we used only nine targets to cover for all  $360^\circ$  hues angles. Moreover, the variety of samples of different materials and textures contributes to the color correction error. The texture of skin across the images of the experiment reported in Chapter 3 is by far more uniform than the many textures available in our samples database. Furthermore, for the few samples having two, or more, colors, such as dotted fabrics, the system took the average color, but we asked the expert to only consider the dominant color. When building the samples database, we intended to produce a “worst case scenario” to test how our method would perform in real conditions. The decoration samples were chosen to exhibit a large variety of colors and textures. Clipping of light colors



**Figure 4.10:** Luma  $Y = (R + G + B)/3$  and normalized color coordinates  $r = R/(R + G + B)$  and  $g = G/(R + G + B)$  compared for each pair of cameras. The x-axis shows (from left to right) cameras HP R-967 and Nokia 7310 and the y-axis shows (from top to bottom) cameras Nokia 7310 and Sony W580i. The black dotted line indicates the identity relation.

also contributes to the error; we see in Figure 4.10 that the luma component  $Y$  shows an important deviation for values close to 1, due to the clipping of light colors in images taken with the Nokia 7310 camera. The average object color differences  $\Delta E_{02}^c$  across cameras remain reasonable and range from 3.49 to 6.18 (Table 4.2).

The color correction of an object is considered good if its color is similar to the colors in the calibration target. In other words, the color correction accuracy depends whether the object color is located “close enough” to the volume formed by the target values in sRGB. Importantly, the target gamuts do not fill the entire sRGB gamut, i.e., the targets are not optimal for all the samples. We tested which samples are most likely to be accurately color corrected by looking at the position of the object color in sRGB with respect to the convex hull of the corresponding calibration target, as already presented in Section 3.4.1. We used as criterion the difference in volume between the convex hulls of the target points with the object color point  $V_{t+o}$  and of the target points alone  $V_t$ . The color correction is considered as sufficient if the difference in volume is less than 10%, i.e., if

$$\frac{V_{t+o} - V_t}{V_t} < 0.1. \quad (4.3)$$

38 samples satisfy the criterion. Figures 4.10 and 4.11 take all 63 samples into account, while Figures 4.12 to 4.14 use the 38 samples satisfying (4.3).

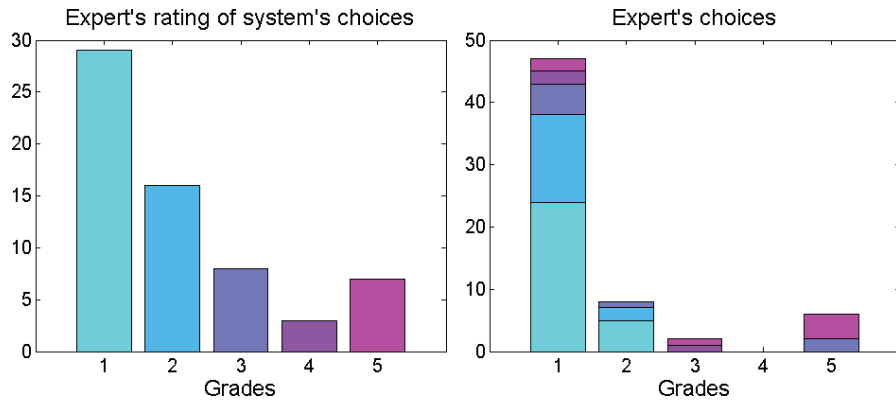
This criterion allows deciding whether the calibration target is adapted for the correction of a given object. This could be used to detect failing cases, as to guide users in the choice of the target.

#### 4.3.4 Color correction and expert’s rating

The expert was shown each sample along with 9 palettes of four colors each, i.e., a total of 36 paints, chosen to be of similar colors as the object. She was first asked to choose a paint out of the 36 that she would pick as the best match for each of the 63 samples. The database of paints did not offer enough colors for her to systematically pick an optimal color. She was thus asked to rate her own choices. The grades go from 1 to 5 according to the following rating:

1. The sophistication of a high quality expert recommendation, perfect given the palette selections
2. Competent work by an expert given the palette selections
3. Close, but not perfect, typical of an untrained consumer
4. Poor selection, other selections are much better
5. Terrible, unacceptable, even for a consumer.

The expert was asked to give integer grades and keep uniform intervals between them. After she made her choice, we presented her the system’s results and asked her to rate them as previously. The system’s and expert’s results were then compared and analyzed. We did not ask the expert to rate the quality of the color palettes picked by the system, but rather to rate the quality of the match.

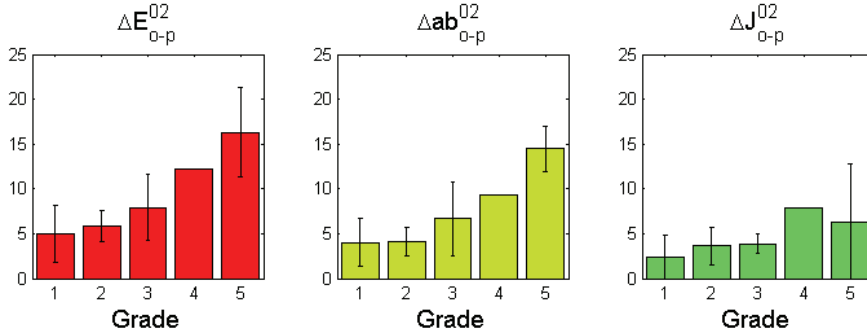


**Figure 4.11:** This figure shows the occurrence of each rating, for the system (left) and expert (right) choices. The colors in the right plot correspond to the grades of the system's choices, i.e., it shows the proportion of samples whose matches were similarly rated.

The expert gave the grade 1, meaning “perfect,” to 29 of the 63 paints selected by our system, while she similarly rated 47 of her own choices. The grades occurrences can be seen in Figure 4.11. The system and expert top choices match for one third of the objects, but the remaining results are also good. Many different paint patches have very similar CIECAM02 values, especially in beige tones. The expert graded most of the system results 1 - as perfect - or 2 - good for an expert. The average grades are 2.09 ( $\sigma = 1.34$ ) and 1.57 ( $\sigma = 1.21$ ) for the system and expert, respectively. These results are satisfying, but do not take the quality of the color correction into account. If we consider only the grades corresponding to the 38 samples satisfying the criterion (4.3), the distribution of grades is as follows: 18 samples receive a 1, 12 samples a 2, 3 samples a 3, 1 sample a 4, and 4 samples a 5. The remaining 25 samples will be ignored in the following sections for a more precise analysis.

The goal of this experiment is to determine if we can give automated color advice that an expert would consider as good. The two important factors are the quality of the color correction and the metric used to match the objects with the paint samples. We have seen that, when the target is correctly chosen according to the object's color, the color correction is good with an accuracy of a few percents in sRGB. We now need to determine whether our metric (4.2) is appropriate.

We assumed that we can mimic how an expert matches a sample with paint colors by minimizing the CIECAM02 distance between them. We verify the validity of this assumption by comparing the CIECAM02 distance (4.2) used by the system to assign matches with the grades the expert gave to the results. Figure 4.12 shows clearly that the paints that are close to the object in CIECAM02 ( $\Delta E_{02} \simeq 5$ ) get the best grade, i.e., what the expert considers a good match corresponds to a small  $\Delta E_{02}$ . Figure 4.12 also shows that the difference in  $(a, b)$  is more critical than the difference in lightness  $J$ . Figure 4.13 shows the same CIECAM02 differences as Figure 4.12,  $\Delta E_{02}$  distances being represented on a polar plot. The radius, i.e., the distance between a cross and



**Figure 4.12:**  $\Delta E_{02}$  differences between the object and paints colors as a function of the grades given by the expert computed for the 38 samples satisfying Equation (4.3).  $\Delta ab$  stands for  $\sqrt{\Delta a^2 + \Delta b^2}$ . The bars show the standard deviation  $\sigma$ . Only one sample was graded as 4.

the origin, corresponds to the CIECAM02 difference between the object and the selected matching paint. The polar angle corresponds to the object hue. The colored lines correspond to the 252 paints in the database, also ordered by their hue angle. This alternative representation of the results shows how the hue of the objects and paints are distributed. It also shows that the grades 4 and 5 (magenta and red crosses) mostly correspond to object hue angles for which the paint density in the database is low. We can thus assume that a denser database would have allowed finding satisfying matching paints for all samples.

#### A modified metric for better results

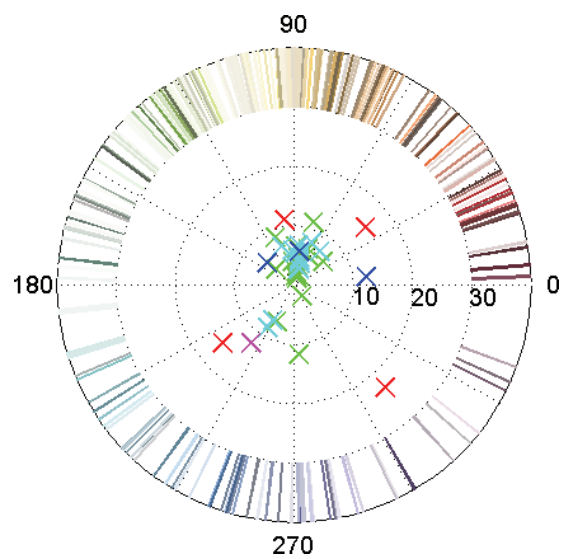
The above results suggest that hue plays a role more important to the expert than lightness when choosing color matches. The current metric (4.2) can be modified to give less weight to the lightness. Moreover, the CIECAM02 model defines cartesian coordinates (Eq. 2.14 and 2.15) that are better perceptual attributes than the approximated cartesian coordinates  $100a$  and  $100b$  used in (4.2). We tested a new metric by varying the parameter  $\alpha \in [0, 2]$  in

$$\Delta E_{02}^{c,\alpha} = \sqrt{(2 - \alpha)(J_o - J_p)^2 + \alpha[(a_{c,o} - a_{c,p})^2 + (b_{c,o} - b_{c,p})^2]} \quad (4.4)$$

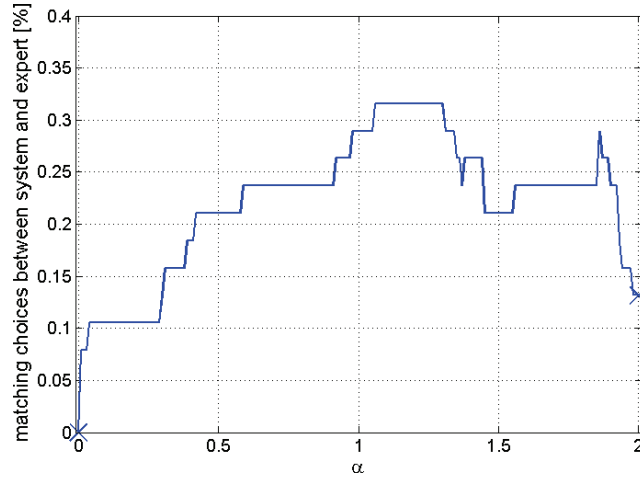
in order to better mimic the expert's method of matching colors. When  $\alpha = 1$ , Equation (4.4) simply becomes

$$\Delta E_{02}^c = \sqrt{(J_o - J_p)^2 + (a_{c,o} - a_{c,p})^2 + (b_{c,o} - b_{c,p})^2}. \quad (4.5)$$

We varied  $\alpha$  by 0.01 steps and ran our system for each  $\alpha$ . We computed how many newly assigned paints match with the expert choices for each iteration. Figure 4.14 shows the result of the optimization. We see that giving slightly more weight to the  $a_c$  and  $b_c$  components gives better results. The maximum is centered around  $\alpha \sim 1.2$ . Looking at the two extreme cases  $\alpha = 0$  and  $\alpha = 2$  also shows that hue is more important than lightness for the expert. Indeed,



**Figure 4.13:** The distance between the crosses and the origin corresponds to the CIECAM02 differences  $\Delta E_{02}$  between the paints and objects computed for the 38 samples satisfying Equation (4.3). The angles represent the objects hues. The color of the crosses corresponds to the grades - 1 is green (18 samples), 2 is cyan (12 samples), 3 is blue (3 samples), 4 is magenta (1 sample), and 5 is red (4 samples). The colored lines correspond to the paints in the database ordered by their hue angle.



**Figure 4.14:** The metric is optimized by weighting the lightness  $J$  with respect to  $(a_c, b_c)$  by varying  $\alpha$ . The optimum is chosen as  $\alpha$  such that the highest number of system and expert recommendations match.

completely discarding the lightness ( $\alpha = 2$ ) still gives some matching results, while we do not obtain any match when  $\alpha = 0$ .

## 4.4 Discussion

Accurate color cannot be retrieved from uncalibrated images taken with uncalibrated cameras. However, a limited range of colors can be estimated by using appropriate color information in the form of a target present in the scene. A color transform mapping the scene target color values onto pre-computed target reference values is calculated by least mean square estimation in sRGB and applied to the entire image. We demonstrate that this method can be successfully applied to Home Décor. All colors can be corrected using a limited number of color calibration targets. When the colored object is within the target colors' convex hull, its color is corrected with an accuracy of  $\Delta E_{02} \simeq 2.19$  ( $\Delta E_{02}^c \simeq 2.03$ ). The color correction consistency across cameras is lower and the average cross-camera values range from  $\Delta E_{02} \simeq 3$  to  $\Delta E_{02} \simeq 6$ . This is due to the variety of samples textures, reflective properties, and color non-uniformities. According to the expert's rating, a distance  $\Delta E_{02} \simeq 5$  still allows a good match. The system presented here is basic and could be easily improved. For example, we used the object pixel mean value, but the distribution of the pixel values could be employed to detect samples that have important textures or color non-uniformities; clipping should also be detected.

The detection of objects was not addressed here. Decoration objects can have any shape and, as opposed to faces, cannot be easily detected in an automated way. The targets could be easily modified by adding a cut-out allowing to place the target on the sample and detected it like any other patch in the target. Moreover, this would ensure that the illumination is the same on both the target and sample, avoiding another important source of error.



---

We show that we can match any colored object by minimizing a CIECAM02 distance between the object color and a database of paints, allowing to return coordinated colors for the object. We obtain results similar to what an expert would do. The metric can be modified by giving more weight to hue to even better match the expert's selection.

The least mean square estimation of the color correction matrix in sRGB allows a fast and computationally low color correction, which can be used in a variety of applications.

Future versions of the method will help users dynamically choose the calibration target using feedback. As we can test whether the object color falls within the targets' gamut, the system will be able to indicate whether the object can be adequately color corrected. If it is not the case, the system will indicate which target users should employ.

## 4.5 Summary

We have demonstrated that the skin color assessment method presented in Chapter 3 can be extended to virtually any object color by using a set of calibration targets instead of one. With nine targets covering all hues, color can be assessed with an accuracy of  $\Delta E_{02}^c \simeq 2$ . The method is applied to a Home Décor application. Rather than generating color harmonies complementing the object, which is a difficult task, we use the corrected object pixel values to find the best match in a database of coordinated paint colors by minimizing a Euclidian distance in CIECAM02 between the object's and paints' colors. The system is fed with an image of a decoration item, which is color corrected. The object's pixels are extracted, their values converted in CIECAM02 and compared against an existing database of CIECAM02 values for paint samples grouped by palettes of four coordinated colors. The system returns the paint color best matching the object and three complementing colors. Our results were rated as very good by a Home Décor expert. The CIECAM02 distances correlate well with the expert's grades and we obtain a consistent cross-camera color correction. The error, measured on the target patches, is of about 2%. The quality of the color correction on actual decoration objects is lower due to variety of textured and non Lambertian materials. It also depends whether the object color lies close to the colors in the calibration target. By comparing the object color with respect to the reference values in sRGB, we can determine if the target is appropriate, which could be employed to guide the user in the target choice.



## Chapter 5

# Illuminant retrieval for fixed location cameras

### 5.1 Introduction

In this chapter, we aim to solve for color constancy in the specific framework of fixed location cameras. Images taken with such devices will exhibit changes in lighting and dynamic content but will also contain static objects in the background. These objects, while unknown, are identical across all images and must thus have the same reflectance spectrum. We use a set of  $N$  images to recover  $N_E = N$  scene illuminants using a few selected surfaces present in every image, i.e., the constant background objects. To solve for color constancy in this framework, we invert an image formation model for a series of test illuminants and obtain a corresponding series of test reflectances. By forcing these reflectances to match, we can deduce the illuminant in every image.

The algorithm consists of two steps: we first define an error function  $f_e$ , the distance between the  $N_\sigma$  reflectance descriptors  $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_{n_\sigma}\}_{n_\sigma=1}^{N_\sigma}$  of the  $N_p$  surfaces present in  $N = N_E$  images and whose variables are the  $N_\varepsilon$  descriptors  $\boldsymbol{\varepsilon} = \{\boldsymbol{\varepsilon}_{n_\varepsilon}\}_{n_\varepsilon=1}^{N_\varepsilon}$  of the  $N_E$  illuminants. The error function  $f_e$  is minimized in a second step. The illuminants corresponding to the  $N = N_E$  scenes satisfy

$$\arg \min_{\boldsymbol{\varepsilon}} f_e(\boldsymbol{\sigma}), \quad (5.1)$$

where the reflectance descriptors  $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon})$  depend on the test illuminant by inversion of the image formation model (2.2). We will see which illuminant and reflectance descriptors can be used to define an appropriate error function  $f_e$ , and how  $f_e$  can be minimized.

## 5.2 Our approach

At each pixel location, the image formation model (Section 2.1) gives a system of three linear equations of the form

$$\rho_k = \mathbf{s}[\lambda]^T \text{diag}(\mathbf{e}[\lambda]) \mathbf{r}^k[\lambda] = \sum_{i=1}^{31} e_i s_i r_i^k, \quad k = 1, 2, 3, \quad (5.2)$$

where  $\mathbf{e}[\lambda] = \{e_i\}_{i=1}^{31}$ ,  $\mathbf{s}[\lambda] = \{s_i\}_{i=1}^{31}$ , and  $\mathbf{r}^k[\lambda] = \{r_i^k\}_{i=1}^{31}$  are the vectors representing the illuminant, reflectance, and sensitivity of the  $k^{\text{th}}$  sensor, respectively. In the present framework, we know the color responses  $\rho_k$  for each surface and the camera sensitivities  $\mathbf{r}^k[\lambda]$ , while both the illuminant and reflectance spectra are unknown. Their 31 samples cannot be retrieved without additional information or without introducing further assumptions. One way of reducing the dimension of this ill-posed problem is to express the illuminant and reflectance spectra as weighted sums of a small number of basis functions. Determining the illuminant and reflectance spectra then reduces to determining the weights.

### 5.2.1 Linear models and illuminant retrieval

In Section 2.6, we saw that daylight illuminants as well as most reflectance spectra could be accurately approximated by weighted sums of three and six to eight basis functions, respectively, and expressed as

$$\mathbf{e}[\lambda] = \sum_{i=1}^{N_\varepsilon} \varepsilon_i \boldsymbol{\mathcal{E}}_i[\lambda] \quad (5.3)$$

$$\mathbf{s}[\lambda] = \sum_{j=1}^{N_\sigma} \sigma_j \boldsymbol{\mathcal{S}}_j[\lambda], \quad (5.4)$$

where  $\boldsymbol{\mathcal{E}}_i[\lambda]$  and  $\boldsymbol{\mathcal{S}}_j[\lambda]$  are the illuminant and reflectance basis functions in vector form and  $\varepsilon_i$  and  $\sigma_j$  are the illuminant and reflectance descriptors, respectively. If we express the reflectance spectra as (5.4), the equations (5.2) are rewritten as

$$\rho_k = \sum_{\lambda} \mathbf{r}_k[\lambda] \sum_{i=1}^{N_\varepsilon} \varepsilon_i \boldsymbol{\mathcal{E}}_i[\lambda] \sum_{j=1}^{N_\sigma} \sigma_j \boldsymbol{\mathcal{S}}_j[\lambda], \quad k = 1, 2, 3. \quad (5.5)$$

We define the  $N_\varepsilon \times N_\sigma$  matrix  $\boldsymbol{\Lambda}_k$  by its elements

$$(\boldsymbol{\Lambda}_k)_{i,j} = \sum_{\lambda} \mathbf{r}_k[\lambda] \boldsymbol{\mathcal{E}}_i[\lambda] \boldsymbol{\mathcal{S}}_j[\lambda] \quad (5.6)$$

and rewrite (5.5) as

$$\rho_k = \boldsymbol{\varepsilon}^T \boldsymbol{\Lambda}_k \boldsymbol{\sigma} = \boldsymbol{\sigma}^T \boldsymbol{\Lambda}_k^T \boldsymbol{\varepsilon}, \quad (5.7)$$

where  $\boldsymbol{\varepsilon}$  and  $\boldsymbol{\sigma}$  are  $N_\varepsilon \times 1$  and  $N_\sigma \times 1$  vectors containing illuminant and reflectance descriptors, i.e., the weights.

This image formation model using linear models for the illuminant and reflectance spectra is called *bilinear*. It is illustrated by the form of (5.7) in which  $\boldsymbol{\sigma}$  and  $\boldsymbol{\varepsilon}$  have symmetrical roles. If we fix  $\boldsymbol{\varepsilon}$ , the system becomes linear in  $\boldsymbol{\sigma}$ , and vice versa. If the illuminant is fixed (5.7) can be written as

$$\rho_k = \Lambda_{\varepsilon,k} \boldsymbol{\sigma}, \quad (5.8)$$

where  $\Lambda_{\varepsilon,k} = \Lambda_k \boldsymbol{\varepsilon}$  is the *lighting matrix*. It can also be expressed in the form

$$\boldsymbol{\rho} = \Lambda_{\varepsilon} \boldsymbol{\sigma}, \quad (5.9)$$

with the  $3 \times N_{\varepsilon}$  matrix  $\Lambda_{\varepsilon} = \mathbf{R} \text{diag}(\mathbf{e}) \mathbf{B}_s$ , where  $\mathbf{R}$  is a  $3 \times 31$  matrix containing the sensor sensitivities and  $\mathbf{B}_s$  is a  $31 \times N_{\sigma}$  matrix containing reflectance basis functions.  $\boldsymbol{\sigma}$  and  $\boldsymbol{\rho}$  are  $N_{\varepsilon} \times 1$  and  $3 \times 1$  vectors, respectively. If  $\mathbf{p} = \bar{\boldsymbol{\rho}}$  represents the average pixel value of a surface extracted from an RGB image, we compute its reflectance descriptors for any illuminant characterized by  $\boldsymbol{\varepsilon}$  by inverting (5.9) as

$$\boldsymbol{\sigma} = \Lambda_{\varepsilon}^+ \mathbf{p}, \quad (5.10)$$

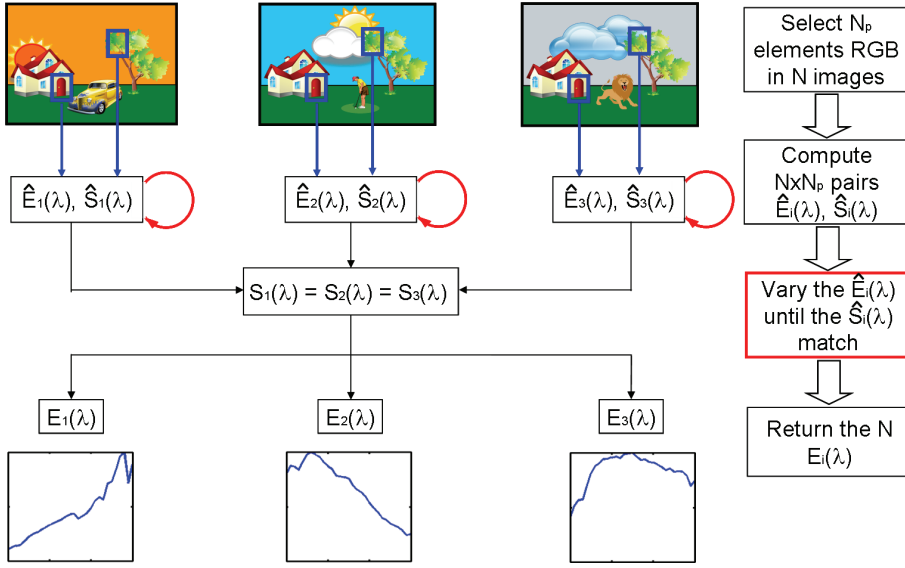
where  $^+$  represents the Moore-Penrose pseudo-inverse. In this expression,  $\boldsymbol{\sigma}$  is an  $N_{\sigma} \times 1$  vector,  $\Lambda_{\varepsilon}^+$  is an  $N_{\sigma} \times 3$  matrix, and  $\mathbf{p}$  is a  $3 \times 1$  vector.

### 5.2.2 Building the error function

We consider  $N$  unknown scenes taken with fixed location cameras and that contain a few static background elements. They are viewed under  $N = N_E$  unknown illuminants, which we assume to be uniform. We extract  $N_p$  patches of uniform color from the objects present in every image and average their pixel values. For each of the  $N \cdot N_p$  surfaces, we solve a system of three linear equations (5.2) for each surface: its mean RGB value  $\mathbf{p}$  allows computing the corresponding reflectance descriptors  $\boldsymbol{\sigma}$  (5.10) for any illuminant. We thus compute  $N \cdot N_p$  systems (5.10) in parallel for a series of test illuminants and obtain a set of reflectance descriptors. By forcing the reflectances of each surface to match across the  $N = N_E$  images, we can deduce the  $N_E$  illuminants. Figure 5.1 illustrates the principle.

Rather than matching reflectances descriptors  $\boldsymbol{\sigma}$  directly, we compute and compare, for each patch, the corresponding color responses  $\mathbf{p}^{D65}$  as viewed under the standard daylight illuminant D65.

Preliminary experiments matching reflectance descriptors  $\boldsymbol{\sigma}$  directly gave poor results. The reflectance basis functions are computed by principal component analysis and ordered by decreasing influence of their contribution to the reflectance spectra. The different weights  $\sigma_i$  are thus not equally important and  $\sigma_1$  contributes more to the resulting spectra than, e.g.,  $\sigma_7$ . We were computing a Euclidian distance between the  $\sigma_i$ 's in  $\mathbb{R}^{N_{\sigma}}$  and, doing so, were giving the same importance to all the components  $\sigma_i$ 's, which is most likely why this approach failed. We decided not to characterize the surfaces by their actual reflectance spectra, but rather by the color that they would have as viewed under the standard daylight illuminant D65. The argument in favor of this approach is the following: the inverse image formation model (5.10) returns only one set of reflectance descriptors for each test illuminant, while there always exists a set of metameric surface reflectances that would yield to



**Figure 5.1:** Illustration of the principle: We select the RGB values of  $N_p$  redundant elements across  $N$  images. By computing metameric pairs of test illuminants and reflectances  $(\hat{E}_i(\lambda), \hat{S}_i(\lambda))$ , and varying  $\hat{E}_i(\lambda)$  until the  $\hat{S}_i(\lambda)$  match, we can retrieve the  $N$  image illuminants.

the same RGB values and that we cannot discriminate using this approach. There is thus no intrinsic reason to match reflectance spectra directly. Instead, we are computing a Euclidian distance between the points in  $\mathbb{R}^3$  representing the color responses denoted  $\mathbf{p}^{D65}$  for each illuminant and each surface. This distance is more meaningful than a Euclidian distance between reflectance descriptors in  $\mathbb{R}^{N_\sigma}$ . Indeed, the three RGB components contribute equally to the resulting color, as opposed to the reflectance descriptors, which correspond to basis functions having different contributions to the reflectance spectra. The color responses  $\mathbf{p}^{D65}$  are computed as

$$p_k^{D65} = \sum_{\lambda} \mathbf{r}_k[\lambda] \sum_{i=1}^{N_\sigma} \sigma_i \mathbf{S}_i[\lambda] \mathbf{e}_{D65}[\lambda], \quad k = 1, 2, 3, \quad (5.11)$$

where  $\sigma_i$  are related to a test illuminant  $\mathbf{e}_{\text{test}}(\lambda)$  via  $\mathbf{\Lambda}_\varepsilon$  (5.10). For any combination of  $N_E$  illuminants we can compute a set of  $N_E \cdot N_p$  corresponding color responses  $\mathbf{p}^{D65}$ , which allows defining an error function reaching its minimum when the sensor responses under illuminant D65 match or, indirectly, when the reflectance spectra match. The error function is computed as the Euclidean distance between the  $N_E \cdot N_p$  sensor responses  $\mathbf{p}^{D65}$  and to a combination of  $N_E$  illuminants represented by  $\mathbf{E}_k = (\mathbf{e}_1(k), \dots, \mathbf{e}_{N_E}(k))$ ,

$$f_e(k) = \frac{1}{N_p} \frac{1}{N} \left[ \sum_{n_p=1}^{N_p} \sum_{j=2, j>i}^N \sum_{i=1}^N [\mathbf{p}_{i, n_p}^{D65}(\mathbf{E}_k) - \mathbf{p}_{j, n_p}^{D65}(\mathbf{E}_k)]^2 \right]^{\frac{1}{2}}, \quad (5.12)$$

where the indices  $i$  and  $j$  run over the images and  $n_p$  runs over the  $N_p$  patches. The double sum on  $i$  and  $j > i$  indicates that the distance is summed over the  $\binom{N_E}{2}$  possible image pairs; for example, if we consider three images, Equation (5.12) becomes

$$f_e(k) = \frac{1}{N_p} \frac{1}{3} \left[ \sum_{n_p=1}^{N_p} [(\mathbf{p}_{1,n_p}^{D65} - \mathbf{p}_{2,n_p}^{D65})^2 + (\mathbf{p}_{2,n_p}^{D65} - \mathbf{p}_{3,n_p}^{D65})^2 + (\mathbf{p}_{1,n_p}^{D65} - \mathbf{p}_{3,n_p}^{D65})^2] \right]^{\frac{1}{2}}. \quad (5.13)$$

A least square error function is used for computational simplicity.

The  $N_E$  scene illuminants corresponding to the  $N = N_E$  images satisfy

$$\hat{\mathbf{E}}_k = \arg \min_k f_e(k). \quad (5.14)$$

We can test all combinations of  $N_E$  illuminants from a test set of  $N_e \geq N_E$  illuminant candidates and return the  $N_E$  illuminants minimizing  $f_e$ . This becomes computationally heavy as the number of iterations increases exponentially with the number of images considered. We can also find the solution of (5.14) by performing a gradient descent on  $f_e$ . In this case, the illuminant spectrum is represented by a linear sum of three basis functions (5.3) and  $f_e$  must be redefined as a function of the illuminant descriptors.

### 5.2.3 Spherical sampling and daylight illuminants

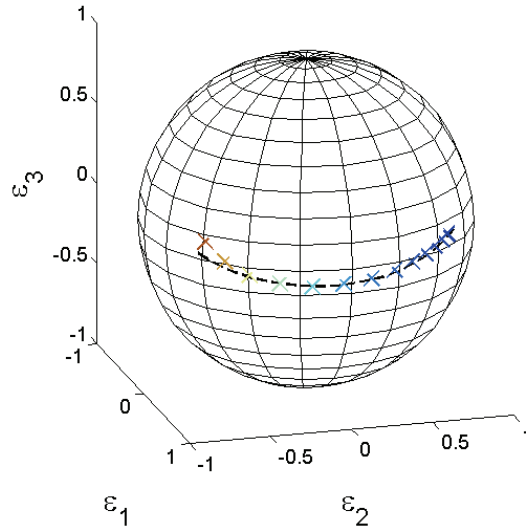
We have seen in Section 2.6 that daylight illuminants can be approximated by a weighted sum of three basis functions. Once the basis functions  $\mathcal{E}_i(\lambda)$  are fixed, the weights  $\varepsilon_i$ , named the *illuminant descriptors*, characterize the illuminant. Using this formalism, any daylight illuminant can be represented as a point in a tri-dimensional space. Moreover, the chromaticity of the illuminant only depends on its relative spectral power distribution; we are generally not interested in absolute values for such spectra. Any illuminant can be normalized and thus be represented by a point on a unit sphere. A daylight illuminant is then expressed by two descriptors, its azimuthal and polar angles  $(\theta_{n_E}, \phi_{n_E})$ , where  $n_E$  indexes the illuminant. The CIE standard daylight illuminants represented on a sphere lie on a line that can be approximated by the quadratic function

$$\phi(\theta) = \frac{\pi}{2} - 0.206\theta^2 + 0.0657\theta - 0.0936, \quad (5.15)$$

represented by the dotted line (see Figure 5.2). A standard daylight illuminant is then described using only one parameter, or descriptor,  $\theta$ . We define and test error functions  $f_e$  according to one, two, or three parameters when the daylight illuminants are represented by one angle  $\theta$ , two angles  $(\theta, \phi)$ , or the usual descriptors  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ , respectively. These functions will be minimized by gradient descent.

### 5.2.4 Gradient descent

A gradient descent algorithm is a first order optimization method allowing to find the minimum of a function. Starting from some initial condition  $x_o$ , one computes the function gradient  $\nabla f$  - or its estimate - and takes one step



**Figure 5.2:** When represented on a sphere, standard daylight illuminants can be approximated by a quadratic function (dashed line). The 13 standard daylight illuminants D40 to D100 are represented, from left to right, by the 'x'.

proportional to the negative of the gradient. The operation is repeated until a minimum is reached. At each iteration, the function is evaluated at a new position  $x_{t+1}$  depending on the values of the function  $f$  at  $x_t$

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t). \quad (5.16)$$

The adaptation rate  $\alpha_t$  can be recomputed at each step or left constant, i.e.,  $\alpha_t = \alpha$ . The first solution can be time consuming, while the second solution may result in slow convergence. Moreover, if the function  $f$  is not globally convex, the algorithm may converge to a local minimum. Minimization by gradient descent is simple, but its convergence is not guaranteed and depends strongly on the adaptation rate  $\alpha$  and step size.

In this chapter,  $f_e$  is minimized as function of the illuminant descriptors  $\varepsilon$  with the following the gradient estimate

$$\nabla_{\varepsilon_i} f_e = f_e(\varepsilon_i + \epsilon) - f_e(\varepsilon_i - \epsilon) \propto \frac{f_e(\varepsilon_i + \epsilon) - f_e(\varepsilon_i - \epsilon)}{\epsilon}, \quad (5.17)$$

where the step size  $\epsilon$  is fixed. If  $\epsilon$  is too small, the convergence may be slow, whereas a large  $\epsilon$  may lead to erroneous results.

At each step, the illuminant descriptors, either  $\theta$ ,  $(\theta, \phi)$ , or  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ , are updated according to

$$\varepsilon_{k+1} = \varepsilon_k - \alpha [f_e(\varepsilon_k + \epsilon) - f_e(\varepsilon_k - \epsilon)]. \quad (5.18)$$

We use fixed  $\alpha$  and  $\epsilon$ . In cases where  $\epsilon$  is too large, it can generate oscillations. To overcome this issue, the sign of the gradient is stored at each step and compared at successive steps. Changes of sign are interpreted as oscillations



and, if they occur, the amplitude of  $\epsilon$  is reduced. The details of the algorithm are reported in Appendix F.

### 5.2.5 Illuminant and image normalization

The “color” of an illuminant does not depend on its intensity. Whether it is specified as an RGB white-point or as its spectral power distribution, the intensity information can generally be discarded. Most color constancy algorithms recover illuminant spectra up to a positive multiplicative factor or as normalized chromaticity coordinates  $(x, y)$  or  $(r, g)$ . Nevertheless, illuminant normalization influences our algorithm, we will thus review different ways of normalizing illuminants and their influence upon the norm of the resulting white-point  $\rho^E$ .

A daylight illuminant expressed as  $E(\lambda) = \sum_{i=1}^3 \epsilon_i \mathcal{E}_i$  can be normalized in the following ways

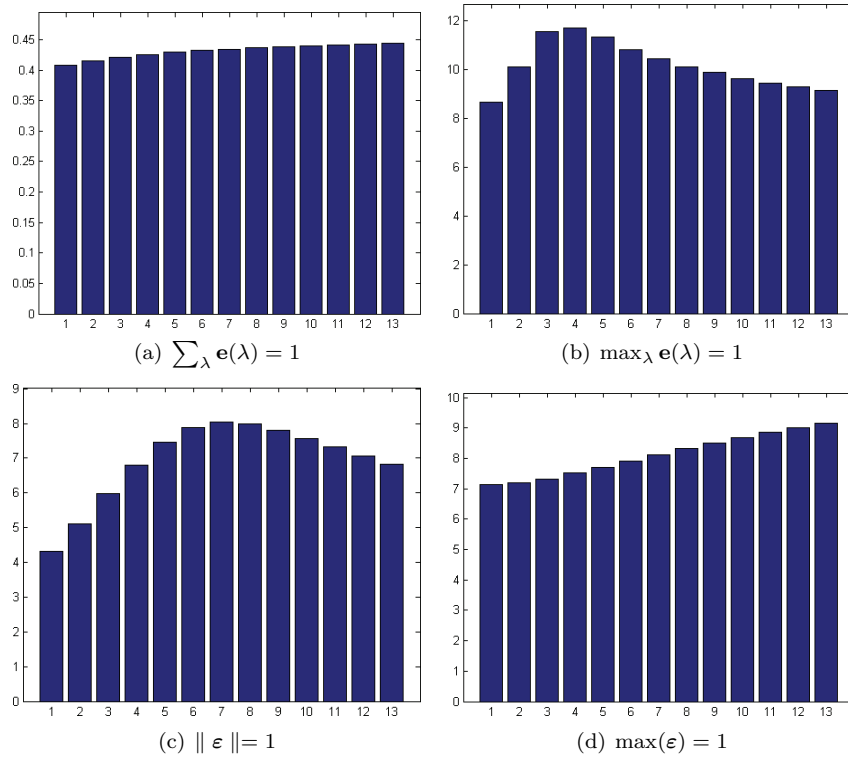
1.  $\int_{\lambda} E(\lambda) d\lambda = 1$  or in vector form  $\sum_{i=1}^{31} e_i = 1$
2.  $\max_{\lambda} E(\lambda) = 1$  or in vector form  $\max_{1 \leq i \leq 31} e_i = 1$
3.  $\|\epsilon\| = 1$ , which is equivalent to representing illuminants on a unit sphere
4.  $\max_{1 \leq j \leq 3} \epsilon_j = 1$ .

The norm of the color response to a perfectly reflective surface  $S(\lambda) \equiv 1$  computed using the image formation model (2.1) changes with the illuminant, independently of its normalization. Let us set  $S(\lambda) \equiv 1$  in (2.1) and compute the white-point of  $E(\lambda)$

$$\rho_k^E = \int_{\lambda} E(\lambda) R_k(\lambda) d\lambda, \quad k = 1, 2, 3. \quad (5.19)$$

When the shape of the illuminant spectrum varies, the surface of the cross-section of  $E(\lambda)$  and  $R_k(\lambda)$  - i.e., the integral of  $E(\lambda)R_k(\lambda)$  - changes. Figure 5.3 shows the norm of the color responses  $\rho^E$  of a white surface  $S(\lambda) \equiv 1$  computed for the 13 standard daylight illuminants D40 to D100 for the four normalizations described above. We see that the norm of  $\rho^E$  varies across the illuminants and the variations of the white-point norm amplitude depend on the normalization. Expressing daylight illuminants with spherical sampling is equivalent to normalizing the illuminant as  $\|\epsilon\| = \sqrt{\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2} = 1$ .

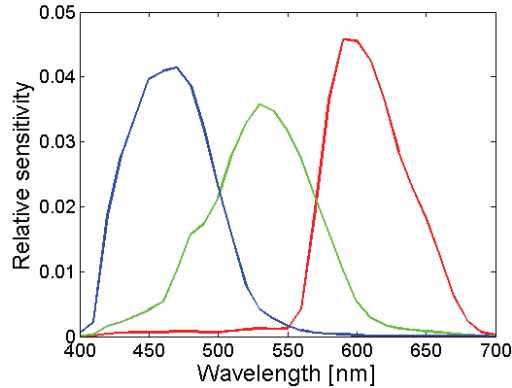
We are working with RGB values that have been normalized by an unknown factor such that the maximum value of the image is equal to one. The normalization of the test illuminant, when computing reflectance descriptors for these RGB values, has an influence on the norm of the reflectance descriptors (5.10). Consequently, the norm of the test color responses  $\mathbf{p}^{D65}$  depends on both the image and test illuminant normalizations. As we are computing a Euclidian distance between these color responses, the relative norm of  $\mathbf{p}^{D65}$  may not be meaningful, which may be a source of error. We thus normalize the color responses  $\mathbf{p}^{D65}$  (5.11) in the error function computation (5.12). We will see that this normalization is unnecessary when the daylight illuminants are expressed using three descriptors (5.3), as the relative amplitudes of  $\epsilon_1$ ,  $\epsilon_2$ , and  $\epsilon_3$  can adjust.



**Figure 5.3:** The norm of the white-point  $\rho^E$  varies across illuminant depending on the illuminant normalization. The numbers 1 to 13 designate the standard daylight illuminants D40 to D100.

### 5.3 Experiments and results

We created sets of RGB values for a series of illuminants and reflectances. The illuminants database consists of a total of 173 spectral power distributions (SPD) containing both computed and measured illuminant spectra, including the 89 measurements from (Funt *et al.* 1995). We created three subsets of illuminants, namely *standard daylight* (13 spectra), *real daylight* (45 spectra), and *artificial* illuminants (36 spectra). Whenever the illuminant description was not completely unambiguous or indicated a mixture of daylight and artificial illuminants, it was left aside. Plots of the selected SPD can be seen in Appendix F.2. The reflectance database contains 24 MacBeth, 462 Munsell, and 219 Natural spectra. We computed RGB patches values for each of the  $24 + 462 + 219 = 705$  reflectance spectra and  $13 + 45 + 36 = 94$  illuminant spectra. The sensor sensitivities are the ones of a Canon 350D camera plotted in Figure 5.4. RGB values were normalized such that the overall maximum is equal to one for each different set of reflectance and each of the 94 illuminants. In other words, the normalization factor varies among illuminants such that it is representative of how RGB images are generally normalized, i.e., such that their maximum value is one. Most simulations were run on synthetic patch values. We generated series of  $10^4$  random combinations of  $N_E$  illuminants



**Figure 5.4:** Canon 350D camera sensitivities.

and  $N_p$  reflectances for each of the three illuminant and three reflectance types and used the corresponding RGB values to test our algorithm.  $N_E$  takes the values 2, 3, 4, 6, and 10 and  $N_p$  takes the values 6 and 10.

The different parts of the chapter correspond to the type of illuminant considered, as reported in Table 5.1. We demonstrate the validity of our method using the small set of 13 standard daylight illuminants in Section 5.3.1. We show that we can retrieve the illuminants by either testing illuminant candidates permutations or by gradient descent. The method is extended to real daylight illuminants in Section 5.3.2. Artificial illuminants, such as fluorescent light sources, often have peaky SPD that cannot be accurately approximated by a small number of basis functions. We see how our method can be applied to such illuminants in Section 5.3.6.

### 5.3.1 Standard daylight illuminants

#### Optimization by testing illuminant combinations

We consider here the 13 standard daylight illuminants D40 to D100 represented in Figure 2.11. The algorithm is tested on several sets of  $10^4$  combinations of synthetic RGB values corresponding to  $N_E$  illuminants and  $N_p$  surfaces, as reported in Table 5.2. We take advantage of the small size of the considered set of illuminants and use  $N_e = 13$  illuminant candidates. For each the  $10^4$  combinations of RGB values corresponding to  $N_E$  illuminants and  $N_p$  surfaces, we test the  $\binom{N_e=13}{N_E}$  illuminants combinations and return the one minimizing the error function  $f_e$  (5.12). The color responses  $\mathbf{p}^{D65}$  (5.11) are normalized. The actual number of illuminant combinations to be tested is  $\binom{N_e}{N_E} N_E!$ ,  $\binom{N_e}{N_E}$  being the number of illuminant combinations multiplied by the number of permutations  $N_E!$ . We sorted the images and used ordered combinations of illuminants to reduce the number of runs by a factor  $N_E!$ . The illuminant estimation error is quantified as the angular error (Section 2.7.3), i.e., the angle in degrees between

Illuminant type	Optimization method	Surface type	Section
Standard daylight	Test combinations	MacBeth Munsell Natural	5.3.1, p. 95
	Gradient descent on $f_e = f_e(\theta)$	MacBeth Munsell Natural	p. 97
Real daylight	Test combinations	MacBeth Munsell Natural	5.3.2, p. 101
	Gradient descent on $f_e = f_e(\theta, \phi)$	MacBeth Munsell Natural	p. 103
	Gradient descent on $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$	MacBeth Munsell Natural	p. 103
Artificial	Test combinations	MacBeth Munsell Natural	5.3.6, p. 113
	Test combinations	Real objects	p. 115

**Table 5.1:** The different parts of this chapter correspond to the different illuminant considered. This table reports the illuminant types, optimization methods, and surface types. It also indicates the section and page for each experiment.

$N_E$	$N_p$	MacBeth	Munsell	Natural
2	6	69.00	79.15	61.75
3	6	88.80	93.67	86.00
4	6	96.80	98.92	92.10
6	6	99.47	99.78	96.88
6	10	100.00	99.90	97.90
10	6	100.00	99.88	98.74

**Table 5.2:** The algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  standard illuminants. This table reports the percentage of the  $10^4 \cdot N$  images for which the standard illuminant was perfectly estimated.

the vectors  $\rho^E$  and  $\hat{\rho}^E$  formed by the real and retrieved white-points in sRGB:

$$e_{ang} = \arccos\left(\frac{\rho^E \circ \hat{\rho}^E}{\|\rho^E\| \|\hat{\rho}^E\|}\right).$$

The resulting angular errors are generally very small, we thus reported the results as the percentage of the  $N_E \cdot 10^4$  images for which the illuminant was perfectly estimated. These values are reported in Table 5.2 for MacBeth, Munsell, and Natural reflectances. The illuminant estimation becomes more robust as the number of illuminants considered increases and is perfect for MacBeth reflectances when  $N_E \geq 6$ . It depends on the reflectance type and is generally better for Munsell reflectance and lower for Natural reflectances.

#### Optimization by gradient descent on $f_e = f_e(\theta)$

The other approach is to minimize the error function  $f_e$  (5.12) by gradient descent. The standard daylight illuminants are then represented using spherical sampling (Section 5.2.3) and are indexed by one angle  $\theta_{n_E}$ .  $f_e$  becomes a function of  $N = N_E$  variables and  $f_e = f_e(\Theta)$ , where  $\Theta = (\theta_1, \dots, \theta_{N_E})$ . This requires to transform the spherical coordinates ( $r = 1, \theta, \phi(\theta)$ ) into the cartesian coordinates ( $\varepsilon_1, \varepsilon_2, \varepsilon_3$ ) prior to the computation of the inverse image formation model (5.10). We used the same  $10^4$  sets of  $N_E$  illuminants and  $N_p$  surfaces RGB as previously. The algorithm was run using two different initializations of the angles  $\theta_{n_E}$ . They were either initialized to  $\theta_{o,n_E} = 0, \forall n_E$  or initialized to  $N_E$  random values  $\theta_{o,n_E} \in [-1, 1]$  and the final angles were averaged over three runs. The range  $[-1, 1]$  represents roughly the angle values in radian corresponding to the 13 standard daylight illuminants. Tables 5.3 and 5.4 report the respective results as the median and maximum angular errors  $e_{ang}$  for each set of parameters. The corresponding error distribution for MacBeth reflectances can be seen in Figure 5.5. We indicated under each plot the percentage of illuminants that were estimated with an angular error under  $3^\circ$ , which is generally considered as acceptable (Hordley 2006). Similar plots for the Munsell and Natural reflectances and the gradient descent parameters are reported in Appendix F.4.1.

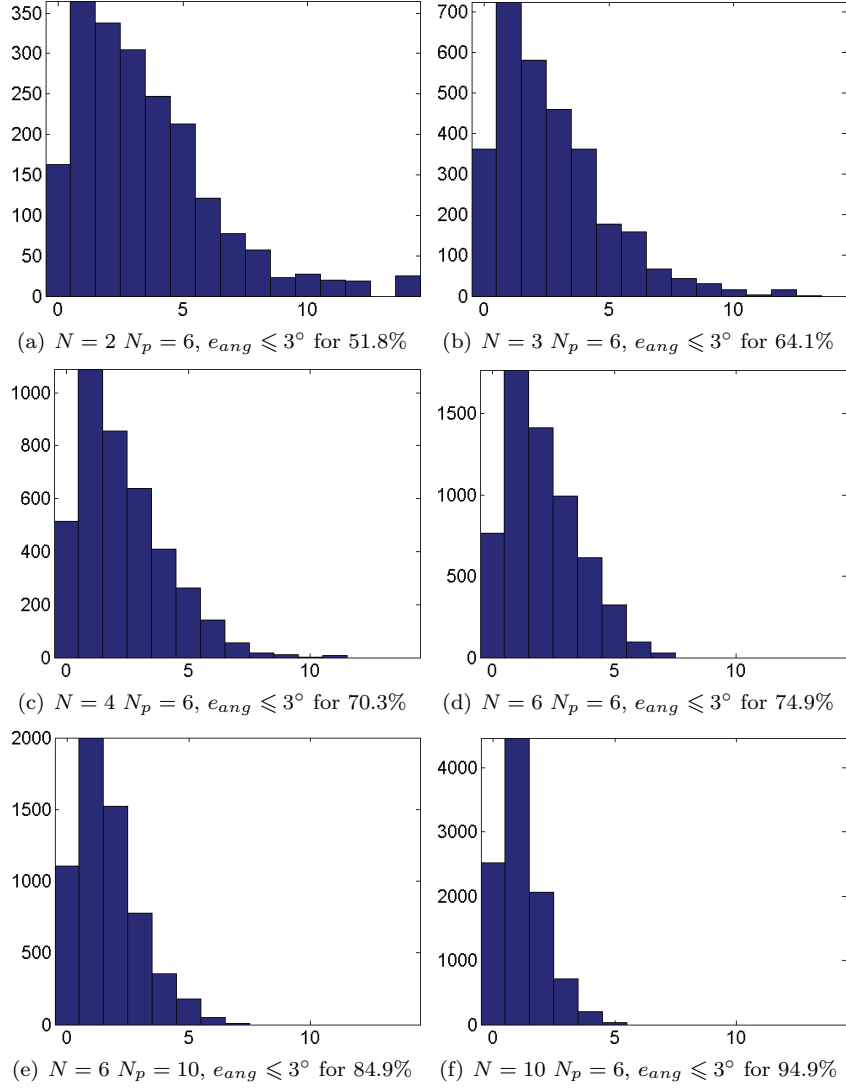
The initialization of  $\theta_{n_E}$  has a limited influence on the final median angular error. If we let the angles  $\theta_{n_E}$  evolve for an arbitrarily large number of itera-

		MacBeth		Munsell		Natural	
$N_E$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	2.88	14.58	2.76	14.45	2.92	14.47
3	6	2.18	12.51	2.12	12.34	2.12	12.49
4	6	1.92	10.99	1.78	10.63	1.94	10.60
6	6	1.81	7.16	1.57	7.61	1.90	8.05
6	10	1.43	6.95	1.34	6.43	1.52	7.22
10	6	0.98	5.13	0.83	4.87	0.95	5.12

**Table 5.3:** Median error angular  $e_{ang}$  computed over  $10^4$  sets of  $N = N_E$  images, i.e.,  $10^4 N$  images. The standard illuminants were computed by gradient descent. The angles were initialized to  $\theta_o = 0$  for all  $N_E$  illuminants. The error distribution is plotted in Figure 5.5.

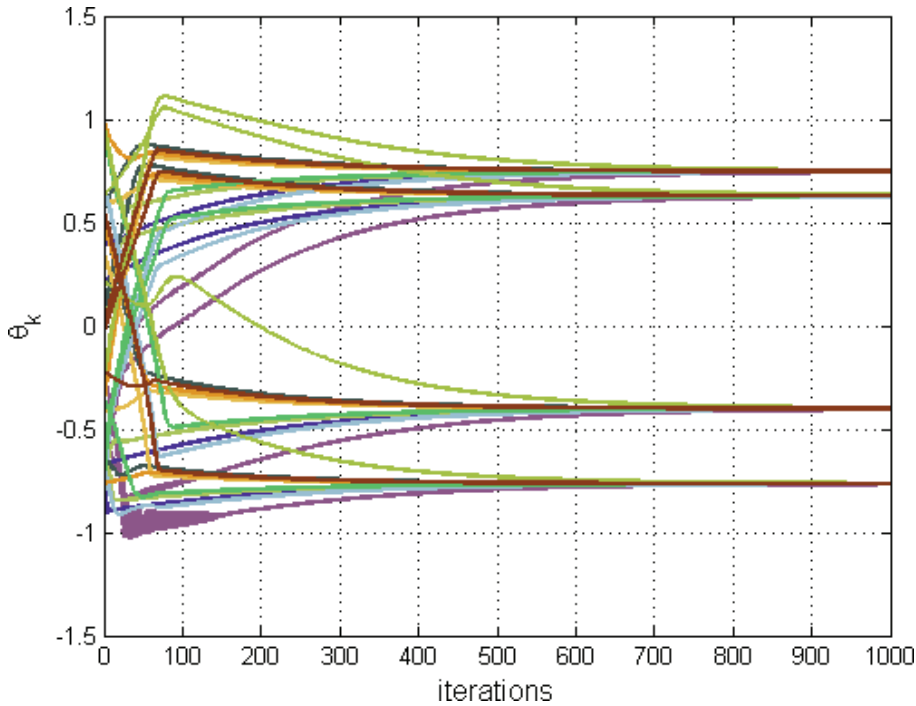
		MacBeth		Munsell		Natural	
$N_E$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	2.80	15.25	2.99	61.00	3.01	15.19
3	6	2.23	27.64	2.20	49.76	2.24	12.85
4	6	2.04	22.87	1.80	10.83	2.09	10.87
6	6	1.81	7.45	1.62	7.32	1.88	8.23
6	10	1.54	6.82	1.47	6.71	1.48	7.35
10	6	1.17	5.43	1.06	5.28	1.30	5.01

**Table 5.4:** Median error angular  $e_{ang}$  computed over  $10^4$  sets of  $N = N_E$  images, i.e.,  $10^4 N$  images. The standard illuminants were computed by gradient descent. The angles were initialized randomly and the final angles were averaged over three iterations.



**Figure 5.5:** Angular error distributions corresponding to Table 5.3. The algorithm was run using standard daylight illuminants and MacBeth reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectances are in Section F.4.1.

tions, they will generally converge to the same result, as illustrated by Figure 5.6, but the convergence can be slow. Imposing the stopping criterion on the norm of the gradient  $\|f(\theta + \epsilon) - f(\theta - \epsilon)\|$  is convenient, but may cause the algorithm not to always completely converge. If we look at Tables 5.3 and 5.4, we see that the final angular errors are still quite close, despite slightly less favorable results when the angles are initialized randomly (Table 5.4). On the contrary, the maximum error can be large when the angles are initialized randomly, as it happened for  $N_E = 2$  and  $N_E = 3$  for Munsell reflectances. An error that large means that the angle  $\theta_{n_E}$  jumped out of the range representing actual illuminants, which can be avoided by initializing all angles to  $\theta_o = 0$ .



**Figure 5.6:** Each set of four lines of the same color represent the evolution of the  $N_E = 4$  angles  $\theta_{n_E}$  indexing standard illuminants. The experiment is run 10 times for different angles initialized randomly in the interval  $[-1, +1]$ . The test illuminants are D45, D55, D90, and D100. The 10 runs converge to the same result.

Table 5.5 focuses on a subset of the results for  $N_E = 4$  and  $N_p = 6$ . It reports the overall smallest and largest angular errors out of the  $10^4$  runs and the corresponding illuminants, as the results for the 10 first runs. We see that the largest error occurs when the illuminants are analogous: it corresponds to the illuminants D40, D45, D50, and D55, which are close, while the smallest error corresponds to illuminants covering the range from D50 to D95. The error order of magnitude is systematically similar and the direction of the shift is the same for all  $N_E$  illuminants.

These results demonstrate that we can retrieve  $N_E$  daylight illuminants from a set of RGB values corresponding to  $N_p$  surfaces as viewed under  $N_E$



	Median angular error				Illuminants			
	$N_E = 1$	$N_E = 2$	$N_E = 3$	$N_E = 4$	$N_E = 1$	$N_E = 2$	$N_E = 3$	$N_E = 4$
min	0.07	0.00	0.16	0.17	D50	D55	D90	D95
1	0.11	0.08	0.03	0.63	D40	D65	D75	D95
2	0.20	0.22	0.18	0.19	D55	D65	D80	D100
3	0.32	0.28	0.33	0.26	D60	D65	D80	D85
4	0.49	0.51	0.42	0.48	D50	D75	D80	D100
5	0.52	0.57	0.48	0.80	D50	D75	D85	D100
6	0.63	0.58	0.63	0.60	D50	D65	D75	D95
7	0.63	0.68	0.76	0.90	D45	D65	D85	D100
8	1.21	1.27	1.19	1.33	D45	D70	D90	D100
9	3.40	3.39	3.36	3.22	D50	D55	D60	D75
10	4.95	5.25	5.29	5.30	D45	D55	D60	D70
max	10.97	10.99	10.89	10.77	D40	D45	D50	D55

**Table 5.5:** We report the ten typical angular errors, as the overall smallest and largest angular errors over the  $10^4$  runs, and the corresponding standard illuminants.

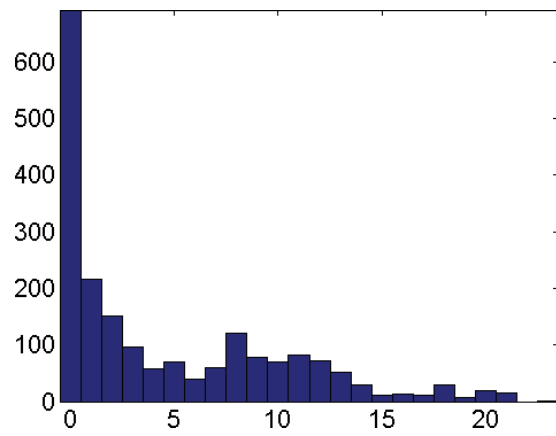
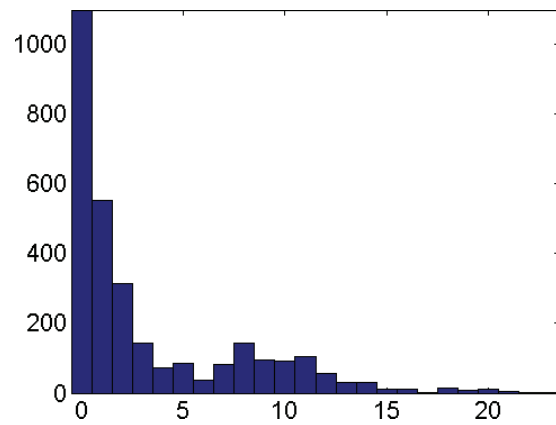
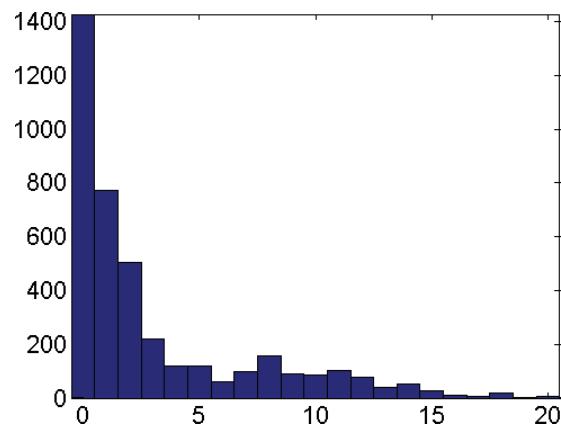
different illuminants by inverting  $N_E$  image formation models in parallel and forcing their output reflectances to match. This is done through the minimization of an error function  $f_e$ , that can be optimized by either testing a set of illuminant candidates combinations or by gradient descent.

The results become more robust with an increasing number of illuminants in both optimization methods. Even when only considering two illuminants, the gradient descent method returns a median angular error under  $3^\circ$ . It falls under  $1^\circ$  when  $N_E = 10$ . The influence of the number of illuminants  $N_E$  is also visible on the maximum error, ranging from  $15^\circ$  when  $N_E = 2$  down to  $5^\circ$  when  $N_E = 10$ .

### 5.3.2 Real daylight illuminants

#### Optimization by testing illuminant combinations

We use here the synthetic RGB patch values generated by combination of 45 real measured daylight illuminants and MacBeth, Munsell, and Natural reflectances. We first select  $N_e = 20$  illuminant candidates from our database of 45 illuminants and test all permutations, as done with the standard illuminants. We generated series of  $10^4$  random combinations of  $N_E$  illuminants and  $N_p$  reflectances and used the corresponding RGB values to test our algorithm.  $N_E$  takes the values 2, 3, and 4 and  $N_p$  takes the value 6. The resulting median and maximum errors are reported in Table 5.6. We see that the illuminant estimation is already robust using a small number of illuminants, with the median angular error of  $1^\circ$  for  $N_E \geq 3$ , but the maximum error can still be large. The error distribution is plotted in Figure 5.7. The percentage of illuminants that were estimated with an angular error under  $3^\circ$  is also indicated. Similar plots for the Munsell and Natural reflectances can be found in Section F.4.2.

(a)  $N = 2$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 56.6%(b)  $N = 3$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 68.7%(c)  $N = 4$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 71.6%

**Figure 5.7:** Angular error distributions corresponding to Table 5.6. The algorithm was run using real daylight illuminants and MacBeth reflectances, by testing all illuminant combinations. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectances are in Section F.4.2.

		MacBeth		Munsell		Natural	
$N_E$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	1.99	22.86	1.65	22.86	2.44	21.20
3	6	1.00	21.62	0.98	21.62	1.02	21.62
4	6	1.00	19.13	1.00	19.91	1.00	19.91

**Table 5.6:** The algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  real daylight illuminants. This table reports the median and maximum angular errors  $e_{ang}$ .

### Optimization by gradient descent

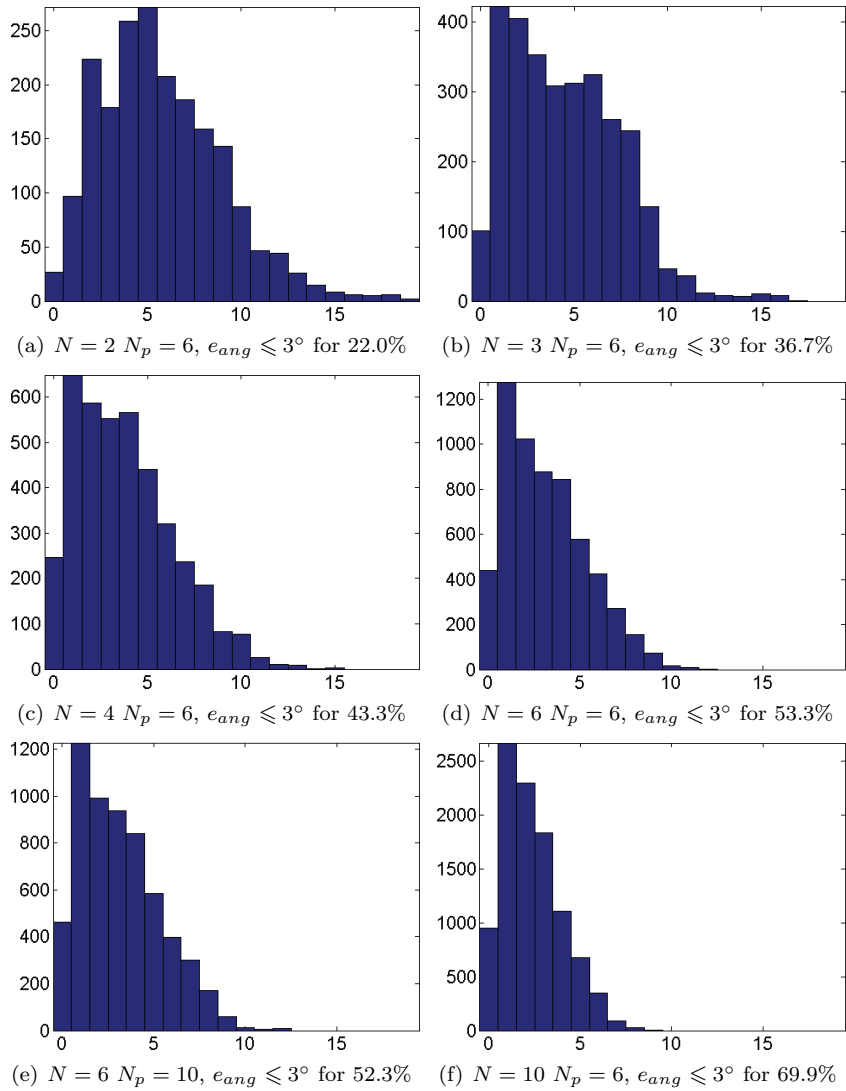
We now propose to minimize the error function  $f_e$  by gradient descent. The measured daylight illuminant spectra are expressed as weighted sums of three basis functions (Section 5.2.1). We present two approaches: we can either use spherical sampling as previously and express the error function with the polar and azimuthal angles as variables, i.e.,  $f_e = f_e(\theta_{n_E}, \phi_{n_E})$  or keep the original linear model formation formulation (5.3) and express the error function  $f_e$  with the three illuminant descriptors as variables, i.e.,  $f_e = f_e(\varepsilon_{n_E,1}, \varepsilon_{n_E,2}, \varepsilon_{n_E,3})$ .

#### Gradient descent on $f_e = f_e(\theta, \phi)$

We have seen that standard daylight illuminants can be represented on a unit sphere and indexed using one angle  $\theta_{n_E}$  (Section 5.2.3). It is naturally extended to real daylight illuminants by letting the polar and azimuthal angles  $(\theta_{n_E}, \phi_{n_E})$  vary. The only modification brought to the algorithm is the change of variables of  $f_e$  from the  $N = N_E$  variables  $(\theta_1, \dots, \theta_{N_E})$  to  $2N_E$  variables  $(\theta_1, \dots, \theta_{N_E}, \phi_1, \dots, \phi_{N_E})$ . This requires to transform the spherical coordinates  $(r = 1, \theta, \phi)$  into the cartesian coordinates and illuminant descriptors  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  prior to the computation of the inverse image formation model (5.10). The angles are initialized such that  $\theta_{o,n_E} = 0$  and  $\phi_{o,n_E} = \phi(\theta_{o,n_E}) \cong \frac{\pi}{2}, \forall n_E$ , according to (5.15). Again, the simulation was run on sets of  $10^4$  RGB values corresponding to random combinations of  $N_E = 2, 3, 4, 6$ , and 10 illuminants and  $N_E = 6$  and 10 surfaces. The results of the simulation are reported in Table 5.7. We observe a similar trend as for the simulation using standard daylight illuminants: Both the median and maximum errors decrease as the number of illuminants  $N_E$  increases. We see though that the maximum error is large for all three reflectance types and  $N_E = 4$  and  $N_p = 6$ . This is due to an isolated set of four illuminants for which the gradient descent diverged, the rest of the results returning angular errors under  $15^\circ$ . The error distributions for the MacBeth reflectances are shown in Figure 5.8. The diverging case is omitted in plot (c). Equivalent plots for the Munsell and Natural reflectances can be found in Section F.4.2.

#### Gradient descent on $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$

We have seen in Section 5.2.5 that the normalization of the test illuminant spectra influences the algorithm's performance. The absolute values of the SPD



**Figure 5.8:** Angular error distributions corresponding to Table 5.7. The algorithm was run using real daylight illuminants and MacBeth reflectances, by minimizing  $f(\theta, \phi)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectances are in Section F.4.2.

		MacBeth		Munsell		Natural	
$N_E$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	5.27	19.33	4.81	18.93	5.15	26.21
3	6	4.24	16.62	3.73	16.23	4.03	26.54
4	6	3.45	52.50	3.42	110.56	3.46	130.21
6	6	2.80	11.77	2.79	13.21	2.92	13.49
6	10	2.85	12.50	2.86	12.66	2.95	12.60
10	6	2.08	8.87	2.06	8.65	2.17	9.99

**Table 5.7:** The algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  real daylight illuminants. The illuminants are described by two angles and the error function  $f_e = f_e(\theta, \phi)$ . This table reports the median angular errors  $e_{ang}$ .

have a direct influence on the norm of the test color responses  $\mathbf{p}^{D65}$ , which are compared in the error function  $f_e$  (5.12). The solution adopted in the two previous gradient descent approaches is to normalize the color responses  $\mathbf{p}^{D65}$ . We now define the error function as a function of not two, but three variables: the illuminants descriptors ( $\varepsilon_1, \varepsilon_2, \varepsilon_3$ ), when the illuminants are expressed as

$$\mathbf{e}[\lambda] = \sum_{i=1}^3 \varepsilon_i \mathcal{E}_i[\lambda].$$

The previous approach constrained the relative importance of  $\varepsilon_1$ ,  $\varepsilon_2$ , and  $\varepsilon_3$  as  $\sqrt{\varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2} = 1$ , while here the  $\varepsilon_i$  can vary independently. There is thus no need to normalize the color responses  $\mathbf{p}^{D65}$ . This way the relative intensity between the RGB patches values is preserved.

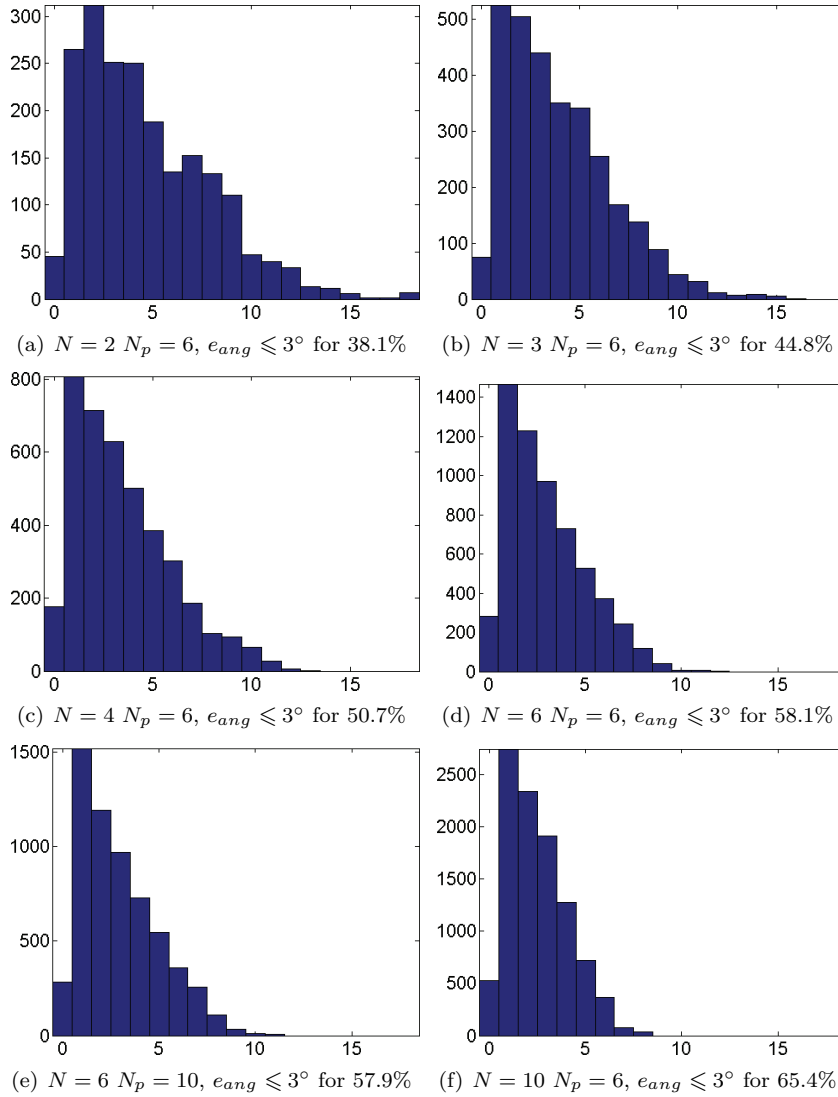
The algorithm is modified in two minors points: the error function now depends on  $3N_E$  variables  $f_e = f_e(\varepsilon_{1,1}, \dots, \varepsilon_{1,N_E}, \varepsilon_{2,1}, \dots, \varepsilon_{2,N_E}, \varepsilon_{3,1}, \dots, \varepsilon_{3,N_E})$  and the transformation from spherical to cartesian coordinates prior to the computation of (5.10) becomes unnecessary. The results using this formulation are presented in Table 5.8. The error distribution for the MacBeth reflectances is shown in Figure 5.9. Equivalent plots for the Munsell and Natural reflectances can be found in Section F.4.2.

Using three illuminant descriptors instead of two gives better results for  $N_E \leq 6$ . The angular error ranges from  $2^\circ$  for  $N_E = 10$  to  $4^\circ$  for  $N_E = 2$ . Both approaches give similar results for  $N_E = 10$ .

### 5.3.3 Comparison with D’Zmura and Iverson’s (1993a) algorithm

We present (D’Zmura and Iverson 1993a, D’Zmura and Iverson 1993b) in detail in Section 2.7 and Appendix B. In this section, we implement both our and D’Zmura and Iverson’s (1993a) algorithms and study their behavior under the influence of added shot noise.

D’Zmura and Iverson’s (1993a) algorithm is extremely sensitive to any deviation from ideal data and only leads to perfect recovery of the illuminants



**Figure 5.9:** Angular error distributions corresponding to Table 5.8. The algorithm was run using real daylight illuminants and MacBeth reflectances, by minimizing  $f(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectances are in Section F.4.2

		MacBeth		Munsell		Natural	
$N_E$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	3.96	18.04	3.91	18.20	3.98	17.98
3	6	3.38	16.02	3.33	15.40	3.42	15.62
4	6	2.96	13.00	2.90	12.90	2.95	13.46
6	6	2.52	11.79	2.43	11.47	2.57	11.34
6	10	2.51	11.36	2.44	11.11	2.59	11.33
10	6	2.19	8.37	2.10	8.30	2.28	9.81

**Table 5.8:** The algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  real daylight illuminants. The illuminants are described by three illuminant descriptors and the error function  $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ . This table reports the median angular errors  $e_{ang}$ .

and reflectances if their spectra can be exactly represented by linear models. Moreover, it is only applicable for specific numbers of illuminants, surfaces, and descriptors. We consider here  $N_p = 3$  surfaces viewed under  $N_E = 2$  illuminants. The algorithm must then be applied with  $N_\varepsilon = 3$  illuminant and  $N_\sigma = 3$  reflectance descriptors. We use the same number of reflectance and illuminant descriptors for comparison with our algorithm and thus minimize an error function  $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  (see Section 5.3.2).

To ensure a perfect recovery of the descriptors, we first build a set of RGB sensor values from ideal tri-dimensional spectra. To do so, we compute the three first basis functions for the illuminants and reflectances by principal component analysis over the 13 standard illuminants and the 24 MacBeth reflectances. Then, for each of them, we build back the tri-dimensional spectra, i.e., compute the  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  and the  $(\sigma_1, \sigma_2, \sigma_3)$  descriptors minimizing, respectively,

$$\| \varepsilon_1 \mathcal{E}_1(\lambda) + \varepsilon_2 \mathcal{E}_2(\lambda) + \varepsilon_3 \mathcal{E}_3(\lambda) - E(\lambda) \| \quad (5.20)$$

and

$$\| \sigma_1 \mathcal{S}_1(\lambda) + \sigma_2 \mathcal{S}_2(\lambda) + \sigma_3 \mathcal{S}_3(\lambda) - S(\lambda) \| \quad (5.21)$$

and then construct the approximated spectra

$$E(\lambda) \simeq \varepsilon_1 \mathcal{E}_1(\lambda) + \varepsilon_2 \mathcal{E}_2(\lambda) + \varepsilon_3 \mathcal{E}_3(\lambda) \quad (5.22)$$

and

$$S(\lambda) \simeq \sigma_1 \mathcal{S}_1(\lambda) + \sigma_2 \mathcal{S}_2(\lambda) + \sigma_3 \mathcal{S}_3(\lambda). \quad (5.23)$$

We finally build RGB sensors responses from these approximated illuminants (5.22) and reflectances (5.23) with the image formation model (2.1).

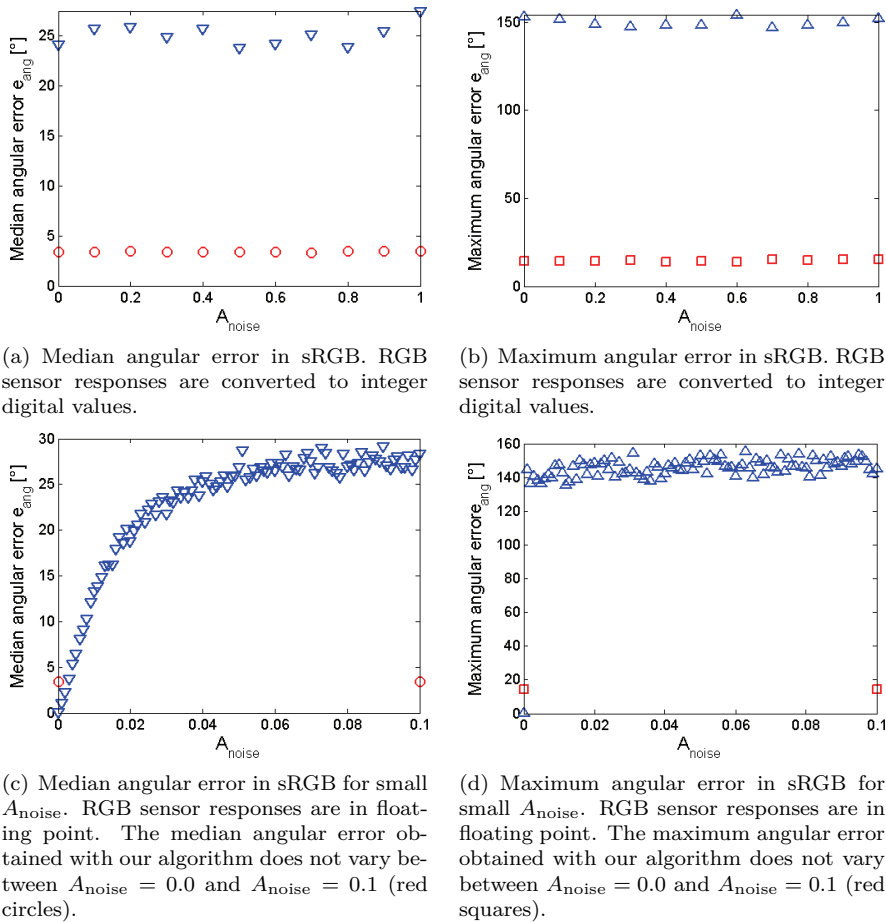
We apply both algorithms on these perfectly tri-dimensional RGB sensor responses and observe their outcomes for an increasing amount of shot noise. The noisy sensor responses are modeled as

$$\rho_i = \rho_{i,o} \pm \sigma_{\text{shot noise}}, \quad (5.24)$$

where  $\rho_{i,o}$  is the ideal sensor response and  $\sigma_{\text{shot noise}}$  is modeled by a Poisson

process of parameter  $\lambda_p = \sqrt{\rho_{i,o}}$ , i.e., centered around  $\sqrt{\rho_{i,o}}$ . The index  $i$  runs over the red, green, and blue sensor responses. In order to control the amount of noise, we introduce a parameter  $A_{\text{noise}} \in [0, 1]$  scaling  $\sigma_{\text{shot noise}}$ . To create a model representative of an actual imaging device, the RGB are converted into integer digital values in  $[0, 255]$  and calculated as

$$\rho_i^{[0,255]} = \rho_{i,o}^{[0,255]} \pm \text{round}(A_{\text{noise}} \sigma_{\text{shot noise}}). \quad (5.25)$$



**Figure 5.10:** Median (left) and maximum (right) angular errors as a function of the amplitude of the added shot noise. The red circles and squares correspond to the results obtained with our algorithm, while the blue triangles correspond to the results obtained with D'Zmura and Iverson's (1993a).

Figure 5.10 reports the median and maximum angular errors for both algorithms as a function of the shot noise amplitude. Our algorithm's performance under the influence of shot noise is only slightly decreased with a median (resp. maximum) angular error ranging from  $3.3^\circ$  (resp.  $13.9^\circ$ ) for  $A_{\text{noise}} = 0$  to  $3.9^\circ$  (resp.  $17.9^\circ$ ) for  $A_{\text{noise}} = 1$ . The median angular error for D'Zmura and



Iverson’s (1993a) algorithm is around  $25^\circ$  (resp.  $150^\circ$ ) even in the absence of noise. While this result may seem surprising, the instability of the algorithm for  $A_{\text{noise}} = 0$  is explained by its high sensitivity to small deviations from ideal data and, in the present case, the error is introduced by the quantization of the digital values. If we run the same experiment for smaller  $A_{\text{noise}}$  while keeping all RGB in floating point, we see that the algorithm indeed performs perfectly and returns a zero median angular error in the absence of noise. It starts to diverge for small values of  $A_{\text{noise}}$  and reaches its maximum median error for  $A_{\text{noise}} \simeq 0.08$ . While our algorithm only approximates the illuminants and never guarantees a perfect recovery of the descriptors, it is quite stable in more realistic conditions. The results for all  $A_{\text{noise}}$  are reported in Table F.5.

### 5.3.4 Gradient descent in presence of noise

We now run our algorithm for  $N_E = 4$  illuminants and  $N_p = 6$  reference surfaces represented by the same RGB sensor responses as in Sections 5.3.1 and 5.3.2, but with added shot noise modeled by (5.25). The amount of noise is controlled by a scaling factor  $A_{\text{noise}} \in [0, 1]$ . We minimize the error function defined as  $f_e(\theta)$ ,  $f_e(\theta, \phi)$ , or  $f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  by gradient descent. Figure 5.11 shows the median and maximum angular errors in sRGB as a function of  $A_{\text{noise}}$ . The results for all  $A_{\text{noise}}$  are reported in Table 5.9.

$A_{\text{noise}}$	$f_e(\theta)$		$f_e(\theta, \phi)$		$f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$	
	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
0.0	1.98	14.85	4.37	25.65	2.96	13.00
0.1	2.32	21.43	4.58	30.62	2.97	13.08
0.2	2.98	48.07	4.94	34.34	2.95	12.89
0.3	4.02	50.07	5.68	32.76	2.99	12.90
0.4	5.92	51.89	6.76	146.12	3.00	13.22
0.5	9.28	51.89	8.23	130.67	3.03	14.93
0.6	11.60	52.04	8.36	152.30	3.12	14.15
0.7	14.99	140.73	11.20	148.31	3.25	15.23
0.8	17.36	143.80	11.81	144.38	3.25	15.90
0.9	18.58	148.56	12.67	150.97	3.34	15.06
1.0	18.62	149.63	13.66	144.64	3.42	18.64

**Table 5.9:** The algorithm was run  $10^4$  times using  $N_p = 6$  patches and  $N_E = 4$  daylight illuminants. This table reports the median and maximum angular errors  $e_{ang}$  as a function on the amount of shot noise added to the sensor responses.

The minimization on  $f_e(\theta)$  is sensitive to noise and the angular error increases rapidly with  $A_{\text{noise}}$ . This is also visible on the standard deviation and on the maximum error. The median (resp. maximum) angular error increases from  $e_{ang} = 1.98^\circ$  (resp.  $e_{ang} = 14.85^\circ$ ) for  $A_{\text{noise}} = 0$  to  $e_{ang} = 18.62^\circ$  (resp.  $e_{ang} = 149.63^\circ$ ) for  $A_{\text{noise}} = 1$ .

We observe a similar trend for the minimization on  $f_e(\theta, \phi)$ , with the median (resp. maximum) angular error increasing from  $e_{ang} = 4.37^\circ$  (resp.  $e_{ang} = 25.65^\circ$ ) for  $A_{\text{noise}} = 0$  to  $e_{ang} = 13.66^\circ$  (resp.  $e_{ang} = 144.64^\circ$ ) for  $A_{\text{noise}} = 1$ .

Moreover, the effect of the quantization error alone is important: If we compare the values reported in Table 5.7, the median angular error was only  $e_{ang} = 3.45^\circ$  for  $N_E = 4$  and  $N_p = 6$ , as opposed to  $e_{ang} = 4.37^\circ$  here.

The minimization on  $f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  is not sensitive to the quantization error and much more robust to shot noise with only a moderate change in performance: The median (resp. maximum) angular error increases from  $e_{ang} = 2.96^\circ$  (resp.  $e_{ang} = 13.00^\circ$ ) for  $A_{\text{noise}} = 0$  to  $e_{ang} = 3.42^\circ$  (resp.  $e_{ang} = 18.64^\circ$ ) for  $A_{\text{noise}} = 1$ .

### 5.3.5 Comparison with a single image color constancy algorithm

In this section, we compare the performance of our algorithm with a single image one. The *gray edge* algorithm (van de Weijer and Gevers 2005) relies on the gray world assumption, yet not applied on the color image, but on its derivative. While this algorithm shows good performance, it can only be applied to real images and is thus not adapted to a simulation on synthetic data. Instead, we use the *shades of gray* (SoG) method (Finlayson and Trezzi 2004), which also relies on the gray world assumption, but employs the Minkowsky  $p$ -norm

$$\|\mathbf{X}\|_p = \left\{ \sum_{i=1}^N |X_i|^p \right\}^{1/p}, \quad (5.26)$$

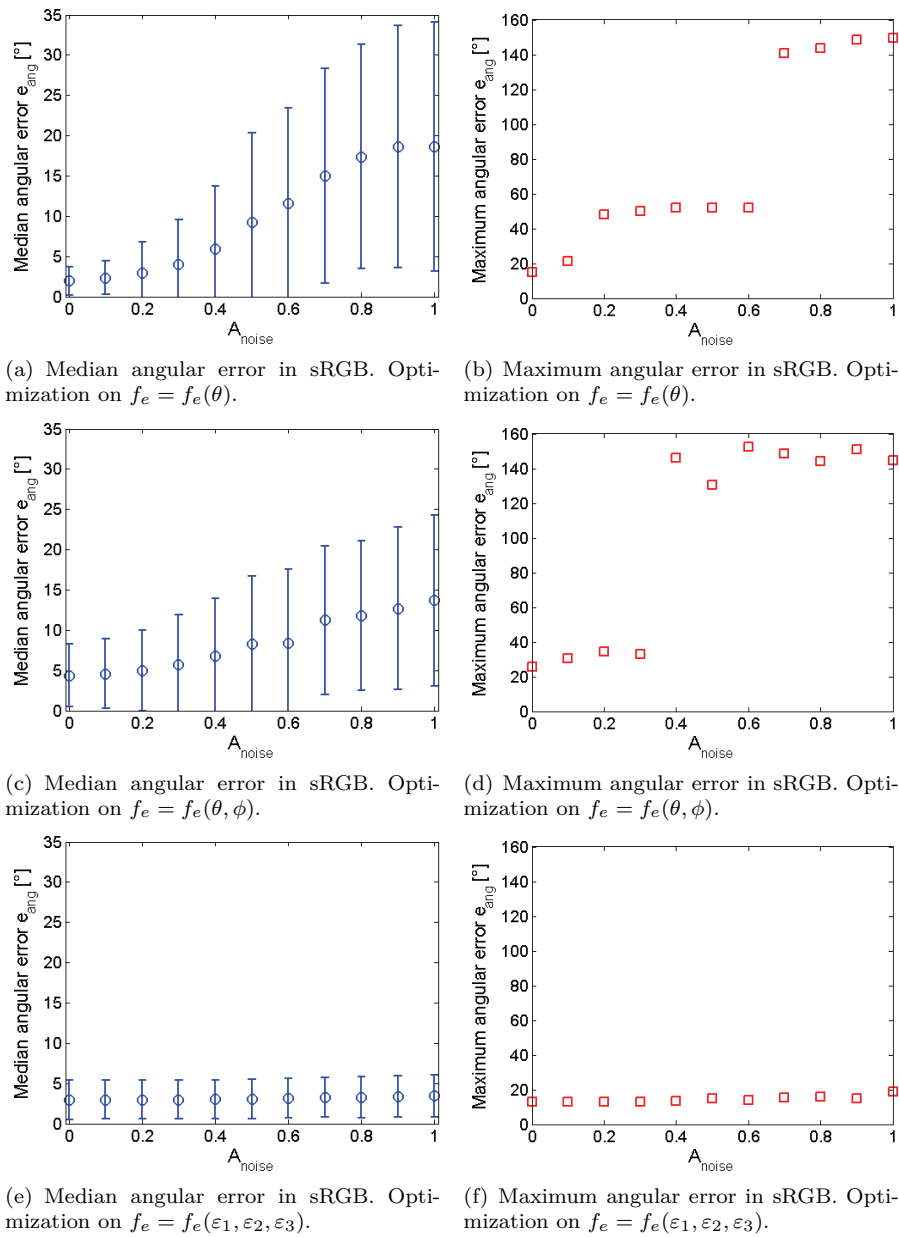
where  $\mathbf{X} = [X_1, \dots, X_N]$  is a vector in  $\mathbb{R}^N$ . More precisely, rather than estimating the white-point from the scene average, it applies the gray world algorithm on the  $p$ -norm of the image. The gray world and the max-RGB algorithms are two particular cases of SoG with, respectively,  $p = 1$  and  $p = \infty$ . Finlayson and Trezzi (2004) showed that the best results are obtained for  $p = 6$ . In (van de Weijer and Gevers 2005), the authors report that, while the  $p = 6$ -norm gray edge performs better than the  $p = 6$ -norm SoG, both return good results, with angular errors computed on a set of real images of  $5.7^\circ$  and  $6.3^\circ$ , respectively.

We apply SoG on the same set of synthetic RGB generated from the standard and measured daylights for the MacBeth, Munsell, and Natural reflectances as in Sections 5.3.1 and 5.3.2. The results are reported in Tables 5.10 and F.6 and should be compared with Tables 5.3, 5.7, and 5.8. Figure 5.12 shows the corresponding error distributions. We also apply SoG for variable amounts of added shot noise, modeled as (5.25). These results are reported in Tables F.7 to F.10.

#### Comparison with SoG in absence of shot noise

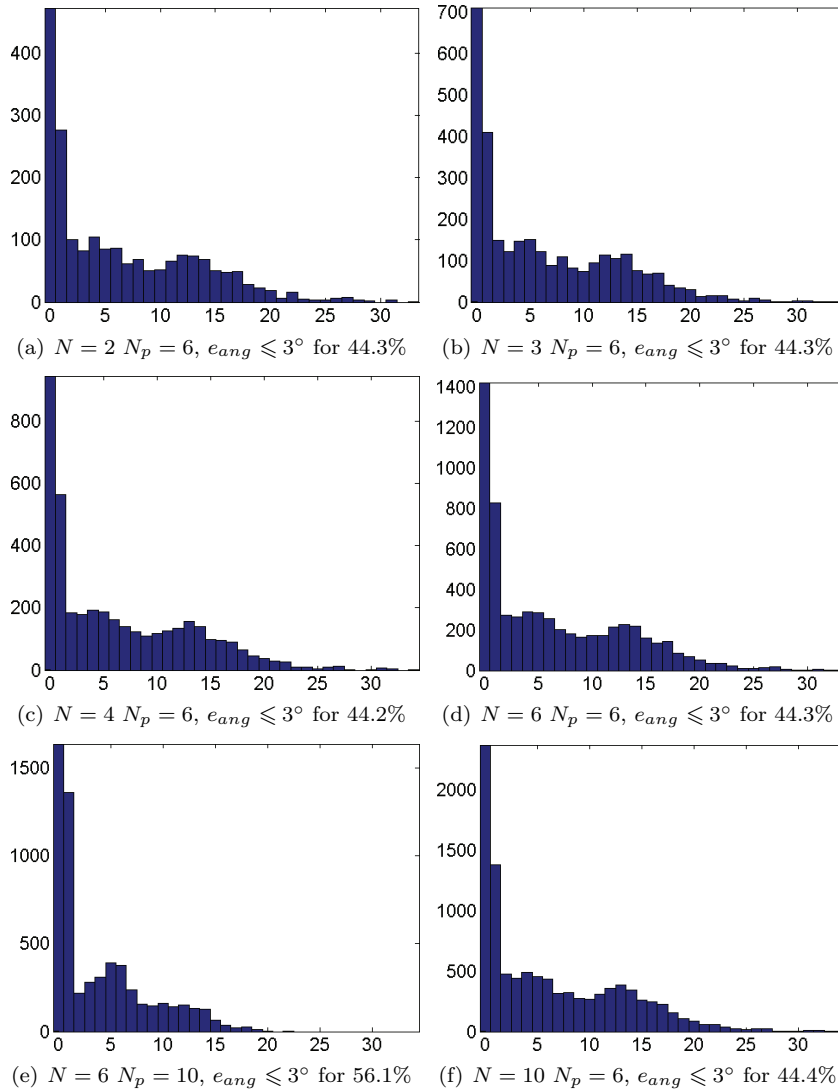
Our algorithm generally outperforms SoG in terms of median angular error. If we consider the optimization by testing illuminants combinations (Tables 5.2 and 5.6), exploiting redundancy across images returns significantly better results.

The results obtained by gradient descent on  $f_e = f_e(\theta)$  (Table 5.3) are generally better, except when considering a larger number of MacBeth reference surfaces ( $N_p = 10$ ), in which case both algorithms return similar results. Indeed, taking  $N_p = 10$  surfaces out of the 24 MacBeth reflectances ensures a large variety of colors, resulting in a good performance of SoG.



**Figure 5.11:** Median (left) and maximum (right) angular errors as a function of the amplitude of the added shot noise for the three gradient descent optimization methods. The error bars represent the standard deviation. The simulations were run on  $N_p = 6$  MacBeth reflectances viewed under  $N_E = 4$  illuminants.  $f_e(\theta)$  was run on standard daylight illuminants, while  $f_e(\theta, \phi)$  and  $f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  were run on real ones.

The gradient descent on  $f_e = f_e(\theta, \phi)$  (Table 5.7) also generally performs better than SoG, except when considering MacBeth reflectances in the two



**Figure 5.12:** Angular error distributions corresponding to Table 5.10. The SoG algorithm was run using standard daylight illuminants and MacBeth reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectances are in Section F.4.5

		MacBeth		Munsell		Natural	
$N$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	4.15	33.41	5.19	31.91	8.06	26.10
3	6	4.26	34.17	5.19	32.54	8.02	27.07
4	6	4.13	34.88	5.17	32.63	8.03	26.57
6	6	4.16	33.72	5.17	32.63	7.98	27.07
6	10	1.51	22.24	4.09	22.74	6.43	29.36
10	6	4.15	34.88	5.17	32.72	8.04	27.07

**Table 5.10:** The SoG algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  standard daylight illuminants. This table reports the median and maximum angular errors  $e_{ang}$ .

cases  $N_p = 2$ , where our algorithm returns a weaker result ( $e_{ang} = 5.27^\circ$ ), and  $N_p = 10$ .

The results obtained by gradient descent on  $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  (Table 5.8) are in overall more favorable than the ones obtained with the SoG algorithm, outside of the particular case of  $N_p = 10$  MacBeth reflectances, which presents ideal conditions for SoG to perform well.

For all the optimization methods, our algorithm systematically remains more robust in terms of maximum error. Also, it is generally less sensitive to the type of reflectances and to the number of reference surfaces and returns better results for Munsell and Natural reflectances.

The SoG results obtained for the real daylight illuminants are very similar to those obtained for standard illuminants and can be found in Section F.4.5.

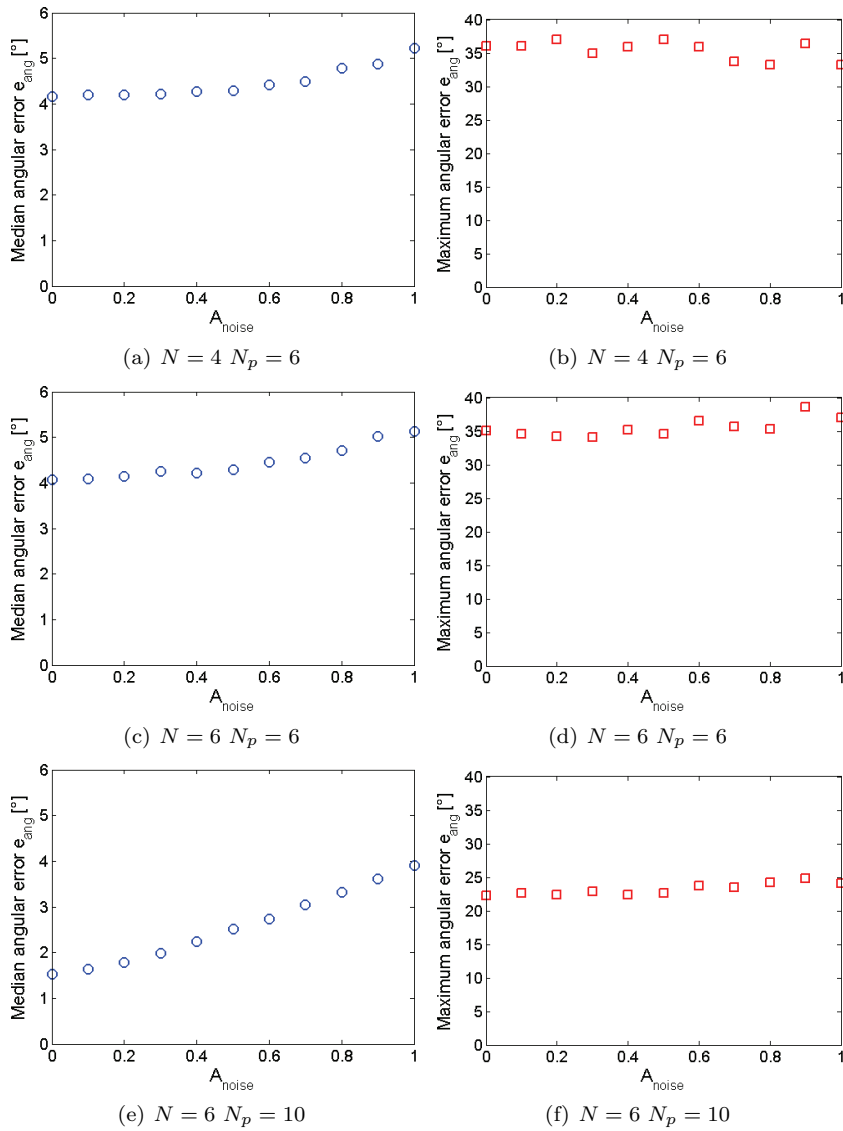
### Comparison with SoG in presence of shot noise

Figure 5.13 shows the influence of shot noise on SoG for MacBeth reflectances. When taking  $N_p = 6$  reference patches, SoG is only slightly dependent on shot noise, with a median (resp. maximum) angular error ranging from  $4.1^\circ$  (resp.  $34^\circ$ ) for  $A_{noise} = 0$  to  $5.2^\circ$  (resp.  $35^\circ$ ) for  $A_{noise} = 1$ . However, when taking  $N_p = 10$  surfaces, it becomes much more sensitive, with a median (resp. maximum) angular error ranging from  $1.5^\circ$  (resp.  $22^\circ$ ) for  $A_{noise} = 0$  to  $3.7^\circ$  (resp.  $27^\circ$ ) for  $A_{noise} = 1$  (see Tables F.7 to F.10).

If we compare these results with Section 5.3.4, SoG is more robust to shot noise than the optimizations by gradient descent on both  $f_e = f_e(\theta)$  and  $f_e = f_e(\theta, \phi)$ . However, the minimization on  $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  showed to be quite stable in presence of noise and outperforms SoG for all noise levels.

### 5.3.6 Artificial illuminants

Only daylight illuminants can be approximated by a small number of basis functions, as opposed to artificial illuminants spectra such as fluorescent lights, which are generally not smooth enough (Smith *et al.* 1992). For example, Romero *et al.* (1997) took seven basis functions to approximate such illuminants. Moreover, they used spectra sampled at  $5\text{ nm}$  intervals, rather than the traditional  $10\text{ nm}$  interval sampling (Smith *et al.* 1992) employed in the



**Figure 5.13:** Median and maximum angular errors plotted as functions of the amplitude of added shot noise for the SoG algorithm. The simulations were run on MacBeth reflectances and standard daylight illuminants.

		MacBeth		Munsell		Natural	
$N_E$	$N_p$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	4.34	36.83	3.64	35.43	5.45	36.83
3	6	3.94	31.08	4.31	30.61	4.34	31.08
4	6	3.35	24.51	3.90	28.16	3.43	28.23

**Table 5.11:** The algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  artificial illuminants. This table reports the median angular errors  $e_{ang}$ .

present approach. We thus cannot apply our gradient descent method for such illuminants.

### Optimization by testing illuminant combinations

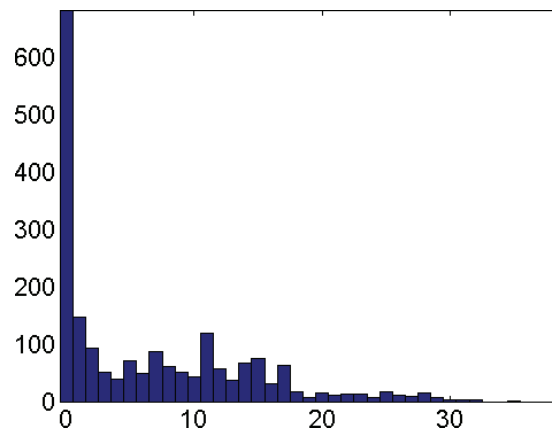
The approach of testing illuminant combinations from a set of illuminant candidates remains valid. We select  $N_e = 20$  illuminants from the 36 artificial illuminants database to run our algorithm. Plots of these illuminants can be found in Figures F.3 and F.4. Table 5.11 reports the median and maximum angular errors  $e_{ang}$  for  $N_E = 2, 3$ , and 4 illuminants. The error ranges from  $3.35^\circ$  for  $N_E = 4$  to  $5.42^\circ$  for  $N_E = 2$ . This is significantly larger than the errors obtained for daylight illuminants. The simulation was run for only  $N_E \leq 4$  due to the large number of combinations  $\binom{N_e=20}{N_E}$  to be tested,  $\binom{N_e=20}{6} = 3.9 \cdot 10^4$  when  $N_E = 6$ . The estimation becomes slightly more robust when  $N_E$  increases, but only half the illuminants are estimated with an angular error smaller than  $3^\circ$ . Figure 5.14 shows the error distributions for MacBeth reflectances. The equivalent plots for Munsell and Natural reflectances can be found in Section F.4.3.

### Lightbooth images

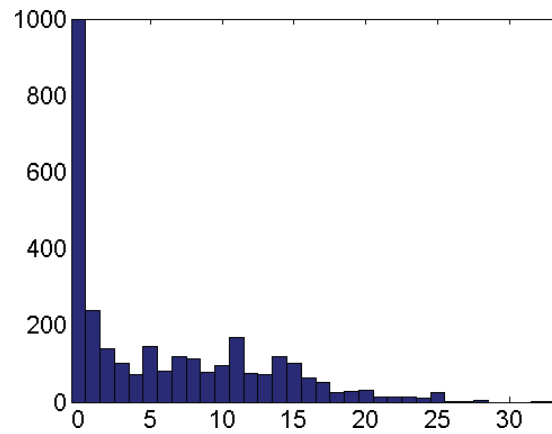
We also tested our algorithm on the four lightbooth images displayed in Figure 5.18. The illuminants are a fluorescent TL83, a fluorescent D65, an incandescent A, and a fluorescent TL84. Their spectra are represented by the red dashed lines in Figures 5.17 and 5.20. We present two typical examples using two different sets of  $N_e = 10$  randomly chosen illuminant candidates, run with the six reference patches indicated in Figure 5.15.

We report two sets of results: Example 1 is reported in Figure 5.16 showing the  $N_e = 10$  illuminant candidates, in Figure 5.17 showing the  $N = 4$  retrieved illuminants, and in Figure 5.18 displaying the original and corrected images. Similarly, Example 2 is reported in Figure 5.19 showing the  $N_e = 10$  illuminant candidates, in Figure 5.20 showing the  $N = 4$  retrieved illuminants, and in Figure 5.21 displaying the original and corrected images.

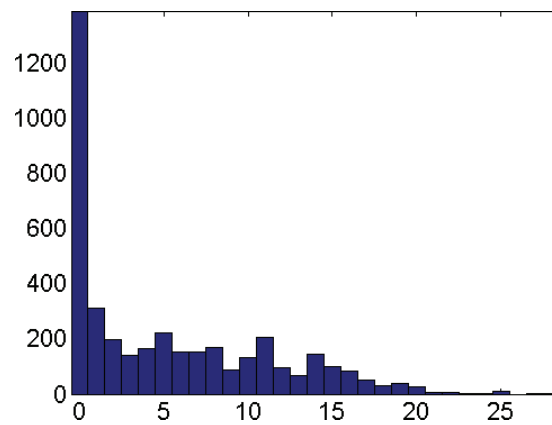
These two examples illustrate two important aspects. The first one is that our method cannot discriminate, for example, an incandescent light from a fluorescent illuminant having a similar white-point, that is, it cannot discriminate metameric illuminants. This is illustrated by Example 1: The reddish fluorescent TL83 is estimated as an incandescent, as displayed in Figure 5.17(a). The



(a)  $N = 2$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 46.9%



(b)  $N = 3$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 47.8%



(c)  $N = 4$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 49.2%

**Figure 5.14:** Angular error distributions corresponding to Table 5.11. The algorithm was run using artificial illuminants and MacBeth reflectances, by testing all illuminant combinations. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectances are in Section F.4.3.





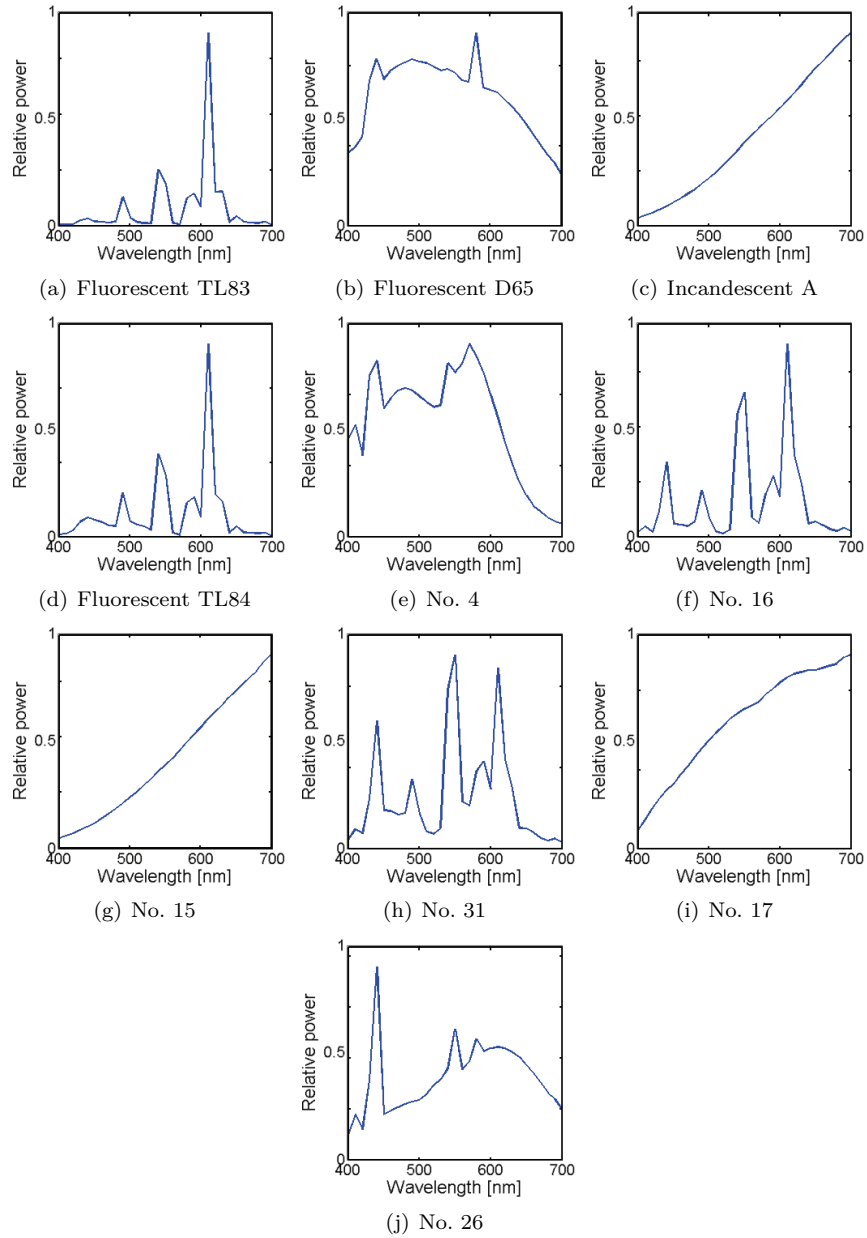
**Figure 5.15:** The six patches used to retrieve the illuminants in the series of images displayed in Figures 5.18 and 5.21 are indicated by the red rectangles.

angular error between the real (red dashed line) and retrieved (solid blue line) illuminants is only  $2.83^\circ$ , despite the very different reflectance spectra. Also, the corresponding image after correction (Figure 5.18(b)) is properly white-balanced.

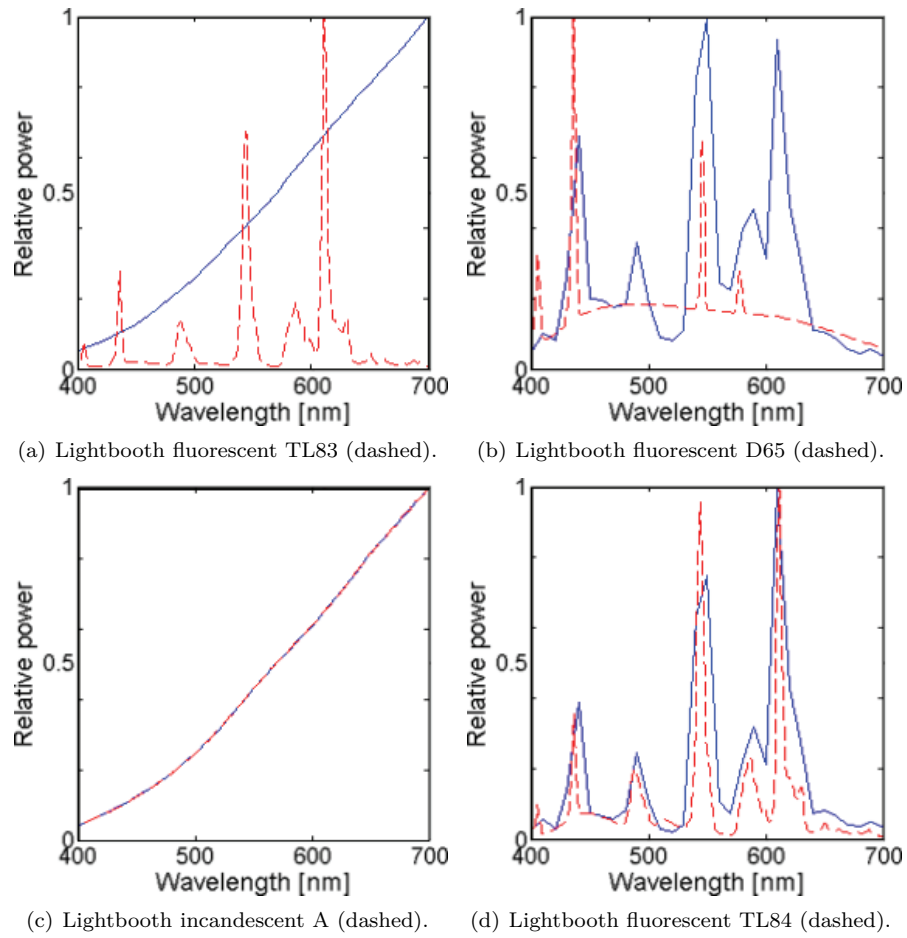
The second aspect concerns the sampling of illuminants spectra, namely that 31 samples may not be sufficient to accurately represent the spectral power distributions of fluorescent illuminants, due to the loss of information caused by the approximation by 31 samples. This is especially visible for the illuminants TL83 and TL84 represented by Figures 5.17(d), 5.20(a), and 5.20(d). For instance, the height of the peak of TL84 at  $550\text{ nm}$  is strongly reduced and the smaller peak around  $440\text{ nm}$  practically disappears. The actual lightbooth illuminants (Figures 5.17 and 5.20, indicated by dashed red lines) measurements are sampled at  $1\text{ nm}$  intervals, i.e., represented by 301 points. Their 31 samples versions are represented in Figures 5.16(a) to (d). Our current model is thus not well adapted for fluorescent illuminants, as the downsampling of illuminant spectra causes all peaks whose width is smaller than  $10\text{ nm}$  to disappear, while they may have a significant influence in the illuminant chromaticity. Still, the resulting spectra allow the images to be properly white-balanced.

### 5.3.7 Reflectances

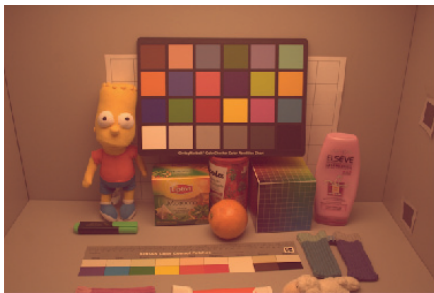
While our aim is primarily to recover illuminant spectra, we also recover an estimate of the reflectances of the surfaces selected to run our algorithm. Figure 5.22 shows an example of the resulting reflectance spectra of the  $N_p = 6$  surfaces used in the simulation using MacBeth reflectances and  $N_E = 4$  stan-



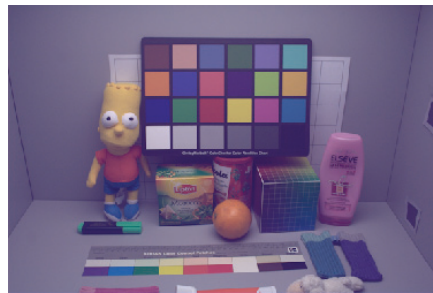
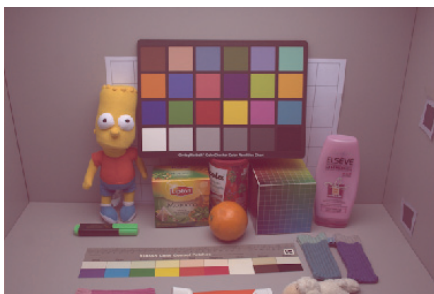
**Figure 5.16:** Test illuminants used in Example 1. Illuminants (a) to (d) are the lightbooth illuminants. The other illuminants are numbered according to their description reported in Table F.4. The resulting images and illuminant spectra are shown in Figures 5.18 and 5.17, respectively.



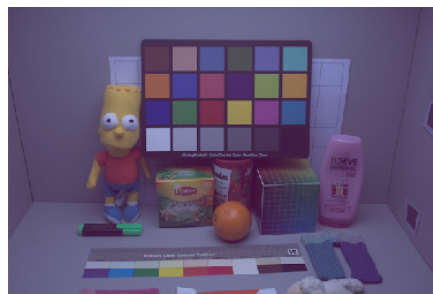
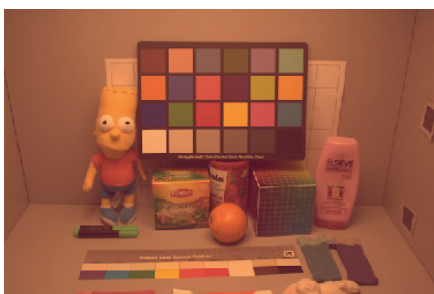
**Figure 5.17:** Resulting illuminants for Example 1. The real and retrieved illuminants are represented by the red dashed lines and solid blue lines, respectively.



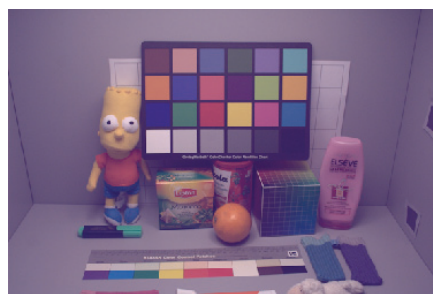
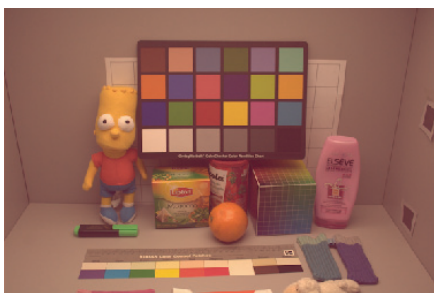
(a) Original fluorescent TL83

(b) Color corrected,  $e_{ang} = 2.83^\circ$ 

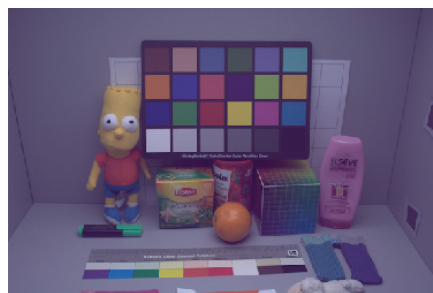
(c) Original fluorescent D65

(d) Color corrected,  $e_{ang} = 16.02^\circ$ 

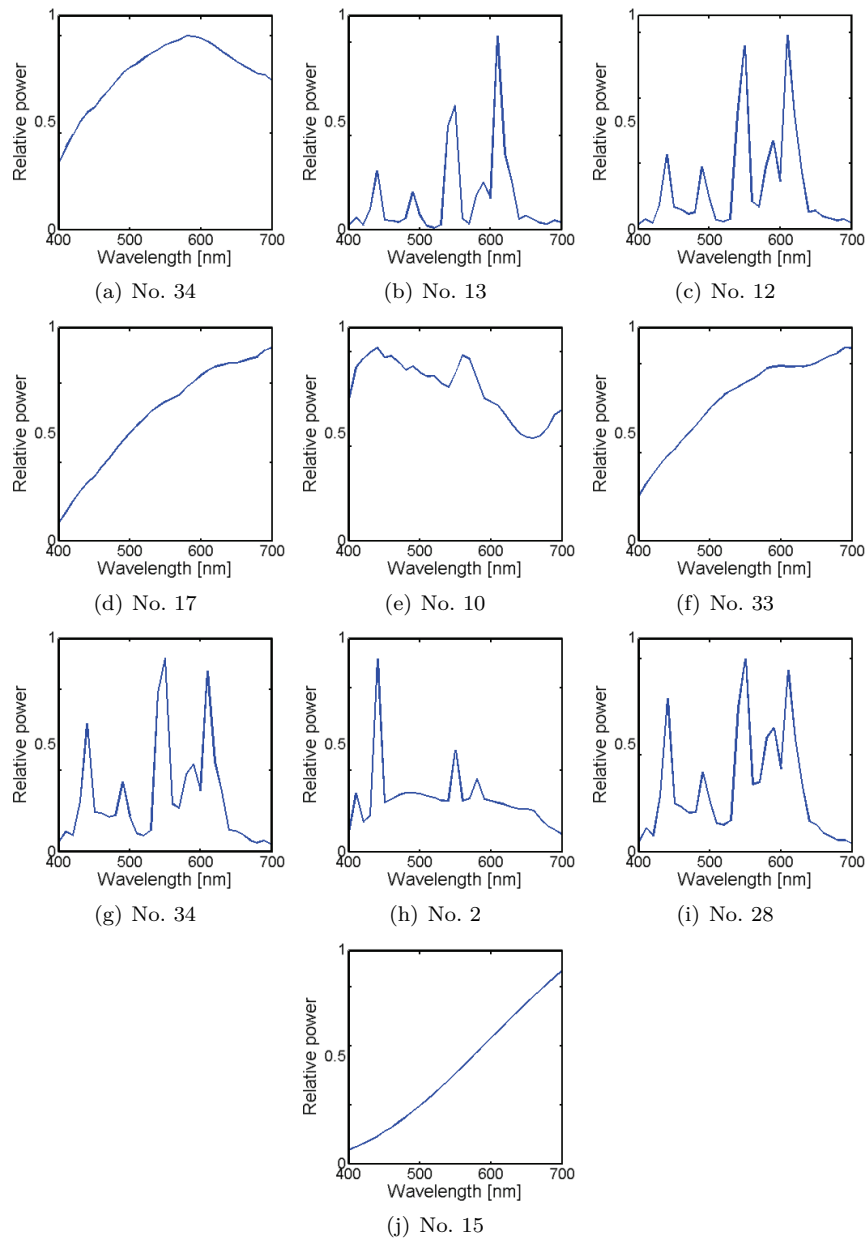
(e) Original incandescent A

(f) Color corrected,  $e_{ang} = 0.02^\circ$ 

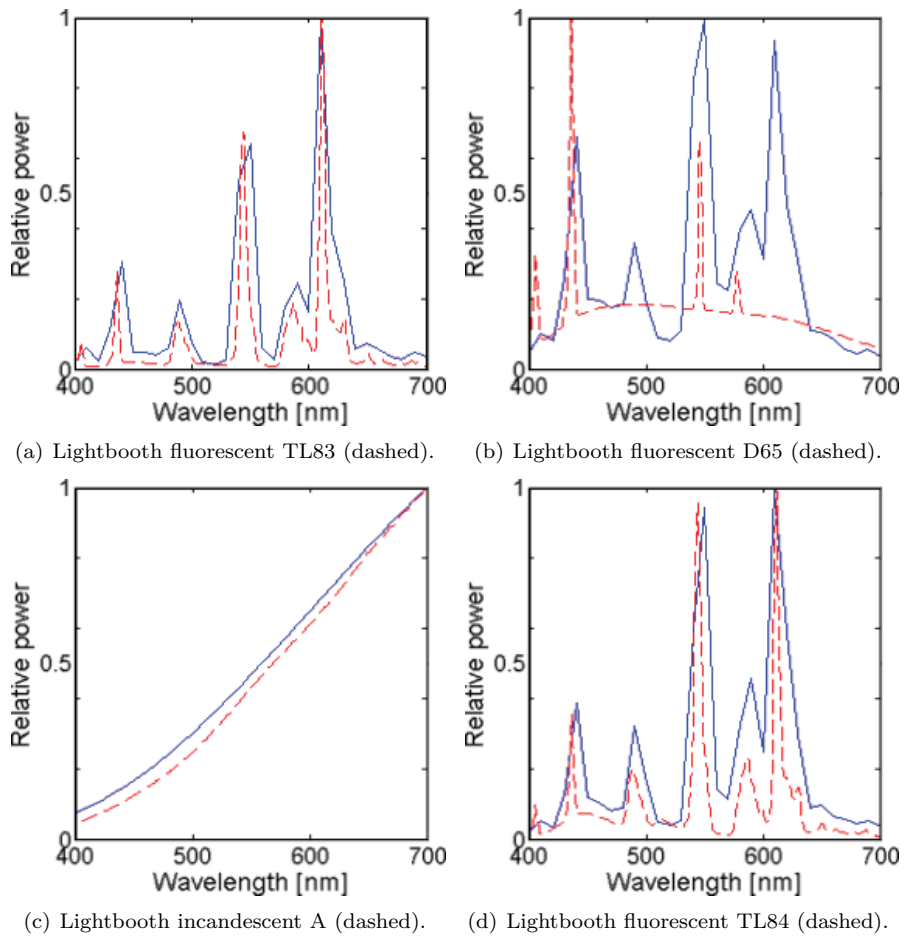
(g) Original fluorescent TL84

(h) Color corrected,  $e_{ang} = 7.03^\circ$ 

**Figure 5.18:** Lightbooth images used in Example 1 and 2. The images in the right column are corrected using the retrieved illuminants displayed in Figure 5.17.



**Figure 5.19:** Test illuminants used in Example 2. The illuminants are numbered according to their description reported in Table F.4. The resulting images and illuminant spectra are shown in Figures 5.21 and 5.20, respectively.



**Figure 5.20:** Resulting illuminants for Example 2. The real and retrieved illuminants are represented by the red dashed lines and solid blue lines, respectively.



**Figure 5.21:** Lightbooth images used in Examples 1 and 2. The images in the right column are corrected using the retrieved illuminants displayed in Figure 5.20.

standard daylight illuminants (Section 5.3.1). We also computed the equivalent CIELAB errors  $\Delta E_{ab}^*$  under D65 between the actual and average retrieved reflectance spectra that we reported below each plot in Figure 5.22. The curves were normalized such that their maximum over all six is 1 for all illuminants. They correspond to the set of surfaces and illuminants yielding to the lowest angular error  $e_{ang}$ . We see that the four retrieved reflectances, indicated by the superimposed solid lines, match. The actual MacBeth reflectances are represented by the dashed lines. Similar plots for the Munsell and Natural reflectance spectra can be found in Section F.4.9, as plot for measured daylight illuminants.

## 5.4 Discussion

We proposed to solve for color constancy using a set of images containing a small number of redundant surfaces. This framework corresponds to fixed location cameras, which take images presenting static background elements. We assume that these elements' reflectances, though unknown, remain constant across images. We invert a series of image formation models in parallel for a set of test illuminants and, by forcing the output reflectances to match, and deduce the illuminants under which each image was captured. These redundant elements are modeled by synthetic RGB patch values. We demonstrate the validity of our method using such patches generated with standard daylight, real daylight, and artificial illuminants.

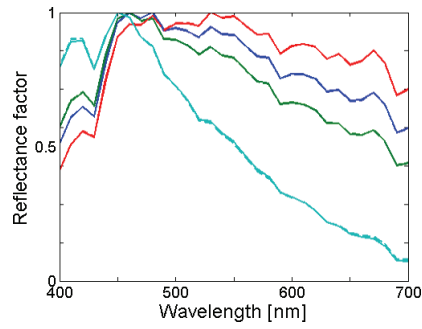
We first present a method to retrieve the illuminants from a limited set of candidate spectra: we compute reflectances by inversion of a series of images formation models from synthetic RGB values for all possible illuminants permutations successively, and select the one for which the output reflectances are the closest, i.e., minimize an error function  $f_e$ . The main advantage of this approach is that it does not require linear models to represent illuminant spectra and can thus be applied to all types of illuminants. A robust estimation necessitates a large and varied candidates database, which causes the number of operations to grow rapidly. Moreover, it also increases with the number of images in the set, as the number of tested combinations  $\binom{N_e}{N_E} N_E!$ , where  $N_E$  is the number of illuminants and  $N_e$  the size of the test set, grows exponentially.

The second optimization method does not require a predefined set of illuminants, because the error function  $f_e$  measuring the surface reflectance similarity is redefined by different variables when a different type of illuminants is considered. Standard daylight illuminants were indexed using one parameter and real daylight illuminants were described by either two or three parameters; the illuminants were then retrieved by performing a gradient descent on the error function.

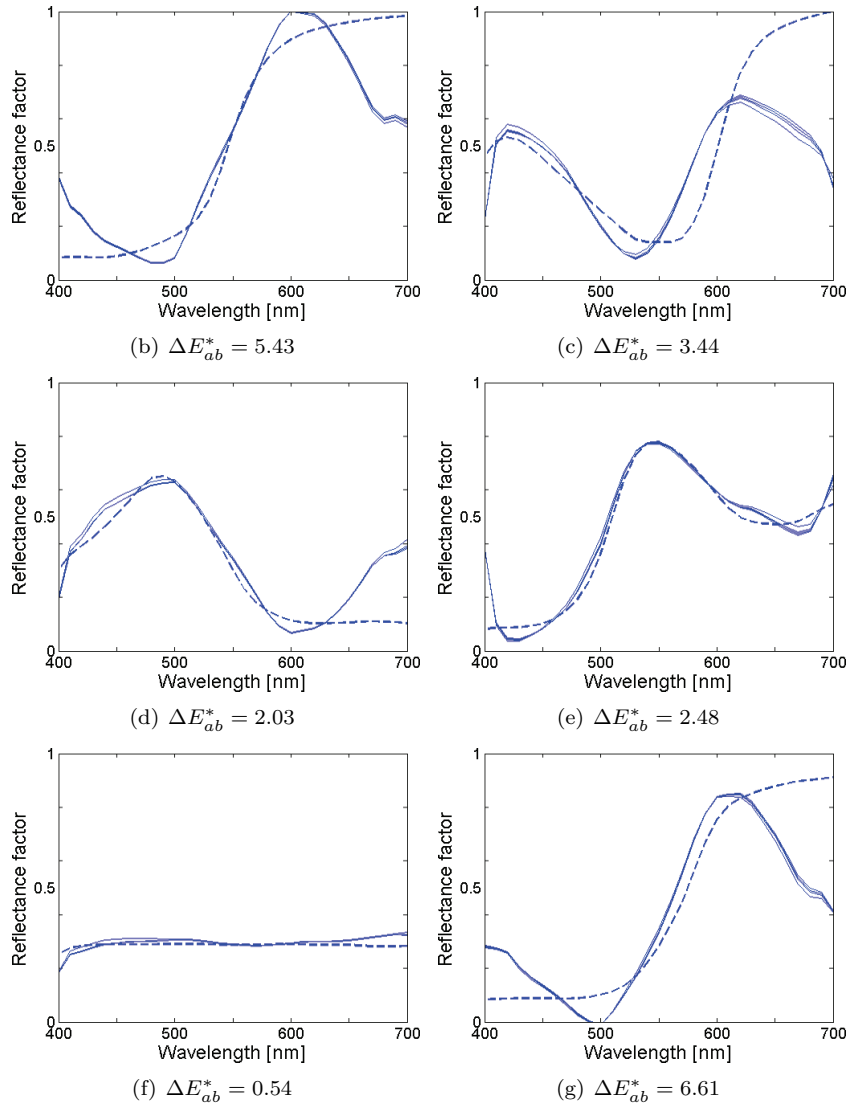
### Daylight illuminants

Our method provides good results for both standard and real daylight illuminants. Standard illuminants estimated by gradient descent (method No. 2) exhibit median angular errors under  $3^\circ$  for  $N_E \geq 2$ . Real illuminants were estimated with median angular errors of  $1^\circ$  by testing combinations of illuminants (optimization method No. 1) with as little as  $N_E = 3$  illuminants and





(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



**Figure 5.22:** Plots (b) to (g) represent the reflectance estimated through the illuminant retrieval. These results were obtained using MacBeth reflectances and standard daylight illuminants (Section 5.3.1). They correspond to the lowest angular error  $e_{ang}$ .

with median angular errors under  $3^\circ$  by gradient descent (optimization method No. 2) with as little as  $N_E = 4$  illuminants and  $N_p = 6$  surfaces. We observed that the estimation becomes more robust as the number of illuminants  $N_E$  increases. There are failing cases and the angular error can be as large as  $100^\circ$ , meaning that the descriptors leave the range of “reasonable” values and the algorithm returns spectra that do not represent any realistic illuminants, which happens if the step size  $\epsilon$  or the adaptation rate  $\alpha$  are too large. The minimization by gradient descent on  $f_e = f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  also showed good performances in the presence of shot noise. It also generally outperformed the results obtained with the single light algorithm *shades of gray*, especially in the presence of shot noise. However, the performance of the minimizations on  $f_e = f_e(\theta)$  and  $f_e = f_e(\theta, \phi)$  decreases rapidly in the presence of noise. It must be emphasized that we did not impose any positivity or shape constraint on the illuminant and reflectance spectra, but only set the initial descriptors to be representative of actual illuminants, such as initializing all illuminants to D65. The range of typical values for each descriptors is known, hence these cases failing due to convergence issues can be detected, and the estimation rerun with new parameters.

### Artificial illuminants

Artificial illuminants were retrieved through method No. 1, i.e., testing illuminant combinations from a set of illuminant candidates. The experiments were run on synthetic images and on four lightbooth images. The resulting median angular errors are significantly larger than with daylight illuminants, but remain under  $4^\circ$  when considering  $N_E = 4$  synthetic illuminants. Applying our method to lightbooth images meets two important issues: Firstly, the current method returns an illuminant having a similar white-point as the actual illuminant, but not necessarily having a spectrum with the same features, such as a reddish fluorescent light approximated by an incandescent illuminant. In other words, it cannot discriminate metameric illuminants. Secondly, real fluorescent illuminants cannot be approximated accurately by 31 wavelength samples ranging from  $400\text{ nm}$  and  $700\text{ nm}$ . Many fluorescent spectra have peaks whose width is smaller than  $10\text{ nm}$  that will not be represented in their downsampled version, while they may contribute significantly to the color of the illuminant.

### General results

Method No. 1 provides good results with a small number of illuminants, but becomes computationally intensive as  $N_E$  increases. The quality of the estimation depends on the content and size of the illuminant candidates database and is thus best adapted to small images sets. Method No. 2, on the contrary, is applicable for a larger number of images and provides good results for  $N_E \geq 4$ . The results obtained by gradient descent may be improved with a better choice of parameters, as the gradient descent used fixed step size  $\epsilon$  and adaptation rate  $\alpha$ . These values could potentially be adapted dynamically, allowing a faster and better convergence and lead to even more favorable results.

This method presents most similarities with (D’Zmura and Iverson 1993a, D’Zmura and Iverson 1993b) presented in Section 2.7 and Appendix B: The authors combined RGB values for a set of surfaces as viewed under several

illuminants into one system of linear equations, where the illuminant and reflectance spectra are expressed with linear models. They first recover the reflectance descriptors by inversion of this system of equations and then take the obtained reflectances to compute the illuminant descriptors by inversion of a simple image formation model. This two-stage recovery method can only be applied with specific numbers of illuminant and reflectance descriptors, namely  $N_\epsilon \leq N_\rho$  and  $N_\sigma \leq N_p$ , where  $N_\rho$  is the number of channels of the imaging device. The principal comparative advantage of our method is that it only imposes weak constraints on the number of images  $N$ , reflectance descriptors  $N_\sigma$ , and surfaces  $N_p$ . While the quality of the illuminant estimation will depend on these parameters, the formalism presented here can be applied to any  $N_E \geq 2$ ,  $N_\sigma \geq 3$ ,  $N_p \geq 1$ , and  $N_\epsilon \geq 1$ . In this thesis we have presented results for a limited number of illuminant descriptors  $N_\epsilon \leq 3$ , but the gradient descent formalism remains applicable when  $N_\epsilon > 3$ . Furthermore, constraints on the number of illuminant descriptors can be avoided by using the first optimization method. Our method is also much more robust to noise. The main comparative disadvantage is that our method requires many iterations, and is thus slower.

## 5.5 Summary

We presented a method allowing to retrieve  $N$  illuminants in a set of  $N$  images containing a limited number of redundant surfaces. This framework corresponds to fixed location cameras, which take images with changing dynamic content, but also containing static background elements. We assume that the reflectance properties of these surfaces, while unknown, remain constant across the images, which allows us to retrieve the illuminant spectra. For each of the surface RGB, we can compute pairs of illuminant and reflectance spectra giving rise to these values. By forcing the reflectance spectra of the redundant image elements to match, we can deduce the illuminant in each image.

We defined an error function that measures the surface reflectance similarity as a function of the  $N$  test illuminants and we presented two optimization methods. The first method considers a limited set of illuminant candidates and tests all combinations from this set. The image illuminants are selected as the combination minimizing the error function. The second method minimizes the error function by gradient descent. We expressed the illuminants using one, two, or three descriptors and applied it to daylight illuminants, which can be accurately represented by a sum of three basis functions. We could estimate the illuminants with an angular error under  $3^\circ$  with as little as  $N = 6$  images and  $N_p = 6$  surfaces. Our method imposes few constraints on the different parameters and can be applied with only  $N = 2$  images,  $N_p = 1$  surface in each images, and  $N_\sigma = 3$  reflectance descriptors. There is no theoretical upper limit on these parameters other than the increasing computation cost.



# Chapter 6

## Conclusion

This thesis covers two quite different projects, which are actually two sides of the same problem, namely to know which illuminant and reflectance information can be extracted from images containing some additional redundant knowledge.

### 6.1 Color assessment from uncalibrated images

#### 6.1.1 Summary

The first project, presented in Chapters 3 and 4, tackles the problem from a practical point of view. The application at hand, namely to give color advice from standard RGB images in consumer oriented services, requires an accurate color correction method that is simple to use, that does not demand any specific training of the customer, and that is inexpensive and fast. Basically, the system works as follows: a customer images an object using a cell phone integrated camera and sends the picture to a server where image information is extracted and exploited in order to return consistent advice. The constraints must be minimal for the user, who should not have any difficulty employing the system and should not need any sophisticated equipment or training.

The solution is to provide the customer with a target designed specifically for the application that s/he must image along with the object of interest. The target patches are compared with reference colors allowing the computation of a transform that accurately corrects the object color. We have demonstrated in the specific case of a makeup advice application that this method can be successfully applied to small ranges of colors, such as skin tones, which can be corrected with a precision of  $\Delta E_{ab}^* < 1$ . The experiment was run on a set of images of 53 subjects, whose skin spectra was also measured. Face colors derived from images were shown to correlate well with spectrally derived values. We have also demonstrated that face colors can be consistently assessed across cameras from various manufacturers and of different qualities and resolutions, without bypassing the in-camera processing.

The method was modified to be extended to any arbitrary color by using several targets instead of one and applied to the specific case of a Home Décor application providing users with a set of colors complementing the one of a

decoration object. We designed a set of nine targets covering all hues and collected a set decoration items samples of various colors and materials. Each sample was imaged with a target consisting of similar reference colors that were employed to compute a linear correction transform. The corrected pixels were then matched with a paint samples database. An expert was asked to rate the quality of the system's pick for each sample, which were generally qualified as really good.

The results underlined an important requirement for the correction to be accurate: the object color must be "close enough" to the reference ones. We propose a method to quantify this distance by comparing the volume spanned by the reference values alone in sRGB with the one spanned by the reference values and the object's. Furthermore, results suggest that the accuracy of the color transform is roughly inversely proportional to the range of colors considered. Indeed, the hue angle spanned by the skin tones is roughly half of the one spanned by each of the Home Décor application targets and the correction accuracy estimated on the skin patches is twice as good in terms of CIELAB differences.

### 6.1.2 Future research

We have demonstrated the feasibility of the color assessment method in the framework of two specific applications. Yet, there are many parameters whose individual influences are not clear. For instance, in the case of the skin color assessment application, the image derived face color depends strongly on the quality of the skin pixels extraction. Similarly, in the case of the Home Décor application, the color is dependent on the material texture and uniformity. While the results are satisfying from a practical point of view, in the sense that the system returns adequate advice, which is sufficiently good for the studied applications, it would be interesting to set a framework in order to estimate the influence of each factor individually, such as the extent and sampling of the reference gamuts without being tributary of the uncontrolled errors due to materials' textures or faces' shadows.

The application itself can also be improved: we compared the relative positions of objects colors with respect to the reference ones to qualify the image correction, but it could also be employed to return feedback to the user, such as detecting when the calibration target is not adapted and guiding in the choice of the appropriate one. Furthermore, we only used average face and object values, while the distribution of the pixels could also be exploited. For example, if they exhibit an important variance, this should be interpreted as a nonuniform object. If this turns out to be robust enough, it could even be used to identify the dominant color and discard the others. Clipping of object pixels should also be detected by examining the pixels distribution.

While face detection is performed using a Viola-Jones algorithm (2001), we have not addressed the detection of objects in the Home Décor application. Due to the variety of shapes and sizes we may encounter, robust automatic detection of objects will be difficult and the simplest alternative is to have a cut-out in the target that exposes the object.

The design and the number of the targets can be modified; for instance, we used solely 9 targets to cover "all colors" in order to obtain a system that is suitable for an untrained customer in the perspective of commercial and

mobile applications. Still, we could think of other versions of this system for professionals that would be slightly more complex and may require a minimal training. It is reasonable to think that with a higher number of targets, a good quality consumer camera, a more sophisticated computer software, and eventually a partial control on the illuminant, high accuracy color assessment can be achieved, which could be employed in applications designed for professionals in various domains such as dermatology, medicine, Home Décor, clothing advice, etc.

## 6.2 Illuminant retrieval for fixed location cameras

### 6.2.1 Summary

We developed the second method in a theoretical framework corresponding to the context of fixed location cameras, such as a surveillance or panorama ones, that capture images containing static elements. While the illuminant and the dynamic scene content vary, we assume that there will be constant background objects present in all the images. We model their surface's pixel values by synthetic RGB ones to demonstrate that, under the assumption that the reflectances of these surfaces remain constant, we can retrieve the illuminants in a set of images. This is achieved through the minimization of an error function that measures the similarity between the descriptors of the surfaces across the image set as a function of the test illuminants. The function reaches its minimum when the reflectances of the static objects match across all images.

We proposed two error minimization methods. The first solution is simply to evaluate the error function for a set of illuminant candidates permutations and return the one minimizing the error. With this approach, the resulting median angular error is of about  $1^\circ$  when three or more images built using standard or real daylight illuminants are considered. We proposed another method better adapted for larger sets of images and suggested to minimize the error by a gradient descent on the illuminant descriptors. The daylight illuminants were described using one, two, or three descriptors. The resulting median angular error falls under  $3^\circ$  when six or more real daylight illuminants are considered.

Our illuminant estimation method by gradient descent uses linear models to represent illuminant and reflectance spectra, but with the advantage of not imposing upper limits on the number of their descriptors. The framework developed here can thus be extended to linear models of arbitrarily high dimensions.

### 6.2.2 Future research

The gradient descent method returned satisfying results, but the algorithm itself can be improved. We have indeed used fixed gradient descent parameters, namely fixed adaptation rate and step size, while a more sophisticated version with adaptive parameters should yield to a faster convergence and maybe even more favorable results. Furthermore, we have limited the gradient descent to daylight illuminants and defined error functions as variables of up to three parameters. The same formalism can be applied to a higher number of

---

illuminant descriptors, in which case convergence should be investigated. If these experiments return positive results, the method could be applied to non-daylight illuminants, which require to be represented by more than three basis functions.

We have set the theoretical framework and validated the approach through experiments on synthetic RGB values. To be applicable in practice, this should also be tested using real images, which rises several issues, namely variation of intensity across the images, mixed illuminants, shadows, and specularities. Mixed illuminants in particular are an issue in the case of a  $360^\circ$  panoramas, where there will be mixtures of illumination conditions such as direct sunlight and indirect skylight. The current version of the algorithm should be robust towards variations in illumination intensity, which is taken care of by simply normalizing the RGB responses. It should also be possible to detect mixed illuminants and shadows by comparing the RGB values of the redundant objects across all the images. Indeed, the ratio of the red, green, and blue components will change if shadows or different illuminants are projected onto some of the static image elements, whereas in the case of a uniform illumination, their relative amplitudes should be constant. Changes in the ratios of the RGB components of two objects can thus be interpreted as a change in illumination. While this would help detecting mixed illuminants, it would not discard their influence. Another approach is to take supplementary information into account, such as the time of the day, to predict the position of shadows, which would greatly increase the complexity of the current method.



# Appendix A

## CIECAM02 model

In this section, we report a summary of derivation of the color appearance model CIECAM02 (Moroney *et al.* 2002, CIE 2004), based on Fairchild (2005).

Input data for the CIECAM02 model includes:

- Relative tristimulus values of the test stimulus  $XYZ$
- Relative tristimulus values of the white point  $X_w Y_w Z_w$
- Adapting luminance, often taken to be 20% of the luminance of a white object in the scene,  $L_A$  in  $cm/m^2$
- The relative luminance of the surround (dim, dark, average)
- A decision whether to discount the illuminant.

The relative luminance of the surround sets the values of an exponential non-linearity  $c$ , of the chromatic induction factor  $N_c$ , and of the maximum degree of adaptation  $F$  (see Table A.1).

Output data of the CIECAM02 model includes:

- Hue
- Lightness
- Brightness
- Chroma
- Colorfulness

Viewing conditions	$c$	$N_c$	$F$
Average surround	0.69	1.0	1.0
Dim surround	0.59	0.9	0.9
Dark surround	0.525	0.8	0.8

**Table A.1:** Input parameters for the CIECAM02 model.

- Saturation
- Equivalent cartesian coordinates.

The first step is a linear von Kries type chromatic adaptation transform (Section 2.5) converting CIE XYZ tristimulus values to RGB responses based on the transform matrix  $\mathbf{M}_{CAT02}$  as

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \mathbf{M}_{CAT02} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad (\text{A.1})$$

where

$$\mathbf{M}_{CAT02} = \begin{pmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{pmatrix} \quad (\text{A.2})$$

The degree of adaptation factor  $D$  is computed as

$$D = F \left[ 1 - \frac{1}{3.6} e^{\left( \frac{-L_A + 42}{92} \right)} \right], \quad (\text{A.3})$$

where  $L_A$  is the adapting luminance and  $F$  is chosen as reported in Table A.1. The responses  $RGB$  computed from Equation (A.1) are converted into adapted tristimulus responses  $R_C G_C B_C$

$$R_C = [(Y_w D / R_w) + (1 - D)]R \quad (\text{A.4})$$

$$G_C = [(Y_w D / G_w) + (1 - D)]G \quad (\text{A.5})$$

$$B_C = [(Y_w D / B_w) + (1 - D)]B, \quad (\text{A.6})$$

where the  $Y$  tristimulus value of the white is generally  $Y_w = 100$ .

A series of factors must be now computed, namely the luminance-level adaptation factor  $F_L$ , the induction factors  $N_{bb}$  and  $N_{cb}$ , and the base exponential nonlinearity  $z$  as follows:

$$k = \frac{1}{5L_A + 1}, \quad (\text{A.7})$$

$$F_L = 0.2k^4 5L_A + 0.1(1 - k^4)^2 (5L_A)^{\frac{1}{3}}, \quad (\text{A.8})$$

$$n = \frac{Y_b}{Y_n}, \quad (\text{A.9})$$

$$N_{bb} = N_{cb} = 0.725 \left( \frac{1}{n} \right)^{0.2}, \quad (\text{A.10})$$

$$z = 1.48 + \sqrt{n}. \quad (\text{A.11})$$

The  $R_C G_C B_C$  adapted responses must be converted from the MCAT02 specification to Hunt-Pointer-Estevéz fundamentals (Hunt and Pointer 1985) that more closely represent cone fundamentals as

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \mathbf{M}_{HPE} \mathbf{M}_{CAT02}^{-1} \begin{pmatrix} R_C \\ G_C \\ B_C \end{pmatrix}, \quad (\text{A.12})$$

where

$$\mathbf{M}_{HPE} = \begin{pmatrix} 0.38971 & 0.68898 & -0.07868 \\ -0.22981 & 1.18340 & 0.04641 \\ 0.00000 & 0.00000 & 1.00000 \end{pmatrix} \quad (\text{A.13})$$

and

$$\mathbf{M}_{CAT02}^{-1} = \begin{pmatrix} 1.096124 & -0.278869 & 0.182745 \\ 0.454369 & 0.473533 & 0.072098 \\ -0.009628 & -0.005698 & 1.015326 \end{pmatrix}. \quad (\text{A.14})$$

Nonlinearities are then applied to the  $R'G'B'$

$$R'_a = \frac{400(F_L R'/100)^{0.42}}{27.13 + (F_L R'/100)^{0.42}} + 0.1 \quad (\text{A.15})$$

$$G'_a = \frac{400(F_L G'/100)^{0.42}}{27.13 + (F_L G'/100)^{0.42}} + 0.1 \quad (\text{A.16})$$

$$B'_a = \frac{400(F_L B'/100)^{0.42}}{27.13 + (F_L B'/100)^{0.42}} + 0.1. \quad (\text{A.17})$$

These values are then used to compute **opponent color responses** ( $a, b$ )

$$a = R'_a - \frac{12G'_a}{11} + \frac{B'_a}{11} \quad (\text{A.18})$$

$$b = \frac{1}{9}(R'_a + G'_a - 2B'_a). \quad (\text{A.19})$$

and the color appearance correlates. **Hue** is then computed as

$$h = \arctan \frac{b}{a}. \quad (\text{A.20})$$

It is measured in degrees counterclockwise from the positive  $a$  axis defined by (A.18). An eccentricity factor is computed as

$$e_t = \frac{1}{4} \left[ \cos \left( h \frac{\pi}{180} + 2 \right) + 3.8 \right]. \quad (\text{A.21})$$

In order to compute lightness, one must first compute an initial achromatic response  $A$  for nonlinear adapted cone responses and for the white  $A_w$  according to

$$A = [2R'_a + G'_a + (1/20)B'_a - 0.305]N_{bb}. \quad (\text{A.22})$$

The **lightness**  $J$  is then computed as

$$J = 100 \left( \frac{A}{A_w} \right)^{cz}. \quad (\text{A.23})$$

The **brightness**  $Q$  is computed as

$$Q = (4/c) \sqrt{J/100} (A_w + 4) F_L^{0.25}, \quad (\text{A.24})$$

where  $A_w$  is the achromatic response for white and  $c$  is the surround factor determined from the relative luminance of the surround, the luminance-level adaptation factor  $F_L$  (A.8).

In order to compute **chroma**  $C$ , an intermediate value  $t$  must be computed as

$$t = \frac{(5 \cdot 10^4 / 13) N_c N_{cb} e_t \sqrt{a^2 + b^2}}{R'_a + G'_a + (21/20) B'_a}, \quad (\text{A.25})$$

then

$$C = t^{0.9} \sqrt{(J/100)} (1.64 - 0.29^n)^{0.73}. \quad (\text{A.26})$$

The **colorfulness**  $M$  is obtained from chroma as

$$M = CF_L^{0.25} \quad (\text{A.27})$$

and the **saturation**  $s$  is obtained from chroma and colorfulness as

$$s = 100 \sqrt{M/Q}. \quad (\text{A.28})$$

The cartesian coordinates for the chroma, colorfulness, and saturation dimensions are given as

$$a_C = C \cos(h) \quad (\text{A.29})$$

$$b_C = C \sin(h) \quad (\text{A.30})$$

$$a_M = M \cos(h) \quad (\text{A.31})$$

$$b_M = M \sin(h) \quad (\text{A.32})$$

$$a_s = s \cos(h) \quad (\text{A.33})$$

$$b_s = s \sin(h). \quad (\text{A.34})$$

The inverse model is summarized in the following procedure:

1. Calculate  $t$  from  $C$  and  $J$ .
2. Calculate  $e_t$  from  $h$ .
3. Calculate  $A$  from  $A_w$  and  $J$ .
4. Calculate  $a$  and  $b$  from  $t$ ,  $e_t$ ,  $h$ , and  $A$ .
5. Calculate  $R'_a$ ,  $G'_a$ , and  $B'_a$  from  $A$ ,  $a$ , and  $b$ .
6. Use the inverse nonlinearity to compute  $R'$ ,  $G'$ , and  $B'$ .
7. Convert to  $R_C$ ,  $G_C$ , and  $B_C$ , via linear transform.
8. Invert the chromatic adaptation transform to compute  $R$ ,  $G$ , and  $B$  and then  $X$ ,  $Y$ , and  $Z$ .

More detailed guidelines can be found in (CIE 2004).

## Appendix B

# Two-stage linear recovery for bilinear models

In (D’Zmura and Iverson 1993a), the authors use the information of several surfaces viewed under several illuminants to solve for color constancy. By combining this information into one system of equations, they retrieve the reflectance and illuminant descriptors. In this section, we present the derivation their model and its resolution.

### Bilinear model for a single view of a single surface

The authors use linear models for reflectance and illuminant spectra (Section 2.6), which are expressed respectively as

$$S(\lambda) = \sum_{j=1}^{N_\sigma} \sigma_j \mathcal{S}_j(\lambda) \quad (\text{B.1})$$

and

$$E(\lambda) = \sum_{i=1}^{N_\varepsilon} \varepsilon_i \mathcal{E}_i(\lambda). \quad (\text{B.2})$$

They also designate the light reflected from a surface by  $L(\lambda)$

$$L(\lambda) = E(\lambda)S(\lambda) = \sum_{i=1}^{N_\varepsilon} \sum_{j=1}^{N_\sigma} \varepsilon_i \sigma_j \mathcal{E}_i(\lambda) \mathcal{S}_j(\lambda) \quad (\text{B.3})$$

and write the sensor responses, that they refer to as “quantum catches,” as

$$\rho_k = \int R_k(\lambda) E(\lambda) S(\lambda) d\lambda = \int R_k(\lambda) \left\{ \sum_{i=1}^{N_\varepsilon} \sum_{j=1}^{N_\sigma} \varepsilon_i \sigma_j \mathcal{E}_i(\lambda) \mathcal{S}_j(\lambda) \right\} d\lambda. \quad (\text{B.4})$$

They define the  $N_\rho \times N_\varepsilon$  bilinear model matrices  $\mathbf{B}_j$  with entries

$$(\mathbf{B}_j)_{ki} = \int R_k(\lambda) \mathcal{E}_i(\lambda) \mathcal{S}_j(\lambda) d\lambda, \quad j = 1, \dots, N_\sigma. \quad (\text{B.5})$$

The  $N_\rho$  rows are indexed by  $k$ , the number of sensors, and the  $N_\varepsilon$  columns are indexed by  $n_\varepsilon$ , the number of illuminant descriptors. There are  $N_\sigma$  matrices  $\mathbf{B}_j$ , one for each reflectance basis function. The  $k^{\text{th}}$  sensor response to a surface viewed under one illuminant, referred as “one view of one surface,” is written as

$$\rho_k = \sum_{i=1}^{N_\sigma} \sum_{j=1}^{N_\varepsilon} \sigma_j (\mathbf{B}_j)_{ki} \varepsilon_i. \quad (\text{B.6})$$

### Bilinear model for multiple views of multiple surfaces

When considering multiple ( $N_E$ ) views of multiple ( $N_p$ ) surfaces, the response of the  $k^{\text{th}}$  sensor produced by the  $n_p^{\text{th}}$  surface viewed under the  $n_E^{\text{th}}$  illuminant is given by

$$\rho_{n_p n_E k} = \sum_{i=1}^{N_\sigma} \sum_{j=1}^{N_\varepsilon} \sigma_{n_p j} (\mathbf{B}_j)_{ki} \varepsilon_{n_E i}. \quad (\text{B.7})$$

Let us introduce the  $N_\rho$ -dimensional “data vector”

$$\mathbf{d}_{n_p n_E} = [\rho_{n_p n_E k}, \dots, \rho_{n_p n_E N_\rho}]^T \quad (\text{B.8})$$

and the  $N_\varepsilon$ -dimensional vector

$$\boldsymbol{\varepsilon}_{n_E} = [\varepsilon_{n_E 1}, \dots, \varepsilon_{n_E N_\varepsilon}]^T. \quad (\text{B.9})$$

For all  $n_p = 1, \dots, N_p$  and  $n_E = 1, \dots, N_E$ , (B.7) can be rewritten in the matrix form

$$\mathbf{d}_{n_p n_E} = \sum_{j=1}^{N_\sigma} \sigma_{n_p j} \mathbf{B}_j \boldsymbol{\varepsilon}_{n_E}. \quad (\text{B.10})$$

Let us now define the  $N_\rho N_E$  entries vector

$$\mathbf{d}_{n_p} = [\mathbf{d}_{n_p 1}^T \dots \mathbf{d}_{n_p N_E}^T], \quad (\text{B.11})$$

the  $N_\varepsilon N_E$  entries vector

$$\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_1^T \dots \boldsymbol{\varepsilon}_{N_E}^T], \quad (\text{B.12})$$

and the  $N_\rho N_E \times N_\varepsilon N_E$  block diagonal matrices whose  $N_E$  blocks are  $\mathbf{B}_j$

$$\mathbf{C}_j = \text{diag}[\mathbf{B}_j, \dots, \mathbf{B}_j]. \quad (\text{B.13})$$

Equation (B.10) becomes

$$\mathbf{d}_{n_p} = \sum_{j=1}^{N_\sigma} \sigma_{n_p j} \mathbf{C}_j \boldsymbol{\varepsilon}. \quad (\text{B.14})$$

### Two-stage linear recovery procedure

For simplicity, the authors treat the case where  $N_p = N_\sigma$  and  $N_\rho = N_\varepsilon$  and extend it to  $N_p > N_\sigma$  afterwards. The matrices  $\mathbf{B}_j$  and  $\mathbf{C}_j$  are thus square. Let us consider the square matrix  $\mathbf{R}$  formed by the reflectance descriptors  $\sigma_{n_p j}$

and designate the entries of its inverse  $\mathbf{R}^{-1}$  by  $\tilde{\sigma}_{jn_p}$ . Eq. (B.14) becomes

$$\sum_{n_p=1}^{N_p} \tilde{\sigma}_{jn_p} \mathbf{C}_j^{-1} \mathbf{d}_{n_p} = \boldsymbol{\varepsilon} \quad (\text{B.15})$$

The surface sensor responses  $\mathbf{d}_{n_p}$  are combined in a  $N_\rho N_e \times N_\sigma$  “data matrix”

$$\mathbf{D} = [\mathbf{d}_1 \dots \mathbf{d}_{N_\sigma}] \quad (\text{B.16})$$

and the inverse reflectance  $\tilde{\sigma}_{jn_p}$  into the  $N_\sigma$  entries column vectors containing the inverse reflectance

$$\tilde{\boldsymbol{\sigma}}_j = [\tilde{\sigma}_{j1} \dots \tilde{\sigma}_{jN_\sigma}]^T. \quad (\text{B.17})$$

The system (B.15) becomes

$$\mathbf{C}_j^{-1} \mathbf{D} \tilde{\boldsymbol{\sigma}}_j = \boldsymbol{\varepsilon}, \quad j = 1, \dots, N_\sigma. \quad (\text{B.18})$$

If we now define the  $N_\rho N_e \times N_\sigma$  matrices  $\mathbf{F}_j$  as

$$\mathbf{F}_j = \mathbf{C}^{-1} \mathbf{D}, \quad j = 1, \dots, N_\sigma, \quad (\text{B.19})$$

it can be written as

$$\mathbf{F}_1 \tilde{\boldsymbol{\sigma}}_1 = \dots = \mathbf{F}_{N_\sigma} \tilde{\boldsymbol{\sigma}}_{N_\sigma} = \boldsymbol{\varepsilon} \quad (\text{B.20})$$

or equivalently as

$$\mathbf{F}_1 \tilde{\boldsymbol{\sigma}}_1 - \mathbf{F}_2 \tilde{\boldsymbol{\sigma}}_2 = \dots = \mathbf{F}_1 \tilde{\boldsymbol{\sigma}}_1 - \mathbf{F}_{N_\sigma} \tilde{\boldsymbol{\sigma}}_{N_\sigma} = 0. \quad (\text{B.21})$$

The last formula can be rewritten in the form of a homogeneous system of equations as follows

$$\mathbf{F} \tilde{\boldsymbol{\sigma}} = 0, \quad (\text{B.22})$$

where  $\mathbf{F}$  is the  $(N_\sigma - 1)N_\rho N_e \times N_\sigma^2$  matrix

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 & -\mathbf{F}_2 & 0 & \dots & 0 \\ \mathbf{F}_1 & 0 & -\mathbf{F}_3 & & 0 \\ \vdots & & & \ddots & \\ \mathbf{F}_1 & 0 & & & -\mathbf{F}_{N_\sigma} \end{pmatrix} \quad (\text{B.23})$$

and  $\tilde{\boldsymbol{\sigma}}$  is a  $N_\sigma^2$ -dimensional column vector given by

$$\tilde{\boldsymbol{\sigma}} = [\tilde{\boldsymbol{\sigma}}_1^T \dots \tilde{\boldsymbol{\sigma}}_{N_\sigma}^T]. \quad (\text{B.24})$$

The reflectance descriptors are computed by inversion of the linear system (B.22) returning the inverse reflectance descriptors  $\tilde{\sigma}_j$  followed by the inversion of the matrix  $\mathbf{R}^{-1}$ . The illuminant descriptors contained in  $\boldsymbol{\varepsilon}$  are then computed as

$$\mathbf{F}_j \tilde{\boldsymbol{\sigma}}_j = \boldsymbol{\varepsilon} \quad (\text{B.25})$$

for any  $j \in [1, N_\sigma]$ .

In the case where  $N_p > N_\sigma$ , the matrices  $\mathbf{B}_j$  are not square anymore and the system cannot be solved as presented above. This can be solved by simply

---

adding entries to the matrices  $\mathbf{B}_j$  to make them square. These supplementary entries must be linearly dependent to the existing columns.



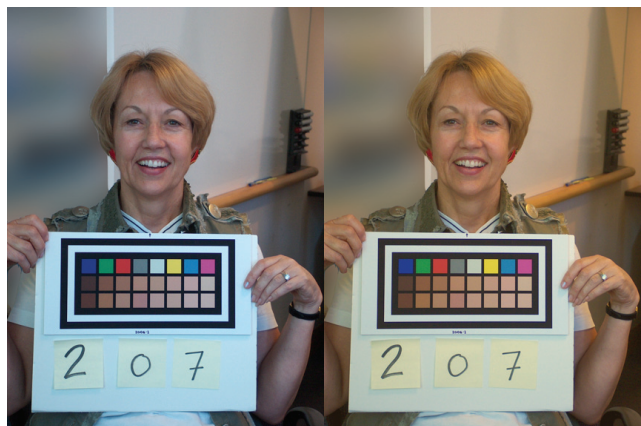
## Appendix C

# Supplementary material for Chapter 3

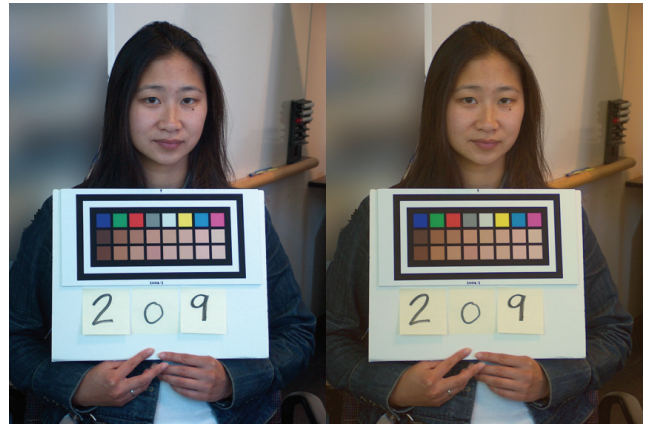
### C.1 Images of the subjects' faces

The images of the 53 subjects studied in Chapter 3 are presented in the following pages. The images are taken with a HP 850 camera under uncontrolled illumination conditions. Each figure shows the uncorrected image (left) and the image corrected with a color transform computed using the 16 skin colored patches (right).

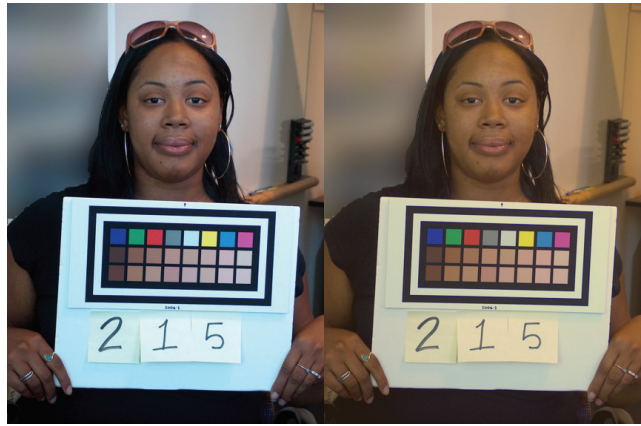








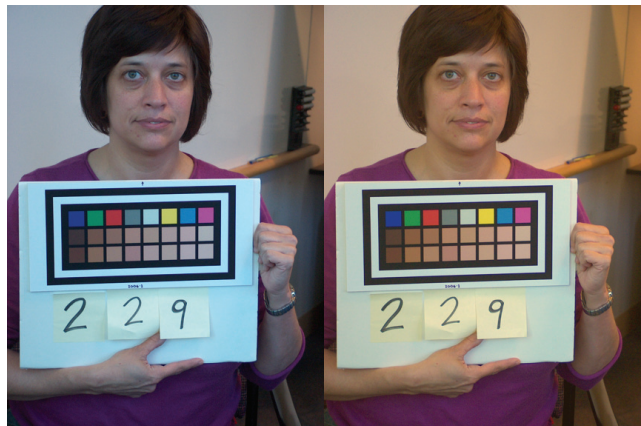








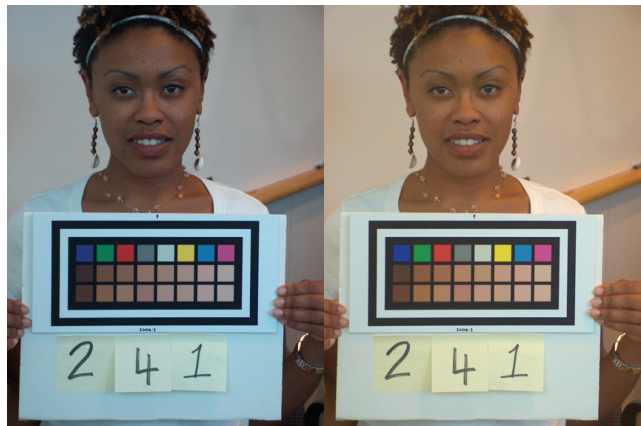
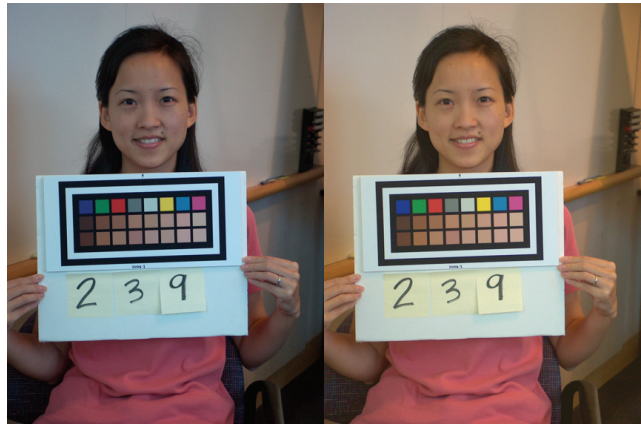




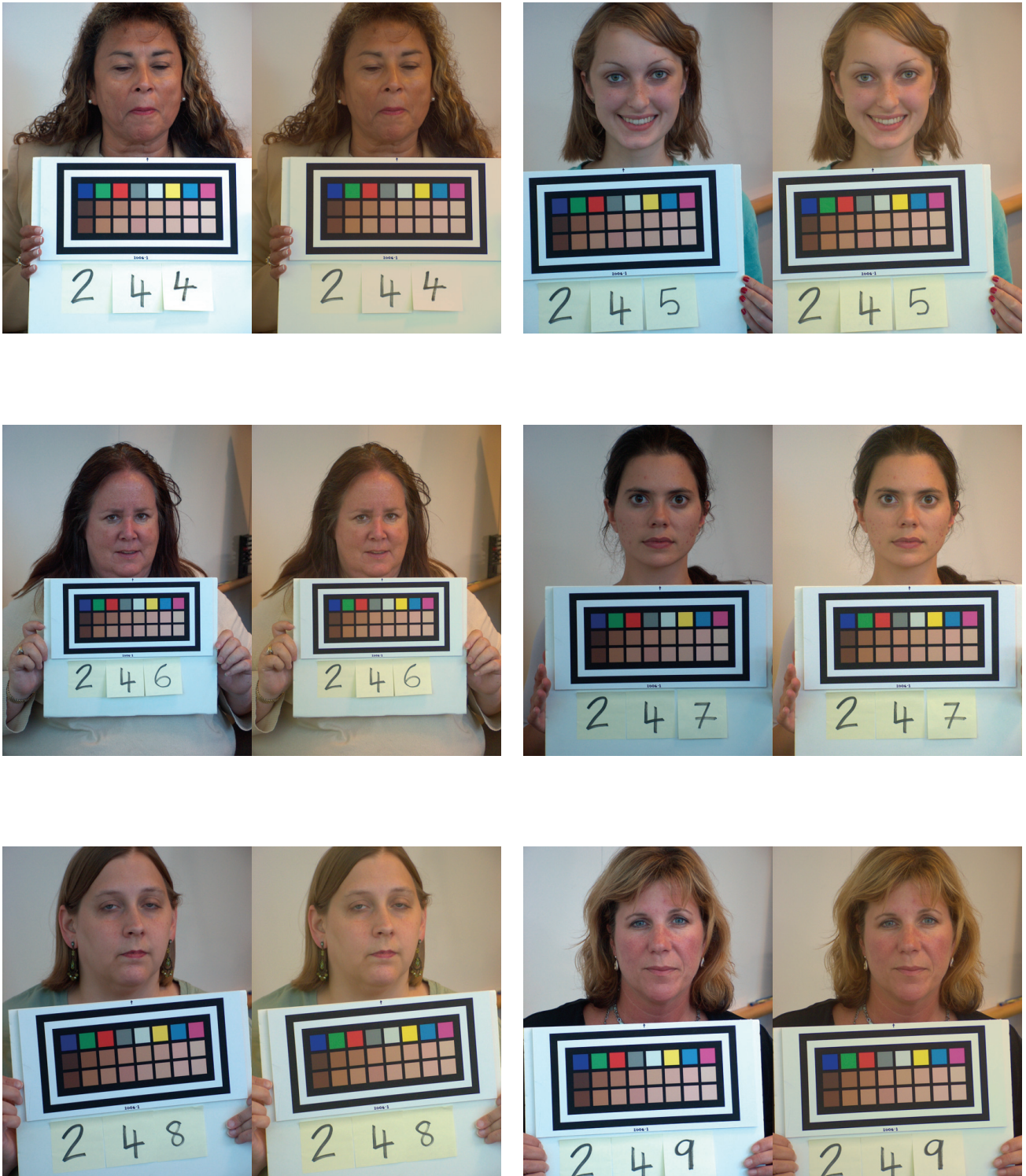






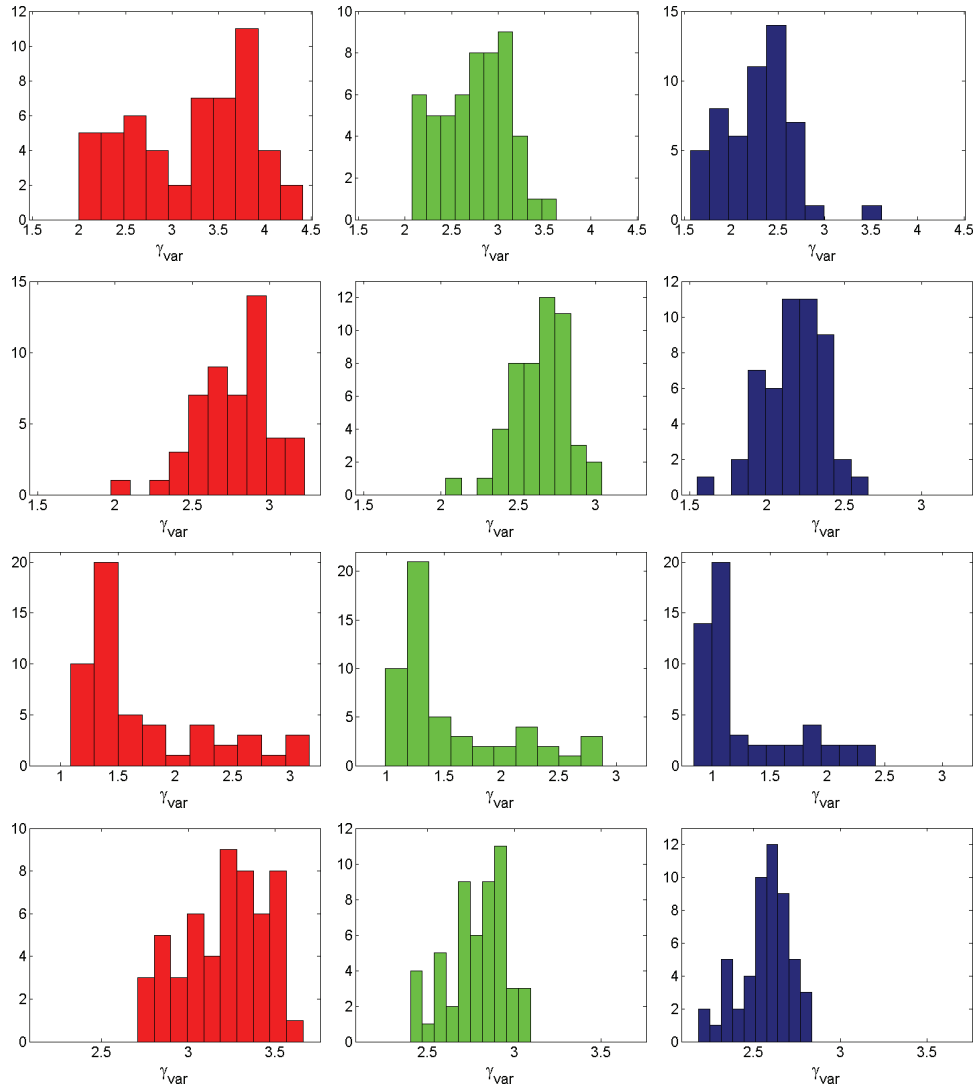








C.2  $\gamma_{\text{var}}$  values for Section 3.4.2



**Figure C.1:**  $\gamma_{\text{var}}$  for, from top two bottom, the HP 850, Canon S400, Nikon D1, and Nokia 6820 cameras and for, from left to right, the red, green, and blue channels.

### C.3 Kruskal-Wallis $p$ -values for Section 3.4.2

	HP	Canon	Nikon	Nokia
HP	0.040	$2.446 \cdot 10^{-08}$	$6.017 \cdot 10^{-13}$	0
Canon	$1.942 \cdot 10^{-07}$	$1.111 \cdot 10^{-03}$	$1.419 \cdot 10^{-11}$	0
Nikon	$3.052 \cdot 10^{-11}$	$7.566 \cdot 10^{-08}$	0.012	$5.932 \cdot 10^{-05}$
Nokia	$2.220 \cdot 10^{-16}$	0	$1.055 \cdot 10^{-13}$	0.872

**Table C.1:**  $p$ -values for the comparison of the transform computed with nonlinear sRGB (NL) values against the one computed with linear sRGB (L1,  $\gamma = 2.4$ ).

	HP	Canon	Nikon	Nokia
HP	0.707	$5.787 \cdot 10^{-11}$	$2.801 \cdot 10^{-11}$	0
Canon	$1.833 \cdot 10^{-03}$	$3.771 \cdot 10^{-08}$	$9.157 \cdot 10^{-08}$	0
Nikon	$4.754 \cdot 10^{-10}$	$5.392 \cdot 10^{-07}$	$2.319 \cdot 10^{-07}$	$1.813 \cdot 10^{-04}$
Nokia	$1.110 \cdot 10^{-16}$	0	0	0.607

**Table C.2:**  $p$ -values for the comparison of the transform computed with nonlinear sRGB (NL) values against the one computed with linear sRGB (L2,  $\gamma_{\text{var}}$ ).

	HP	Canon	Nikon	Nokia
HP	0.029	$3.250 \cdot 10^{-07}$	$9.117 \cdot 10^{-13}$	$1.110 \cdot 10^{-16}$
Canon	$6.148 \cdot 10^{-13}$	$1.483 \cdot 10^{-03}$	$6.490 \cdot 10^{-08}$	0
Nikon	$3.606 \cdot 10^{-13}$	$3.102 \cdot 10^{-03}$	$2.935 \cdot 10^{-07}$	0
Nokia	$2.220 \cdot 10^{-16}$	0	$4.149 \cdot 10^{-13}$	0.665

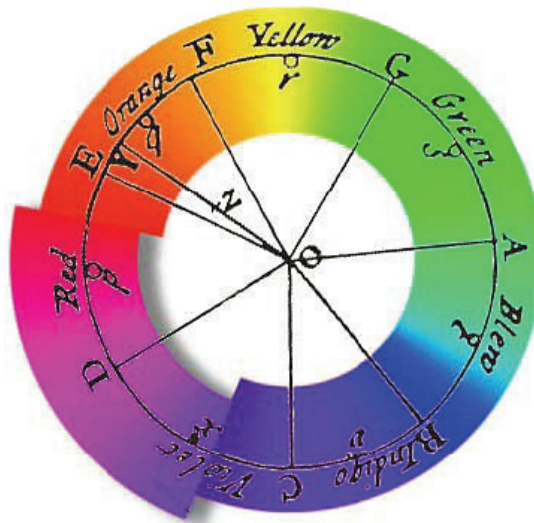
**Table C.3:**  $p$ -values for the comparison of the transform computed with linear sRGB (L1,  $\gamma = 2.4$ ) values against the one computed with linear sRGB (L2,  $\gamma_{\text{var}}$ ).



## Appendix D

# Historical review of color harmony theories

Color harmony is defined as two or more colors seen in neighboring areas that produce a pleasing effect (Judd and Wyszecki 1975). While this defines color harmony, it does not explain how such harmonies can be created. This section presents a short historical review of different color harmony theories.



**Figure D.1:** Newton arranged the spectral colors on a hue circle. Missing colors are those obtained by mixtures of red and blue. Source: Westland *et al.* (2007).

Newton first understood that white light could be separated into a set of monochromatic colored components using a prism. He rejected the widely accepted Aristotle idea that all colors arise from black and white. He identified seven spectral colors - red, orange, yellow, green, blue, indigo, and violet and

noticed that the two ends of the visible spectrum were similar. He thus arranged the spectral colors on a circle (see Figure D.1), which was not closed at the time, as certain hues cannot be obtained with a single monochromatic light. He also understood that this representation allowed to geometrically predict hue and saturation of light mixtures. While he worked mostly with additive mixtures of lights, artists worked with subtractive mixtures of pigments. It probably led to a confusion between the additive and subtractive mixtures theories and to the many primary colors definitions and resulting hue circles. The basic idea of a color wheel is the arrangement of opposed hues on a circle. Nowadays, it is known that the optimal additive and subtractive primaries are respectively red, green, and blue, and yellow, cyan, and magenta. These primaries are not a consequence of the physical properties of light, but are due to the physiological response of the human visual system. Secondary hues of the additive RGB color system are hues obtained by adding two primaries: yellow is *red + green*, magenta is *red + blue*, and cyan is *green + blue*. *Tertiary hue* refers to hues created by mixing a primary and a secondary hue or two secondary hues. Depending on the context, the term *primary* is also used to indicate the *psychovisual primary hues* or *pure hues* red, green, yellow, and blue.

Goethe (1810), known as one of the pioneers of color harmony theory, completely rejected Newton's work and objected that science and mathematics could not explain colors<sup>1</sup>. He believed that harmonious colors must be opposed to each other on a hue circle and conceptualized the concept of *completing colors* - named *complementary colors* nowadays.

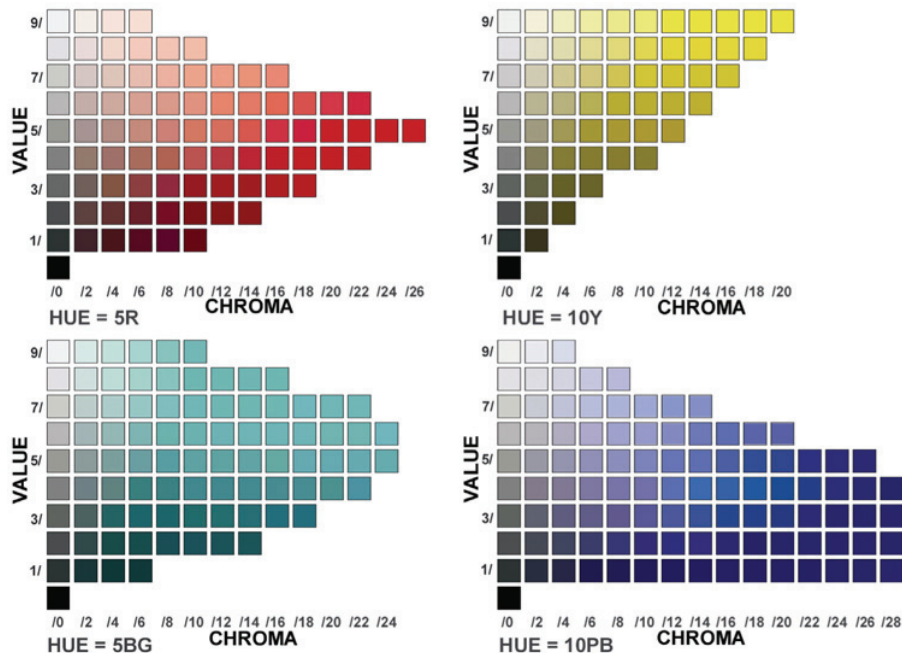
Chevreul (1839) created a circle composed of 64 hues derived from three primary hues, red, yellow, and blue<sup>2</sup>. He identified two types of color harmonies: *analogous harmonies* consist of colors having either neighboring lightness and saturation and the same hue or neighboring hue and the same lightness and saturation; *contrasted harmonies* consist of colors having either lightness and saturation far apart and the same hue or hue far apart and the same lightness and saturation. He seems to be the first to have used contrasts of lightness and saturation in addition to contrasts of hues.

Munsell, Ostwald (1916), and Itten (1961) are three important contributors from the early 19th century. Their work has in common that color harmonies are defined using either a color solid or a color order system to represent geometrical relations between colors. Munsell (1905) developed a color representation system based on the perceptual attributes of colors: hue, chroma, and lightness. His system is a solid formed by approximately perceptually spaced colored samples, the *Munsell Colors*. The central vertical axis represents achromatic colors ordered uniformly by increasing lightness, named *Munsell Value*. Colored samples are ordered such that their chroma, named *Munsell Chroma*, increases when they are placed further away from the achromatic axis. Each "wedge slice" of the solid represents a hue. The samples were presented in a book: one hue per page and ordered by increasing chroma - from left to right - and increasing lightness - from bottom to top. The solid formed by the color samples corresponds to real colors and is thus not symmetrical. A saturated

<sup>1</sup>Newton was actually accused to have destroyed the beauty of the rainbow by explaining the origin of its colors and supposedly inspired John Keats' poem, *Lamia*.

<sup>2</sup>Artists' primaries are still red, yellow, and blue, probably historically due to the colors of available pigments.

yellow is indeed lighter than an equally saturated blue, as shown in Figure D.2.

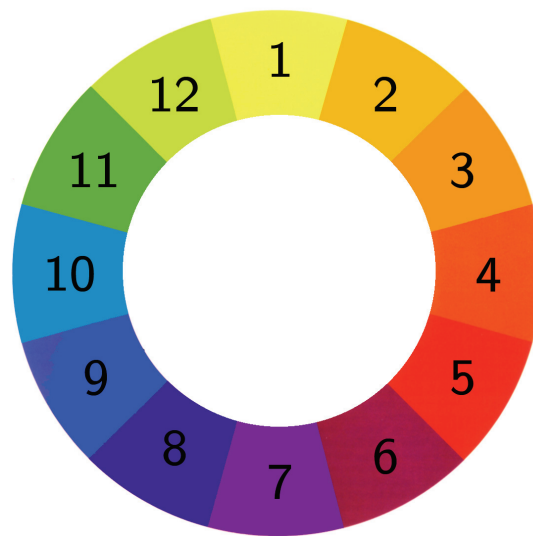


**Figure D.2:** Example of Munsell hues: colors are ordered from left to right by increasing chroma and from bottom to top by increasing lightness. Munsell solid is not symmetrical. Source: Malacara-Hernandez (2002)

The success of the Munsell system is in the description of colors in terms of their perceptual attributes in a uniform color system. Munsell also defined rules for color harmonies using paths in the space defined by his ordering system. He also defined the “strength” of a color as the product of its Munsell chroma and value. He also suggested that if a color is stronger than another, its area should be inversely proportional to its strength for the two colors to remain balanced.

Moon and Spencer (1944) attempted to provide a formal framework for the existing color harmony theories. They took advantage of the then recent advances in colorimetry and used Munsell colors CIE XYZ tristimulus values (Section 2.4.1) to specify color harmonies rules geometrically. They introduced a Euclidian metric on an  $\omega$ -space and transformed  $(X, Y, Z)$  coordinates into the cylindrical coordinates  $(\omega_1 = r \cos \theta, \omega_2 = z, \omega_3 = r \sin \theta)$  that they supposed to correlate with chroma ( $r$ ), lightness ( $z$ ), and hue ( $\theta$ ). They tested various color harmony schemes defined by varying one, two, or three of the variables  $\omega$  and by descriptions such as “points on a straight line” or “n points on a circle with center on neutral axis.” This experiment was not very successful and may have produced better results using a perceptual color space such as CIELAB created three decades later.

In contemporary color theories, several color combinations based on 12 hues



**Figure D.3:** Most contemporary color theories agree on several basic combinations of 12 tertiary hues.

are commonly accepted as harmonious (refer to Figure D.3):

**Monochromatic harmonies** consist of colors having the same hue.

**Complementary harmonies** consist of colors having opposed hues (e.g., hues 1 and 7).

**Analogous harmonies** consist of colors having similar hues (e.g., hues 1, 2, and 3).

**Split-complementary harmonies** consist of two colors of similar hues facing a third color (e.g., hues 1, 6, and 8).

This summarizes the commonly accepted codes of color harmony. However, these rules do not take into account the more subjective aspects of color, such as emotion or symbolism. Red, orange, and yellow are generally considered as *warm colors* and associated warmth, fire, or even blood and anger. Blue, purple, and green are considered as *cool colors* and are associated with calm, air, or even cold and indifference. Some believe that color and color associations have an influence on our mood and health. It must also be noted that we reviewed color harmony theories from an occidental point of view, whereas color symbols can be completely different in other cultures. For example, Chinese brides marry in red; white, the color of mourning, would certainly be inappropriate for a wedding.



## Appendix E

# Supplementary material for Chapter 4

### E.1 Images of decoration samples and complementing color palettes

The 63 decoration samples used in Chapter 4 are presented in the following pages. The images were taken under fluorescent light with a HP R-967 camera. Each figure represents the uncorrected image (left), the corrected image (middle), and the palette returned by the system. Each palette contains four paint samples, the top left color corresponds to the color best matching the samples as returned by the system (right). The palettes were generated from paint reflectance spectra measured with an Eye-One Pro spectrometer used to compute sRGB values. Under each image, we reported  $\Delta V = \frac{V_{i+o} - V_i}{V_i}$  (4.3) giving an estimate of the distance between the reference target values and the object's color and the grade assigned by the expert to the system's match.



Figure E.1:  $\Delta V = 0.21$ , grade = 2



Figure E.2:  $\Delta V = 0.01$ , grade = 1



Figure E.3:  $\Delta V = 0.12$ , grade = 1



Figure E.4:  $\Delta V = 0.01$ , grade = 1



Figure E.5:  $\Delta V = 0.31$ , grade = 1



Figure E.6:  $\Delta V = 0.21$ , grade = 3



Figure E.7:  $\Delta V = 0.48$ , grade = 1



Figure E.8:  $\Delta V = 0.02$ , grade = 1



Figure E.9:  $\Delta V = 0.06$ , grade = 1



Figure E.10:  $\Delta V = 0.00$ , grade = 1





Figure E.11:  $\Delta V = 0.01$ , grade = 2



Figure E.12:  $\Delta V = 0.01$ , grade = 1



Figure E.13:  $\Delta V = 0.06$ , grade = 2



Figure E.14:  $\Delta V = 0.00$ , grade = 1



Figure E.15:  $\Delta V = 0.21$ , grade = 1



Figure E.16:  $\Delta V = 0.00$ , grade = 2



Figure E.17:  $\Delta V = 0.00$ , grade = 1



Figure E.18:  $\Delta V = 0.01$ , grade = 1



Figure E.19:  $\Delta V = 0.00$ , grade = 2



Figure E.20:  $\Delta V = 0.10$ , grade = 1



Figure E.21:  $\Delta V = 0.04$ , grade = 1

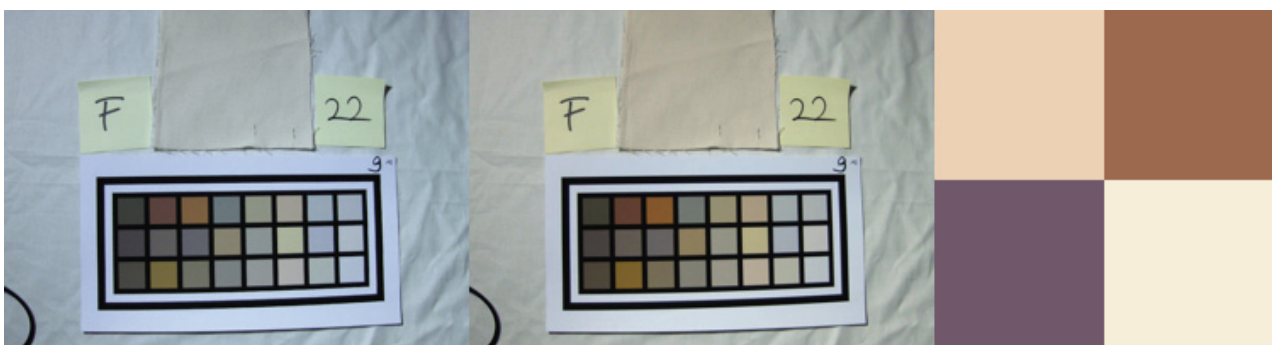


Figure E.22:  $\Delta V = 0.00$ , grade = 2





Figure E.23:  $\Delta V = 0.20$ , grade = 1



Figure E.24:  $\Delta V = 0.00$ , grade = 2



Figure E.25:  $\Delta V = 0.07$ , grade = 3



Figure E.26:  $\Delta V = 0.00$ , grade = 1



Figure E.27:  $\Delta V = 0.21$ , grade = 3



Figure E.28:  $\Delta V = 0.20$ , grade = 5



Figure E.29:  $\Delta V = 0.00$ , grade = 3



Figure E.30:  $\Delta V = 0.16$ , grade = 3



Figure E.31:  $\Delta V = 0.00$ , grade = 1

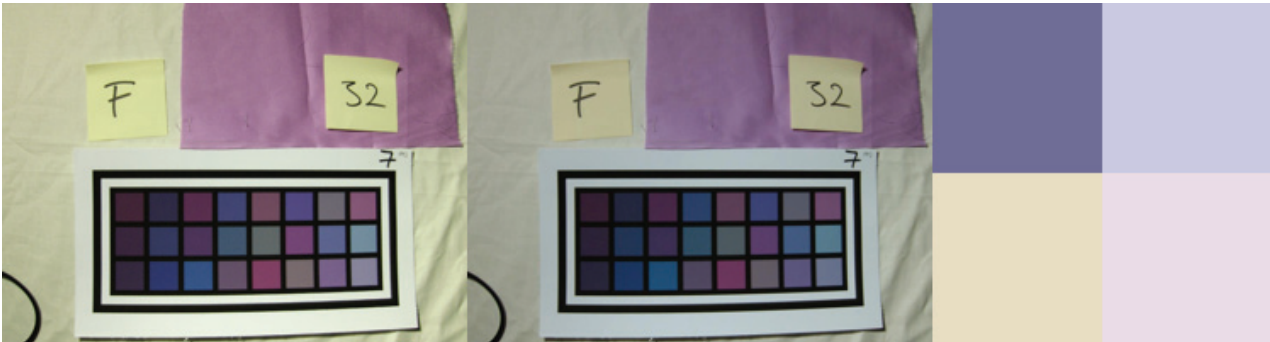


Figure E.32:  $\Delta V = 0.09$ , grade = 5



Figure E.33:  $\Delta V = 0.04$ , grade = 1

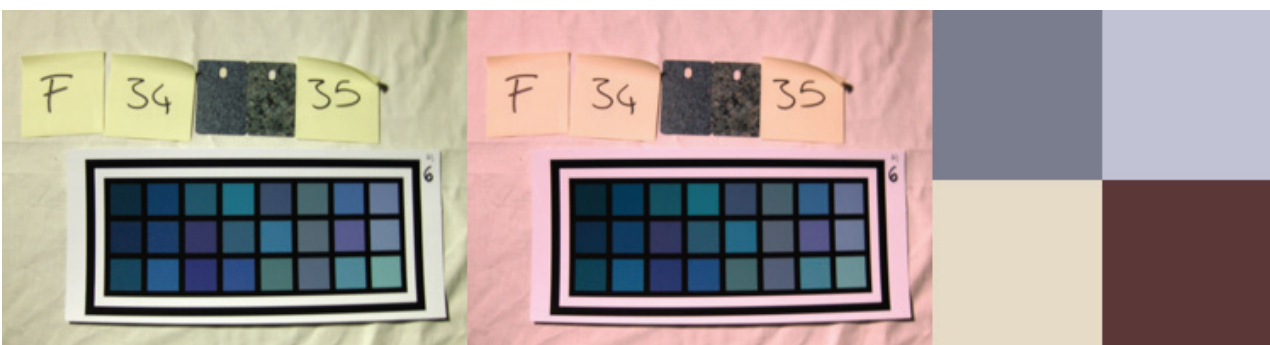


Figure E.34:  $\Delta V = 0.17$ , grade = 4



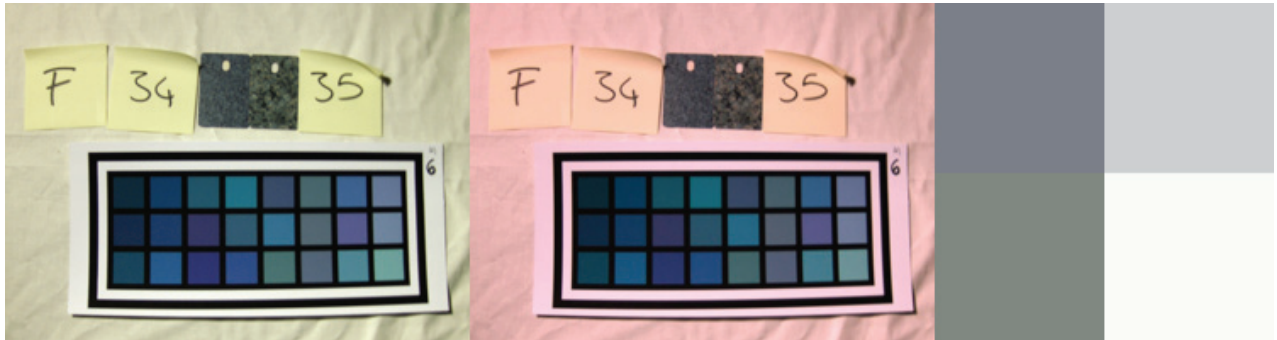


Figure E.35:  $\Delta V = 0.29$ , grade = 3



Figure E.36:  $\Delta V = 0.05$ , grade = 4

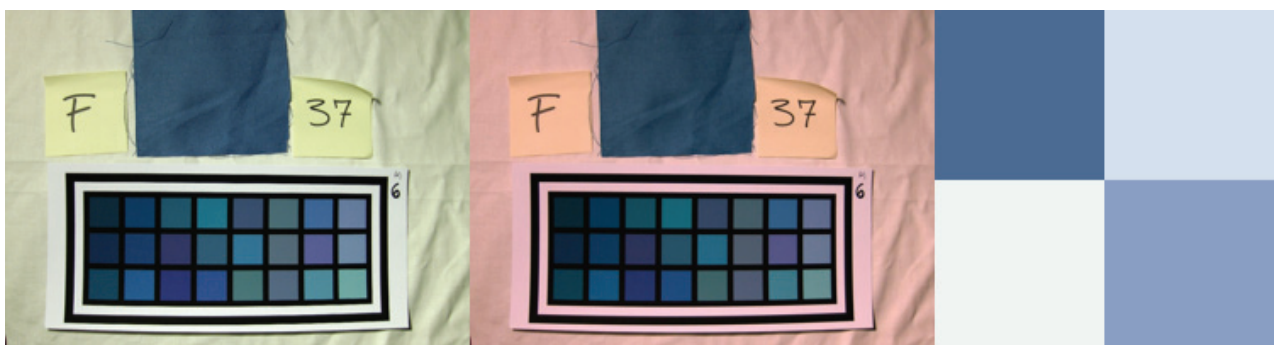


Figure E.37:  $\Delta V = 0.00$ , grade = 2

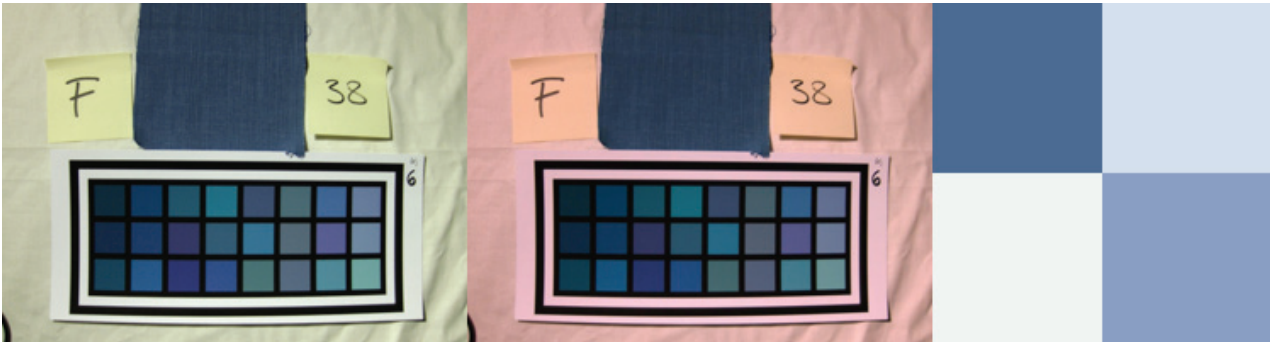


Figure E.38:  $\Delta V = 0.01$ , grade = 1

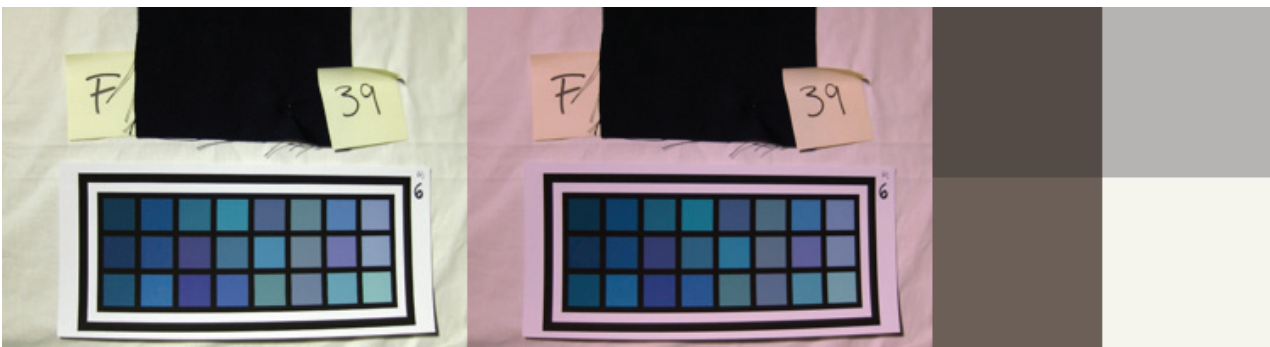


Figure E.39:  $\Delta V = 0.41$ , grade = 5



Figure E.40:  $\Delta V = 0.04$ , grade = 5

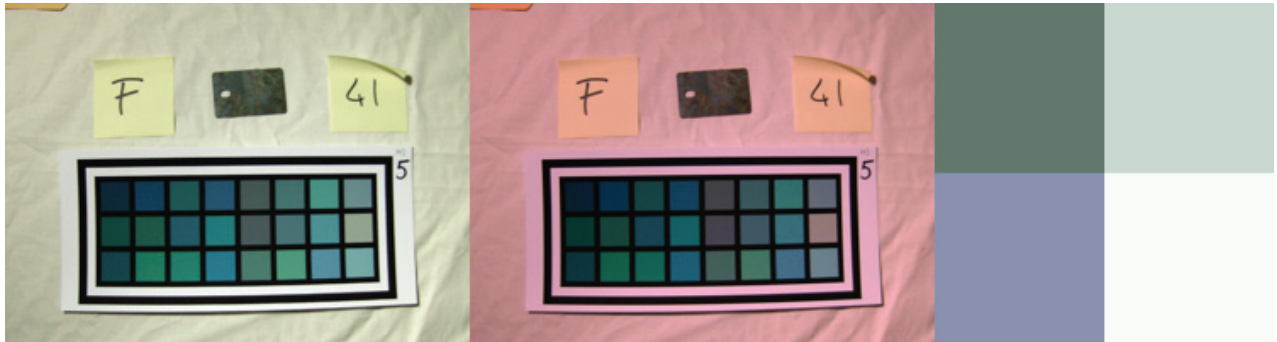


Figure E.41:  $\Delta V = 0.19$ , grade = 2

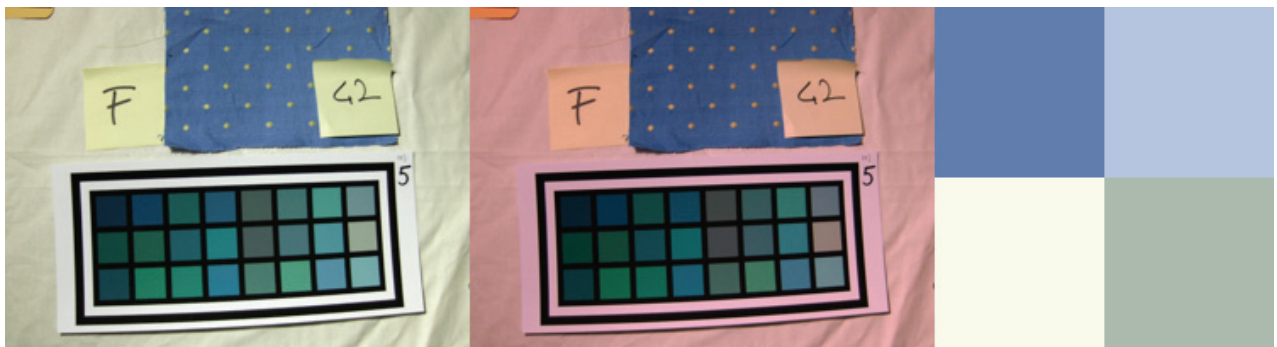


Figure E.42:  $\Delta V = 0.29$ , grade = 2



Figure E.43:  $\Delta V = 0.09$ , grade = 2

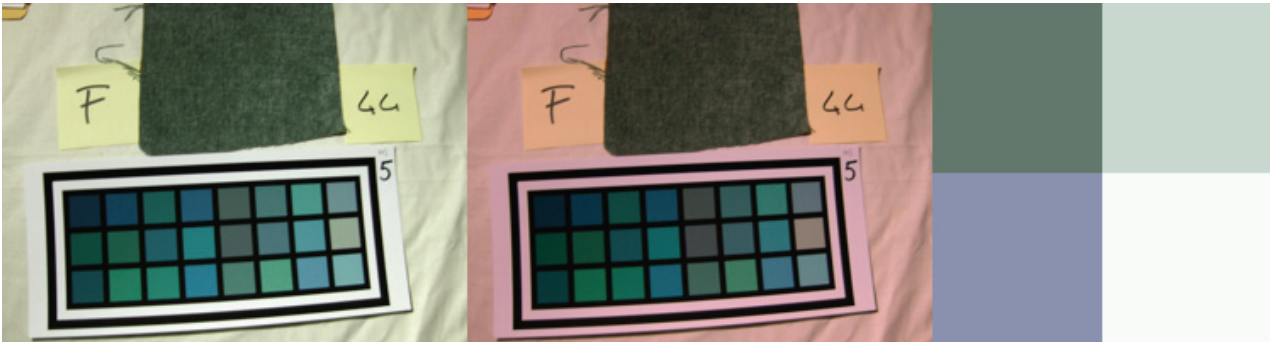


Figure E.44:  $\Delta V = 0.14$ , grade = 1

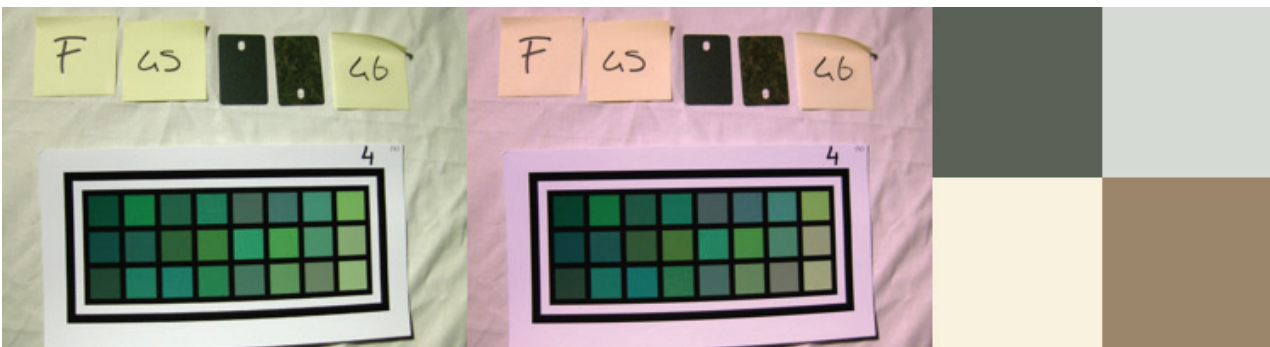


Figure E.45:  $\Delta V = 0.21$ , grade = 1

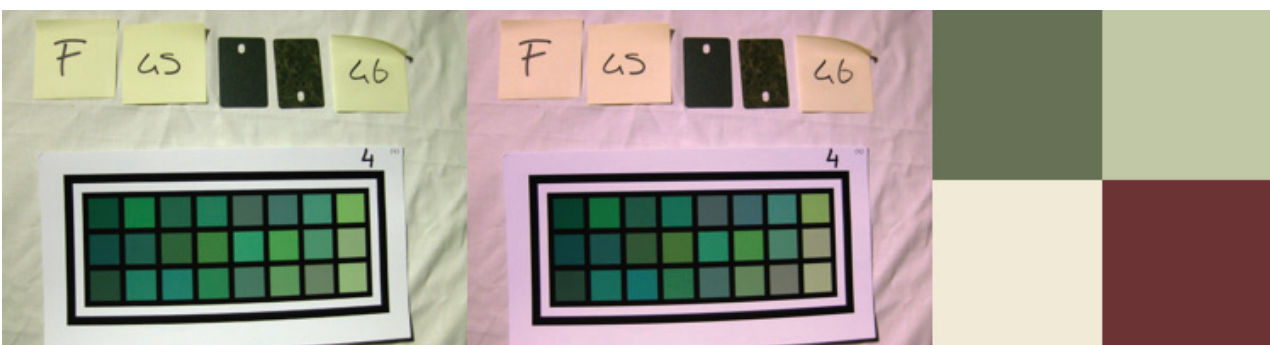


Figure E.46:  $\Delta V = 0.31$ , grade = 1



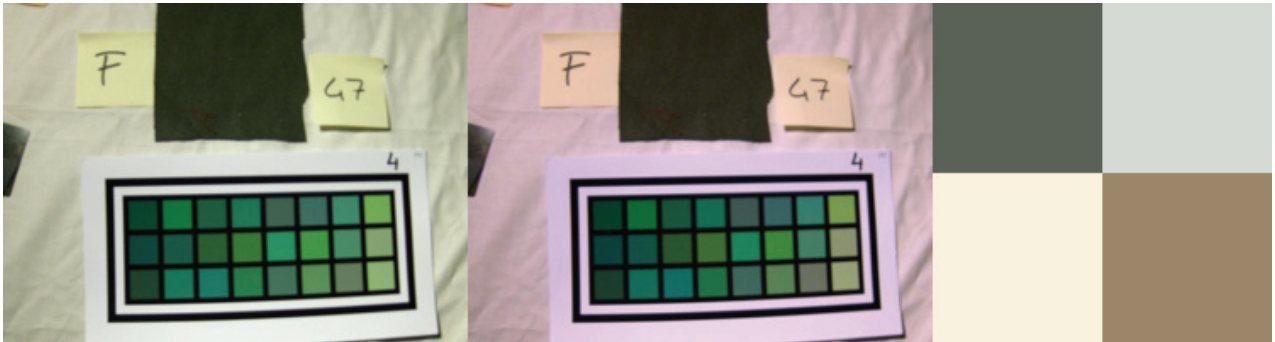


Figure E.47:  $\Delta V = 0.27$ , grade = 1



Figure E.48:  $\Delta V = 0.20$ , grade = 5



Figure E.49:  $\Delta V = 0.05$ , grade = 1

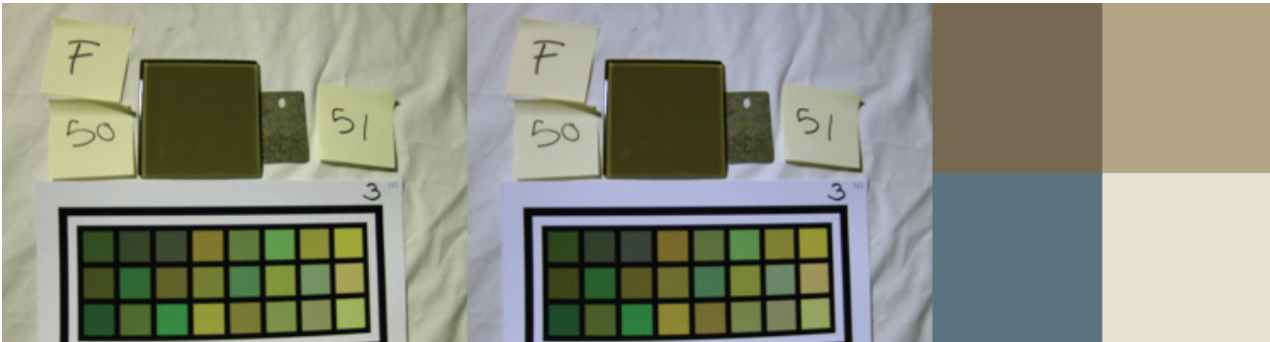


Figure E.50:  $\Delta V = 0.04$ , grade = 5

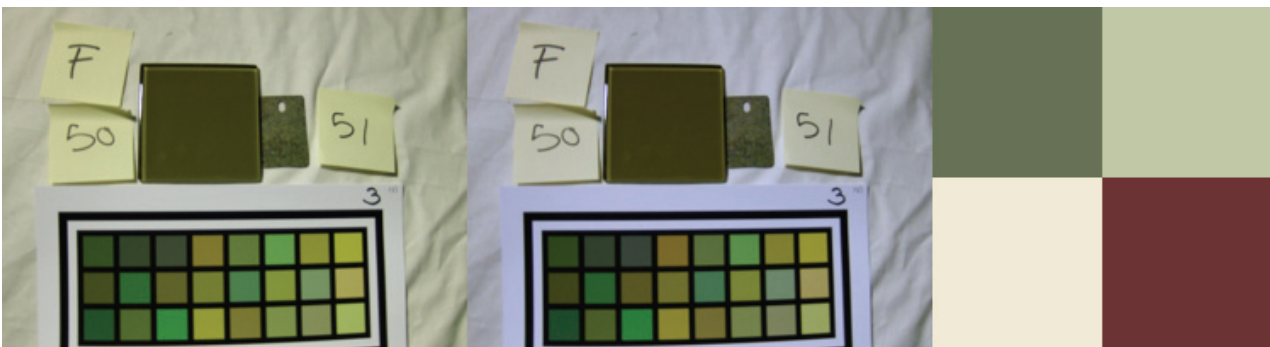


Figure E.51:  $\Delta V = 0.01$ , grade = 2

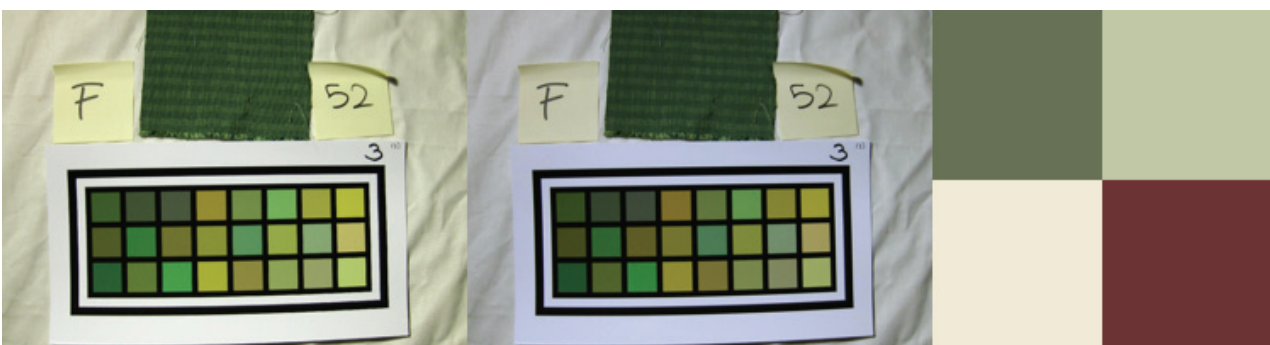


Figure E.52:  $\Delta V = 0.01$ , grade = 3

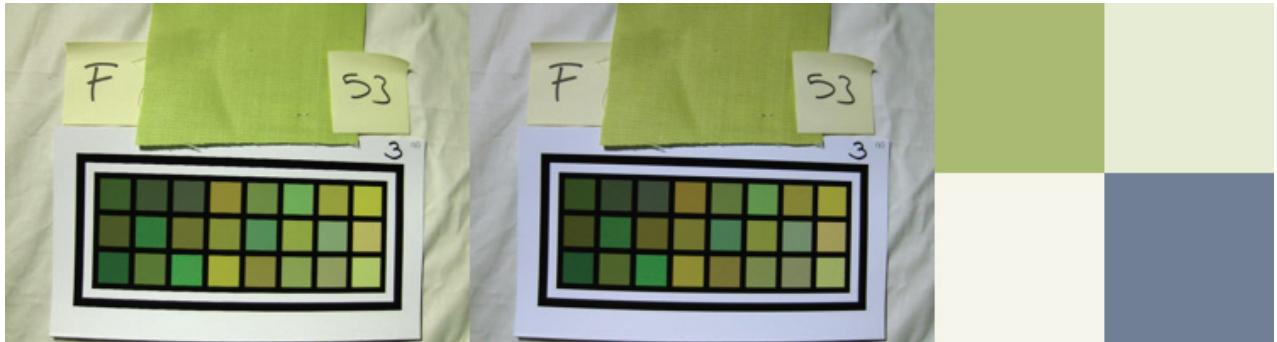


Figure E.53:  $\Delta V = 0.00$ , grade = 1



Figure E.54:  $\Delta V = 0.20$ , grade = 1

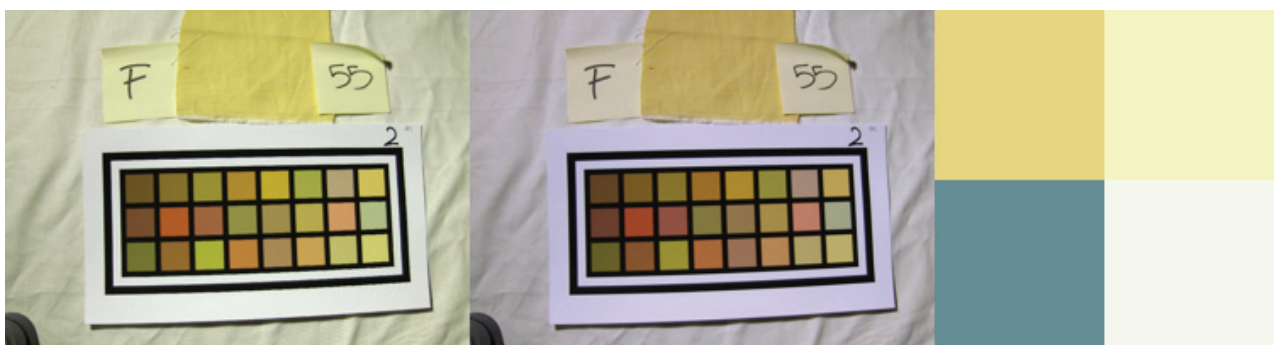


Figure E.55:  $\Delta V = 0.00$ , grade = 2

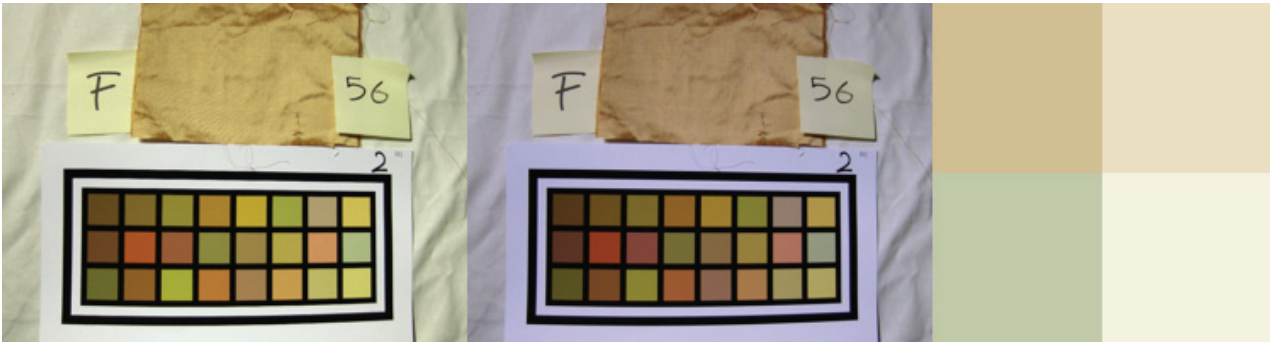


Figure E.56:  $\Delta V = 0.00$ , grade = 2

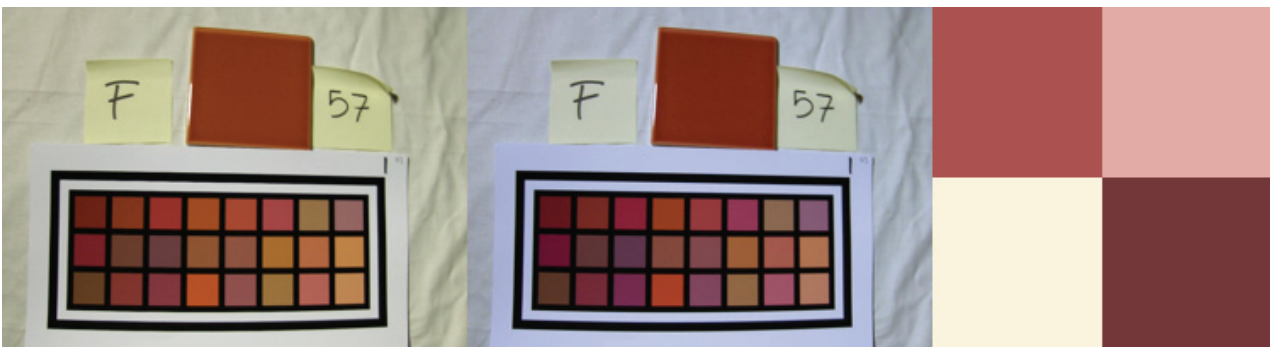


Figure E.57:  $\Delta V = 0.04$ , grade = 5



Figure E.58:  $\Delta V = 0.07$ , grade = 2





Figure E.59:  $\Delta V = 0.12$ , grade = 4



Figure E.60:  $\Delta V = 0.15$ , grade = 2



Figure E.61:  $\Delta V = 0.00$ , grade = 1



Figure E.62:  $\Delta V = 0.15$ , grade = 3



Figure E.63:  $\Delta V = 0.12$ , grade = 1

## Appendix F

# Supplementary material for Chapter 5

### F.1 MatLab code

The algorithm for the illuminant retrieval method for fixed location cameras presented in Chapter 5 consists of the following steps:

1. Select the illuminant type (standard daylight, real daylight, or artificial), the number of images (or illuminants)  $N$ , and the number of patches  $N_p$ .
2. Load the patch pixel values
  - (a) by loading an image and selecting manually  $N_p$  patches in the images
  - (b) or by using MacBeth, Munsell, or Natural synthetic patches values.
3. Minimize the error function  $f_e$  (5.12)
  - (a) by testing all combinations of  $N$  illuminants from a set of  $N_e$  illuminant candidates
  - (b) or by minimizing  $f_e$  by gradient descent
4. Return the  $N$  illuminants minimizing  $f_e$ .

It was coded in MatLab. Three main functions are used in the code:

- `fnIllu2Ref1Descriptors` takes the illuminant descriptors  $\boldsymbol{\varepsilon}$ , the pixel values  $\mathbf{p}$ , the illuminants basis functions  $\boldsymbol{\mathcal{E}}$ , the reflectance basis functions  $\boldsymbol{\mathcal{S}}$ , and the camera sensitivities  $\mathbf{r}$  as input and returns the reflectance descriptors  $\boldsymbol{\sigma}$  for the  $N_p$  surfaces. The code is displayed below.
- `fnRef12rho_D65` takes the illuminants basis functions  $\boldsymbol{\mathcal{E}}$ , the reflectance basis functions  $\boldsymbol{\mathcal{S}}$ , and the surface reflectance descriptors  $\boldsymbol{\sigma}$  computed by `fnIllu2Ref1Descriptors` as input and returns the color responses  $\mathbf{p}^{D65}$  of the surfaces viewed under the illuminant D65 (5.11) simply computed as (5.10).

- `fnEuclidianErrorMultiImages` takes the color responses  $\mathbf{p}^{D65}$  of all surfaces computed by `fnRefl2rho_D65` and returns the error (5.12) computed over all pairs of images. The input argument `PairedIndices` is a list of images (or illuminants) number pairs as they should be compared in the double sum of the error function (5.12). For example, if  $N = 3$ , it contains the pairs (1,2), (1,3), and (2,3). The code is displayed below.

### `fnIllu2ReflDescriptors`

```

1 function s = fnIllu2ReflDescriptors(e1, e2, e3, PixelValues, S, E, R)

2 NReflDesp = size(S, 2);
3 NBPatches = size(PixelValues,1);
4 NImages = size(PixelValues,2)/3;
5 s = zeros(NReflDesp, NBPatches*NImages);

6 n = 1;
7 while n <= NImages
8     M = R' *diag(Illu)* S(:,1: NReflDesp);
9     pinvM = pinv(M);
10    for p = 1:NBPatches
11        s(:,p +(n-1)*NBPatches) = pinvM* PixelValues(p,3*n-2:3*n)';
12    end
13    n = n+ 1;
14 end

```

### `fnEuclidianErrorMultiImages`

```

1 function Error = ...
    fnEuclidianErrorMultiImages(rho_D65, NBPatches, NImages, PairedIndices)

% normalization (comment if needed)
2 rho_D65 = rho_D65./repmat(sqrt(sum(rho_D65.^2, 2)), 1, 3);

3 err_tmp = [];
4 for j = 1:size(PairedIndices,1)
    % indices in rho_D65 that correspond to the paired images
5     p1 = PairedIndices(j,1); ind1 = (p1-1)*NBPatches + 1: p1*NBPatches;
6     p2 = PairedIndices(j,2); ind2 = (p2-1)*NBPatches + 1: p2*NBPatches;
    % euclidian distance
7     err_tmp = [err_tmp; (rho_D65(ind1,:) - rho_D65(ind2,:)).^2];
8 end
9 Error = sqrt(sum(err_tmp(:)))/(NBPatches*NImages);

```

For any set of  $N$  test illuminants indexed by  $\mathbf{E}_k = (\mathbf{e}_1(k), \dots, \mathbf{e}_N(k))$ , the error  $f_e$  (5.12) can be computed using these three functions. To find the illuminants combination  $\mathbf{E}_k$  satisfying

$$\arg \min_k f_e(k), \quad (\text{F.1})$$



one must either compute  $f_e(k)$  for a set of illuminant candidates and return  $k$  for which  $f_e$  is minimum or minimize  $f_e$  by gradient descent.

### Gradient descent

When  $f_e$  is minimized by gradient descent, the illuminants are represented by one angle  $\theta$  (standard daylight illuminants), two angles  $(\theta, \phi)$ , or the three illuminants descriptors  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  (real daylight illuminants). The function `fnIllu2Ref1Descriptors` takes the illuminant descriptors  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  as input. When the illuminants are expressed using polar angles, they must first be converted into cartesian coordinates using a function `fnCoordPolar2Cartesian` (line 3). Below is the portion of the code performing the gradient descent in the case of standard daylight illuminants.

```

1  for j = 1:NbImages
    % Error(Theta + deltaTheta)
2  new_theta = theta + eps(:,j);
3  [e1, e2, e3] = ...
    fnCoordPolar2Cartesian(1, new_theta, pi/2-fnTheta2Phi(new_theta));
4  s = fnIllu2Ref1Descriptors(e1, e2, e3, PixelValues, S, E, R);
5  rho_D65 = fnRef12rho_D65(S, R, s);
6  Error_plus(j) = ...
    fnEuclidianErrorMultiImages(rho_D65, NbPatches, NbImages, PairedIndices);
7 end
8 for j = 1:NbImages
    % Error(Theta - deltaTheta)
9  new_theta = theta - eps(:,j);
10 [e1, e2, e3] = ...
    fnCoordPolar2Cartesian(1, new_theta, pi/2-fnTheta2Phi(new_theta));
11 s = fnIllu2Ref1Descriptors(e1, e2, e3, PixelValues, S, E, R);
12 rho_D65 = fnRef12rho_D65(S, R, s);
13 Error_minus(j) = ...
    fnEuclidianErrorMultiImages(rho_D65, NbPatches, NbImages, PairedIndices);
14 end

15 grad = Error_plus - Error_minus;

22 sign_grad_new = sign(grad);
23 theta = theta - (alpha.*grad)';
24 if sum(abs(sign_grad_old-sign_grad_new))> 0
25     eps = 0.98*eps;
26 end
27 sign_grad_old = sign_grad_new;

```

Lines 22 and 24 to 27 help suppressing oscillations - detected by the change of sign of `grad` - by reducing the norm of the gradient computation step size  $\epsilon$ , expressed by the variable `eps` in the code.

When the illuminant is expressed with polar and azimuthal angles  $(\theta, \phi)$ , the variable `theta` containing  $N \times 1$  angles  $\theta$  is replaced by the variable `angle = [theta,phi]` containing  $N$  angles  $\theta$  and  $N$  angles  $\phi$  and line 3 is replaced by

---

```
[e1, e2, e3] = fnCoordPolar2Cartesian(1, angle_new(1:NbImages), ...  
    pi/2-angle_new(NbImages+1:2*NbImages));
```

When the illuminant is expressed using  $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$  lines 3 and 10 are simply suppressed. Lines 1 to 27 are repeated until the norm of the gradient is smaller than a fixed threshold, coded as `while norm(grad) < norm_grad`.

### F.1.1 Parameters

The parameters  $\alpha$  and  $\epsilon$  were determined by trial-and-error. We started by running the algorithm for each new number of images  $N$  and patches  $N_p$  on a dozen of randomly chosen illuminant sets for arbitrarily large  $\alpha$  and  $\epsilon$  values, which were decreased until the illuminant descriptors remained in the range of values representing actual illuminants and converged to reasonable results. However, it is not possible to find optimal fixed parameters that would yield to an optimal convergence for all the images. Every set of patches and illuminants spectra brings different RGB patches values and thus define a new error function, whose optimal parameters may be different. The parameters used in the simulations of Sections 5.3.1 and 5.3.2 are reported in Tables F.1 to F.3.

$N$	$N_p$	$\alpha$	$\epsilon$	norm_grad
2	6	10.00	0.05	$10^{-4}$
3	6	15.00	0.05	$10^{-4}$
4	6	15.00	0.05	$10^{-4}$
6	6	15.00	0.05	$10^{-4}$
6	10	25.00	0.05	$10^{-4}$
10	6	25.00	0.05	$10^{-4}$

**Table F.1:** Parameters for the minimization by gradient descent for standard illuminants. They correspond to the results presented in Tables 5.3 and 5.4.

$N$	$N_p$	$\alpha$	$\epsilon$	norm_grad
2	6	15.00	0.01	$10^{-5}$
3	6	30.00	0.01	$10^{-5}$
4	6	50.00	0.01	$10^{-5}$
6	6	50.00	0.01	$10^{-5}$
6	10	50.00	0.01	$10^{-5}$
10	6	50.00	0.01	$10^{-5}$

**Table F.2:** Parameters for the minimization by gradient descent for real illuminants.  $f(\theta, \phi)$ . They correspond to the results presented in Table 5.7.

---

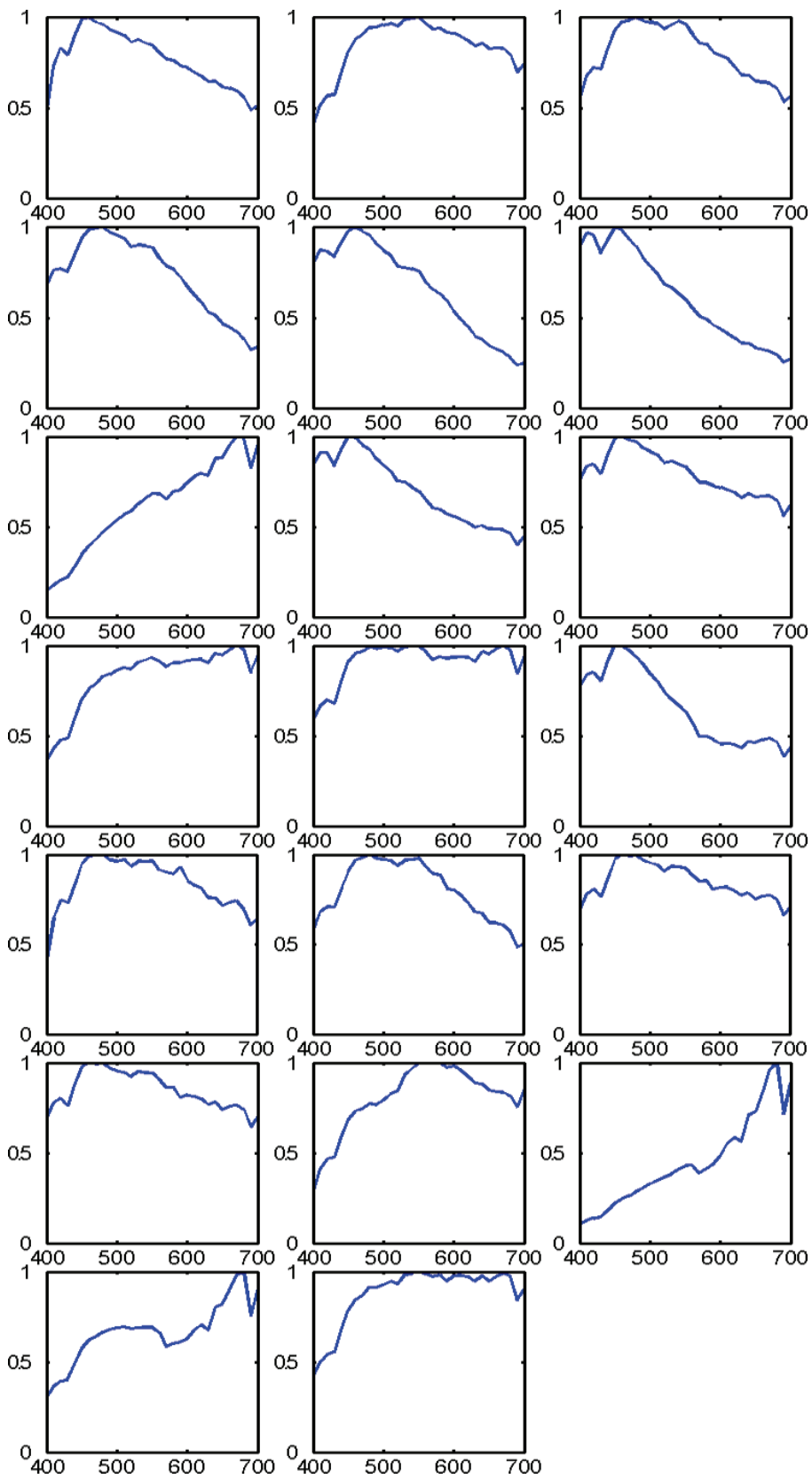
$N$	$N_p$	$\alpha$	$\epsilon$	norm_grad
2	6	50.00	0.01	$10^{-5}$
3	6	50.00	0.01	$10^{-5}$
4	6	50.00	0.01	$10^{-5}$
6	6	50.00	0.01	$10^{-5}$
6	10	50.00	0.01	$10^{-5}$
10	6	100.00	0.01	$10^{-5}$

**Table F.3:** Parameters for the minimization by gradient descent for real illuminants.  $f(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ . They correspond to the results presented in Table 5.8.

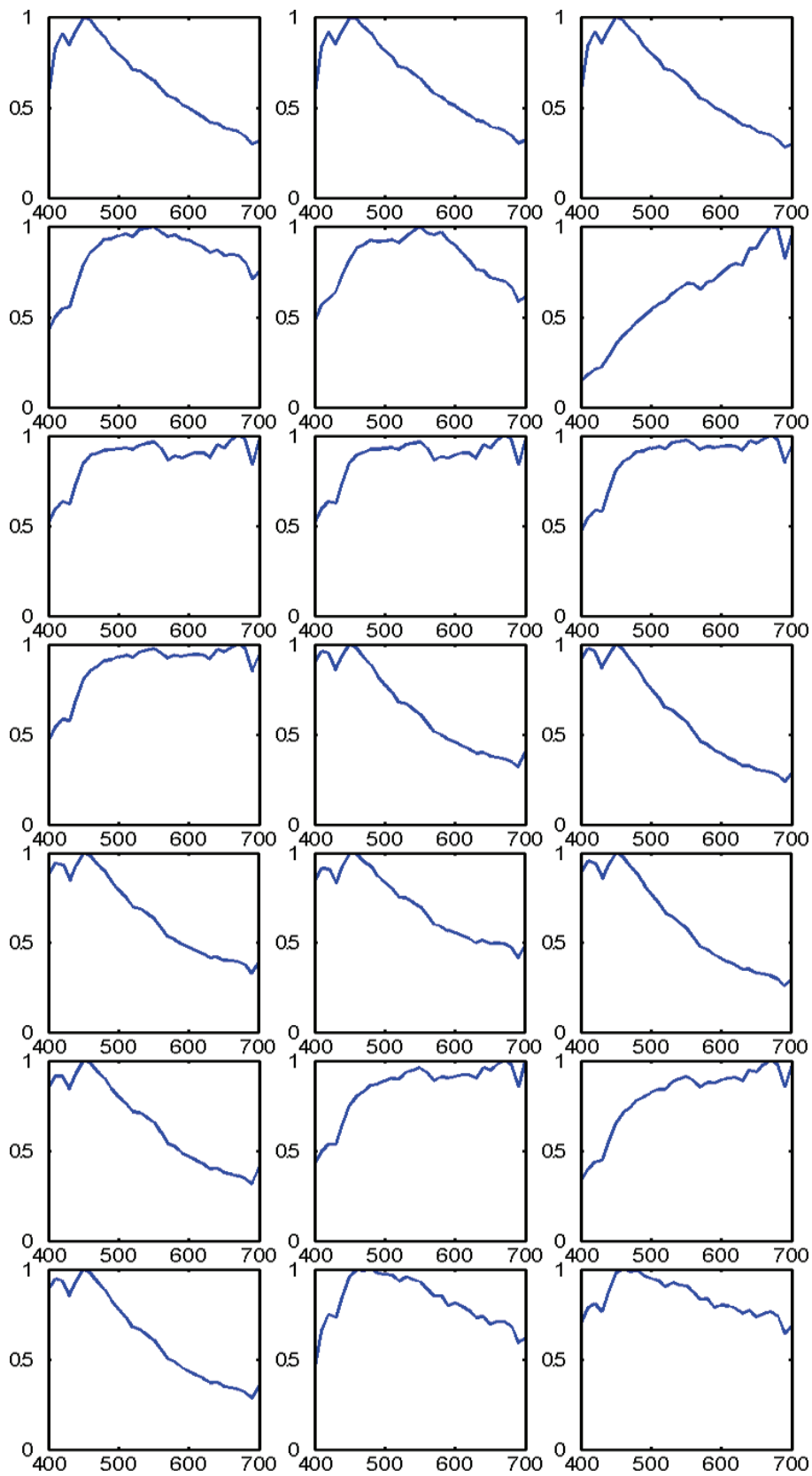
---

## F.2 Test illuminants

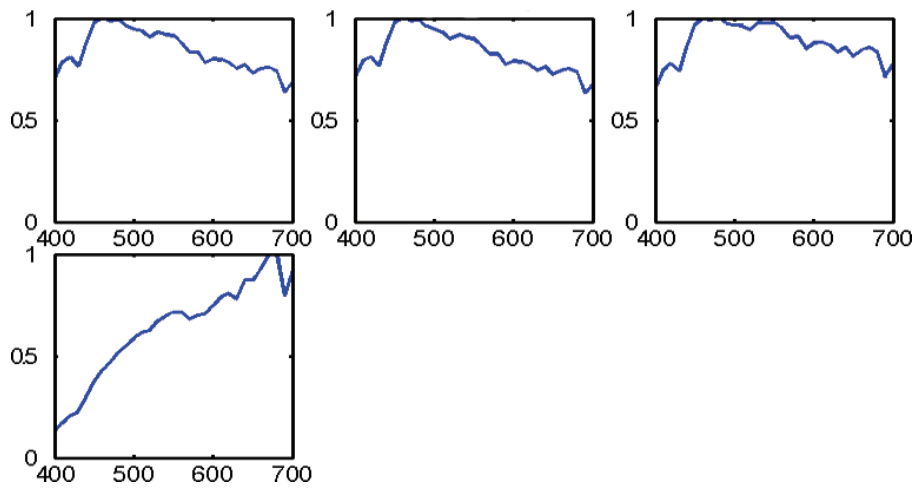
The simulations presented in Sections 5.3.2 and 5.3.6 used 45 and 36 illuminant spectra, respectively. Subsets of  $N_e = 20$  illuminants were chosen as illuminant candidates. The spectral power distributions of these illuminants are shown in Figures F.1 to F.4. Table F.4 presents the description of the artificial illuminants selected to run the experiments in Section 5.3.6. The descriptions are reported as provided in the illuminant database. We added measurements of our laboratory’s lightbooth, a GTI ColorMatcher, designated by “IVRG lightbooth.”



**Figure F.1:** Test daylight illuminants, subset of  $N_e = 20$  spectral power distributions used as reference illuminants. The x-axis represents the wavelength in nanometers. The y-axis represents the relative power.



**Figure F.2:** Test daylight illuminants. The x-axis represents the wavelength in nanometers. The y-axis represents the relative power.



**Figure F.2:** Test daylight illuminants. The x-axis represents the wavelength in nanometers. The y-axis represents the relative power.



	Illuminant description
1	F2 <sup>1</sup>
2	F7 <sup>1</sup>
3	F11 <sup>1</sup>
4	Daylight <sup>2</sup>
5	Spectrumlite <sup>2</sup>
6	Daylight D50 filters <sup>3</sup>
7	CWF <sup>3</sup>
8	Horizon <sup>3</sup>
9	Incandescent <sup>3</sup>
10	Daylight with d65 filters <sup>3</sup>
11	Grahams's desk at 2:00pm with blue sky AND overhead <sup>4</sup>
12	Lab 56 <sup>5</sup>
13	CF <sup>5</sup>
14	Daylight ultra 2 <sup>5</sup>
15	Halogen <sup>5</sup>
16	Philips universal home light <sup>5</sup>
17	Solux bulb 3500K
18	IVRG Lightbooth fluorescent TL 83
19	IVRG Lightbooth fluorescentD65
20	IVRG Lightbooth incandescent A
21	Coolwhite <sup>2</sup>
22	MacBeth booth SpectraLite cwf
23	Lab overhead <sup>4</sup>
24	Xenon lamp spectra from Greeley
25	Brillo fluorescent <sup>5</sup>
26	CWD <sup>5</sup>
27	KBU fluorescent <sup>5</sup>
28	US residential Florescent <sup>5</sup>
29	Philips econo-watt alta <sup>5</sup>
30	Philips worklite <sup>5</sup>
31	Philips home cool <sup>5</sup>
32	Solux bulb 3758K
33	Solux bulb 4000K
34	Solux bulb 4500K
35	Solux bulb 4700K
36	IVRG Lightbooth fluorescent TL84

**Table F.4:** Artificial lights descriptions. The illuminant spectra 1 to 20 and 21 to 36 are plotted in Figures F.3 and F.4, respectively.

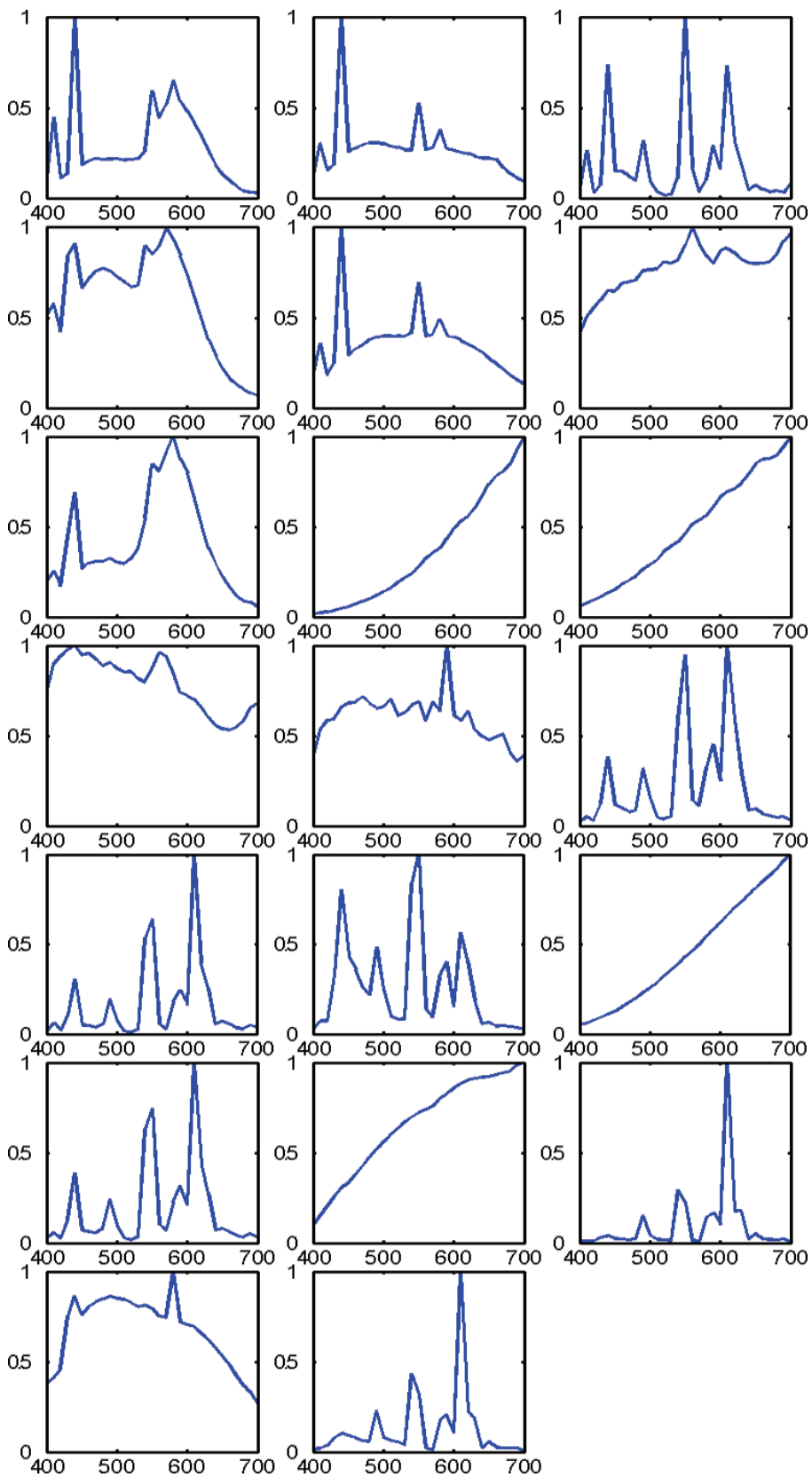
<sup>1</sup>Digitized from Hunt's "Reproduction of Color"

<sup>2</sup>Digitized from Sylvania phosphor mixtures

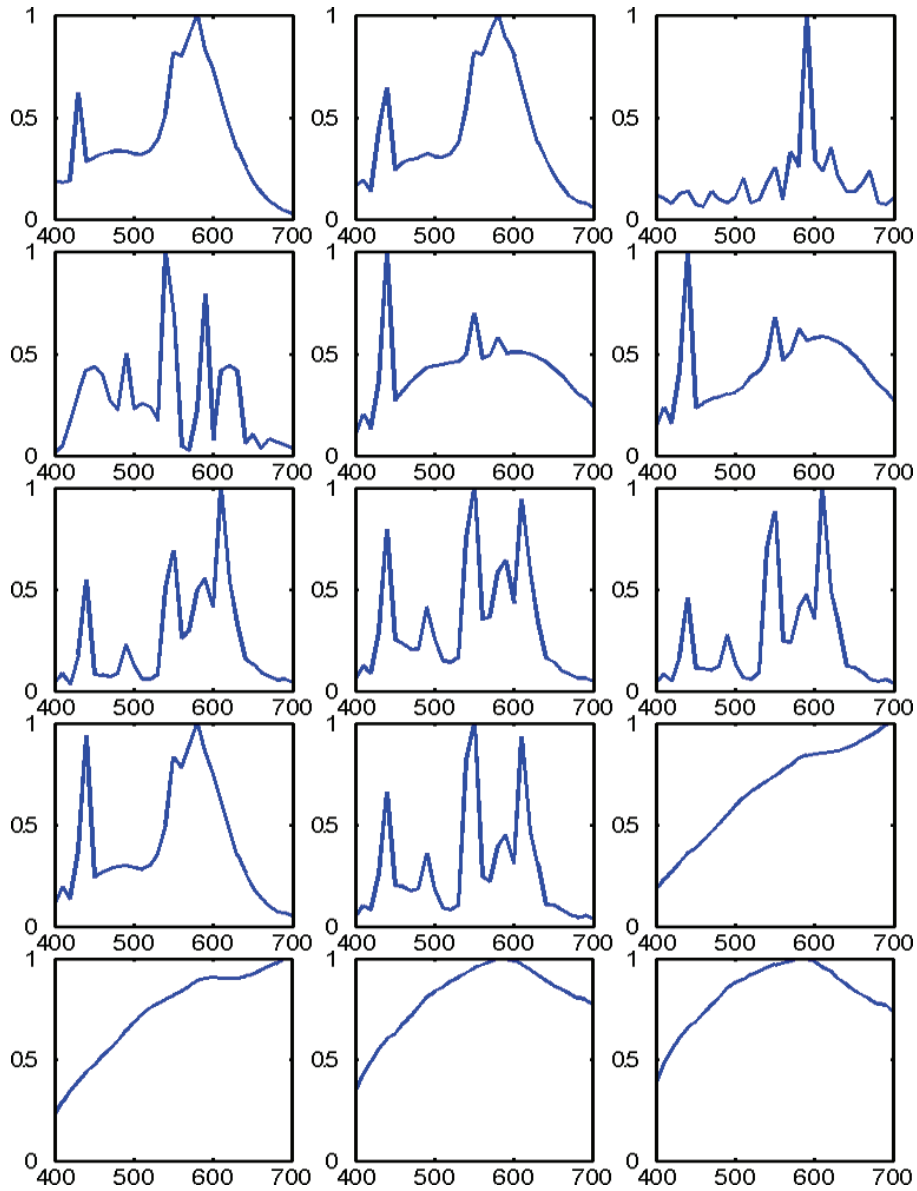
<sup>3</sup>MacBeth booth SpectraLite

<sup>4</sup>Simon Fraiser University's dataset

<sup>5</sup>Various fluorescents

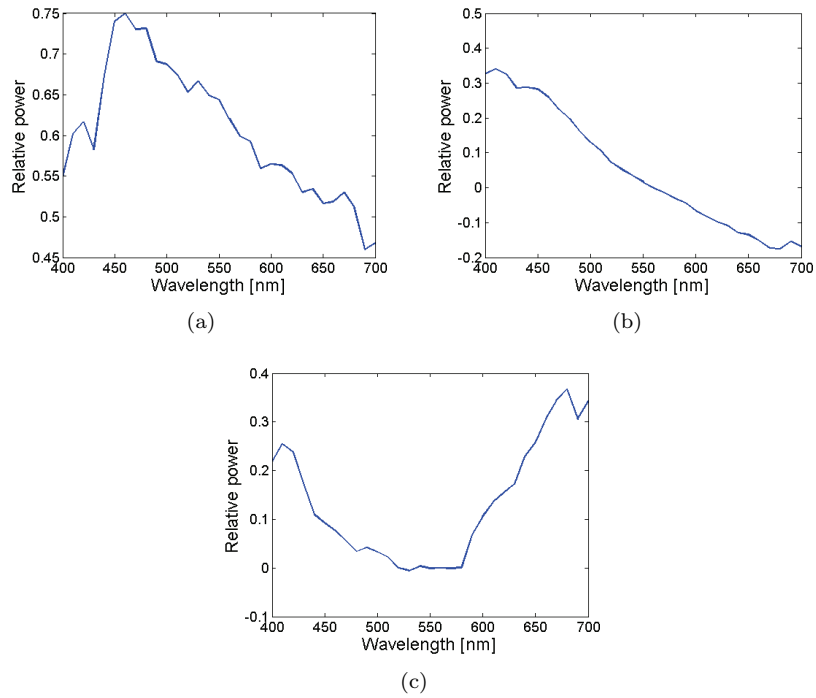


**Figure F.3:** Test artificial illuminants, subset of  $N_e = 20$  spectral power distributions used as reference illuminants. The x-axis represents the wavelength in nanometers. The y-axis represents the relative power.

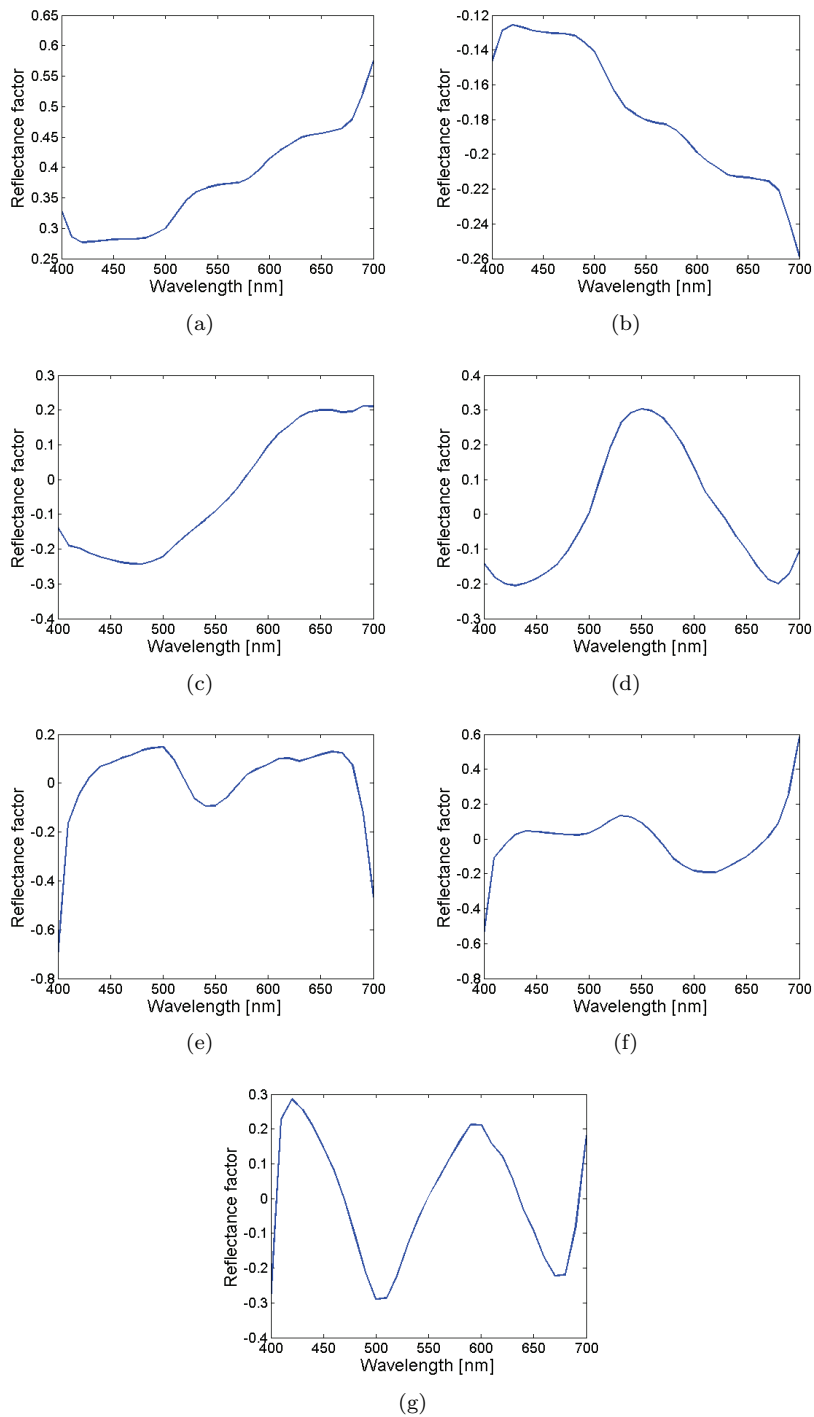


**Figure F.4:** Test artificial illuminants. The x-axis represents the wavelength in nanometers. The y-axis represents the relative power.

### F.3 Illuminant and reflectance basis functions



**Figure F.5:** Daylight illuminants basis functions computed by principal component analysis on the 45 real daylight illuminants represented in Figures F.1 and F.2.

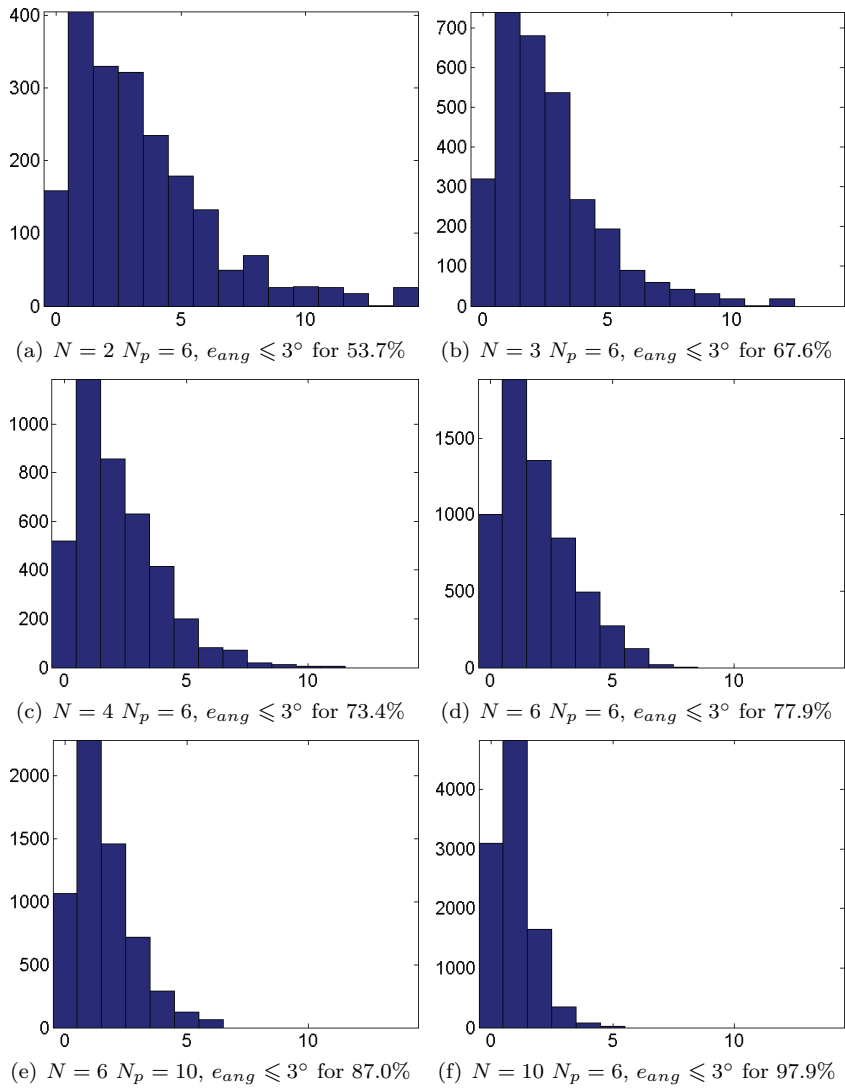


**Figure F.6:** Reflectance basis functions computed by principal component analysis on the MacBeth, Munsell, and Natural reflectance spectra.

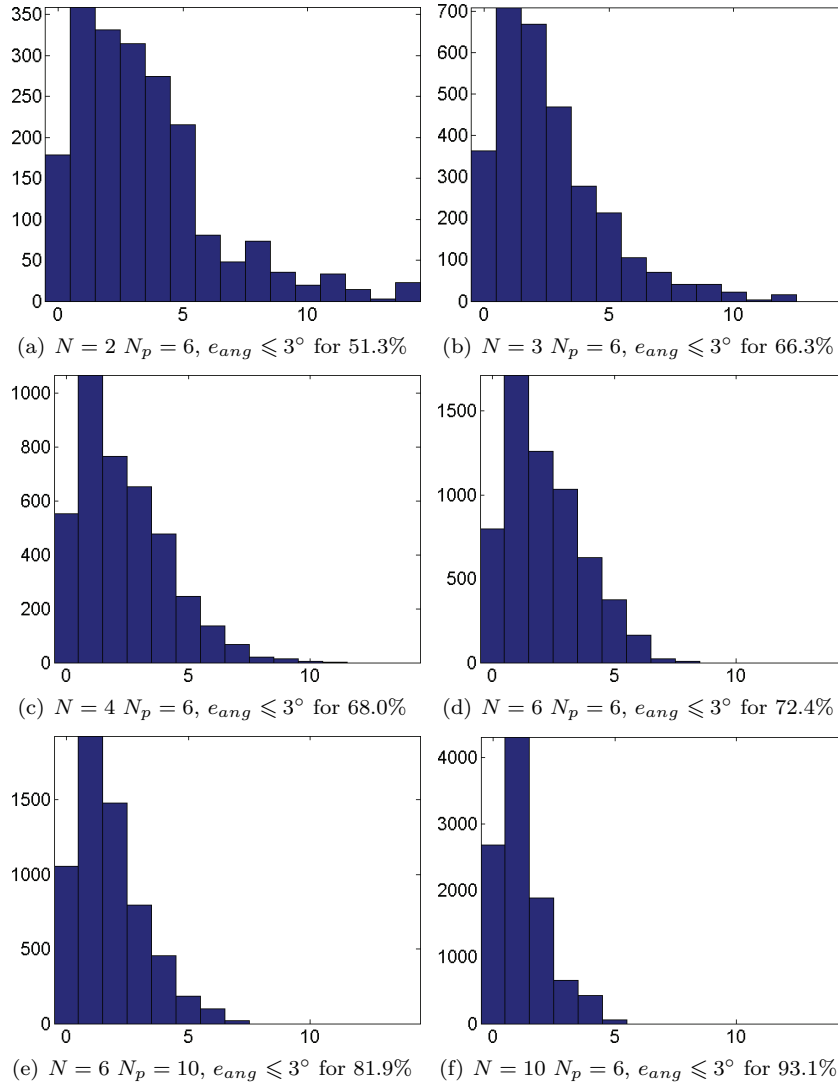
## F.4 Additional results

This section reports additional results for the simulations presented in Section 5.3.

### F.4.1 Angular error distributions for standard daylight illuminants

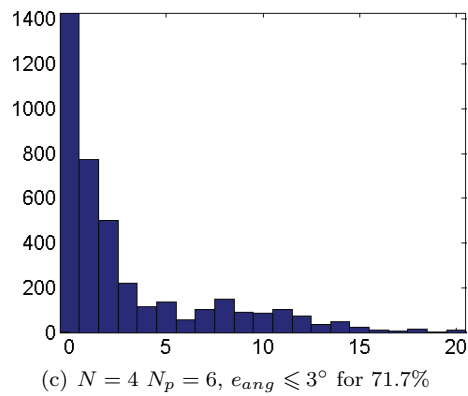
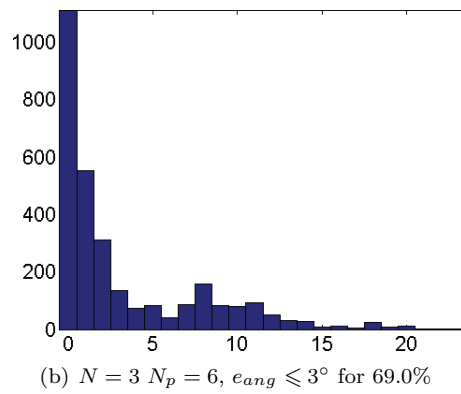
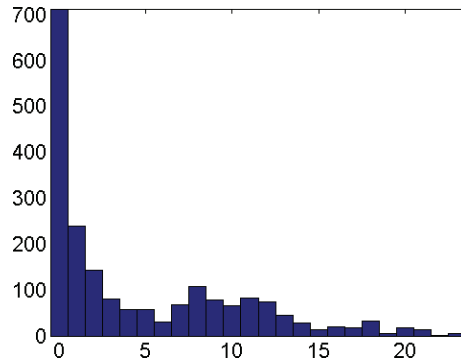


**Figure F.7:** Angular error distributions corresponding to Table 5.3. These results were obtained using Munsell reflectances and standard daylight illuminants. The error function is expressed as  $f(\theta_n)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.



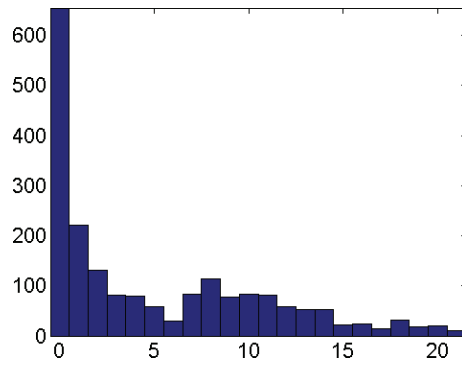
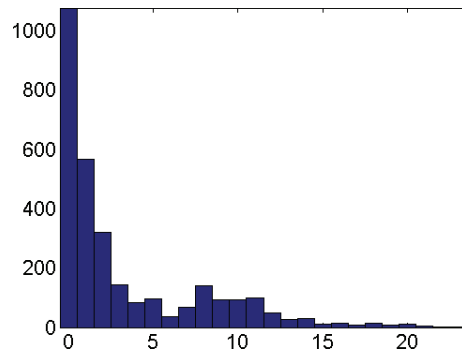
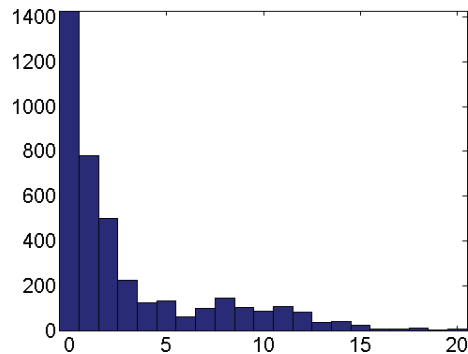
**Figure F.8:** Angular error distributions corresponding to Table 5.3. These results were obtained using Natural reflectances and standard daylight illuminants. The error function is expressed as  $f(\theta_n)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

### F.4.2 Angular error distributions for real daylight illuminants

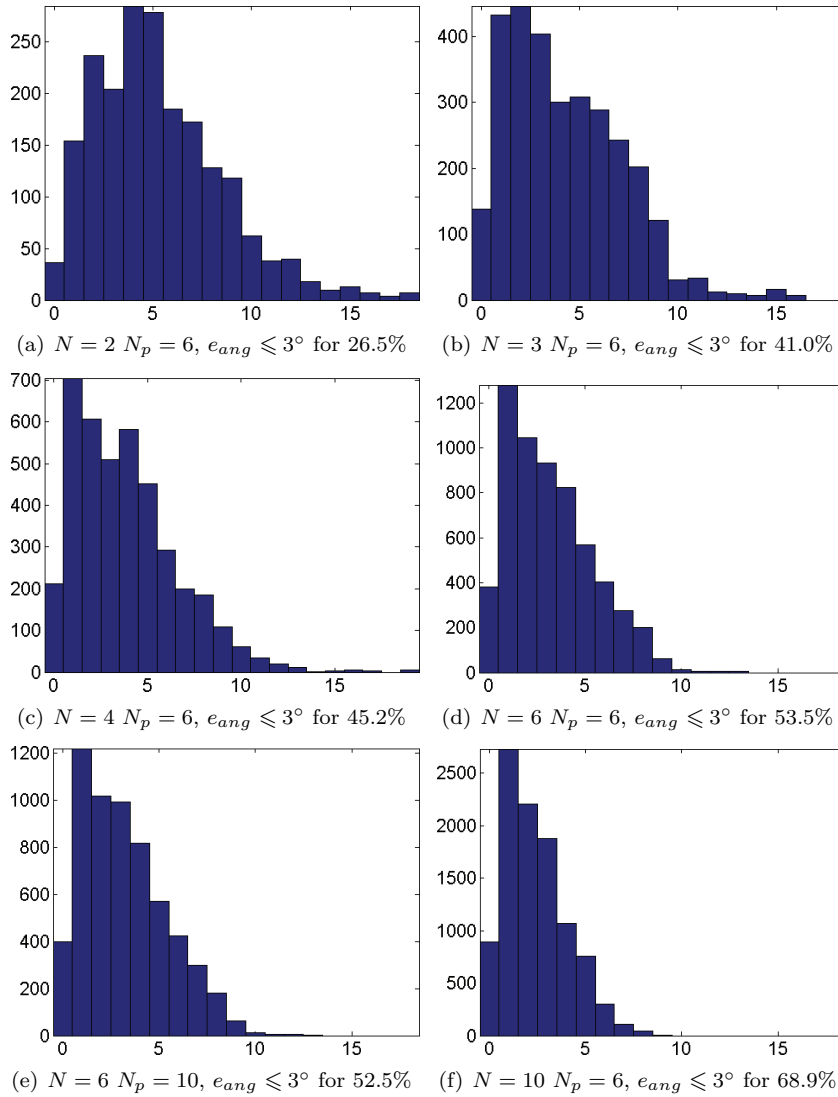


**Figure F.9:** Angular error distributions corresponding to Table 5.6. These results were obtained using Munsell reflectances and real daylight illuminants. The error function is minimized by testing a set of test illuminants. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

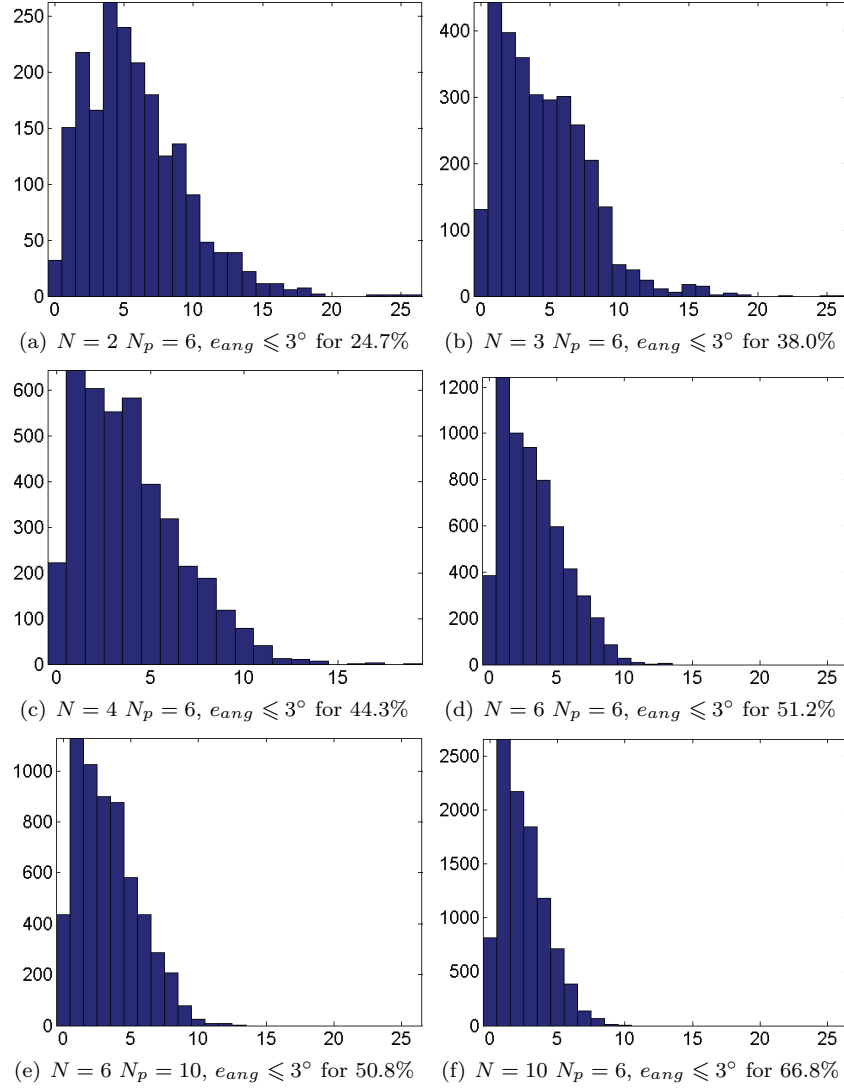


(a)  $N = 2$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 53.2%(b)  $N = 3$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 68.6%(c)  $N = 4$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 71.9%

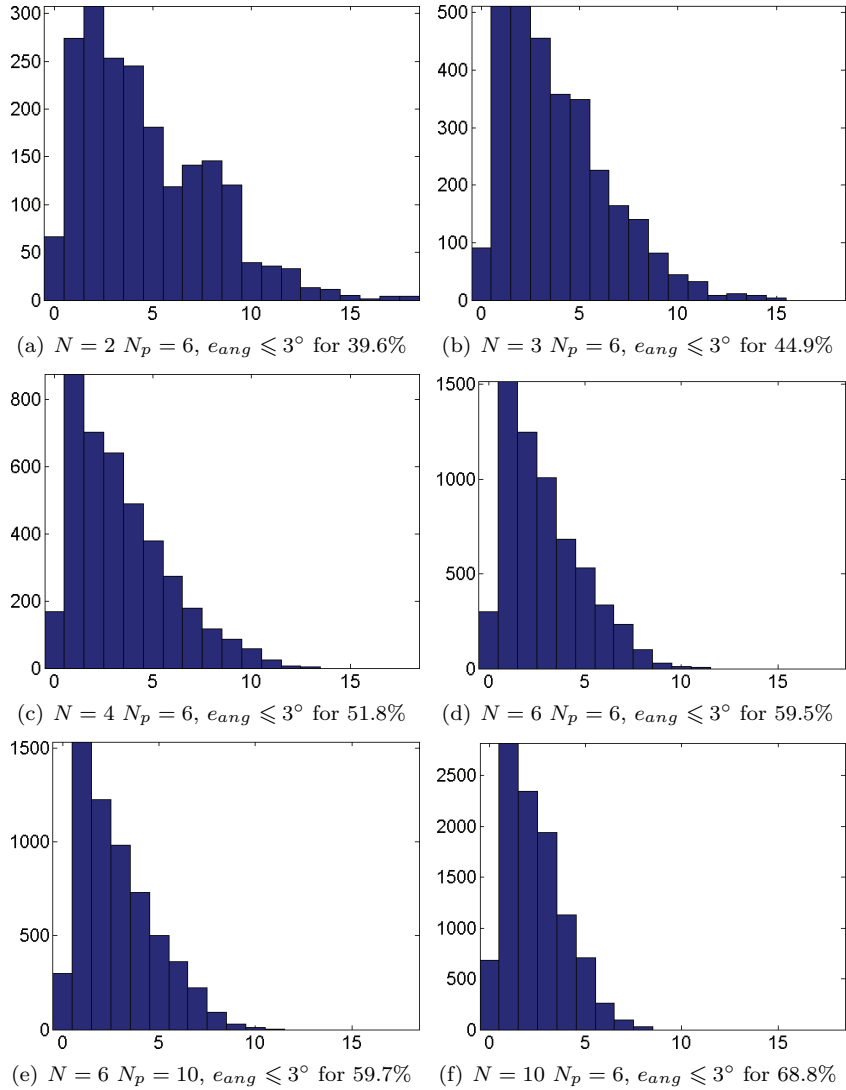
**Figure F.10:** Angular error distributions corresponding to Table 5.6. These results were obtained using Natural reflectances and real daylight illuminants. The error function is minimized by testing a set of test illuminants. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.



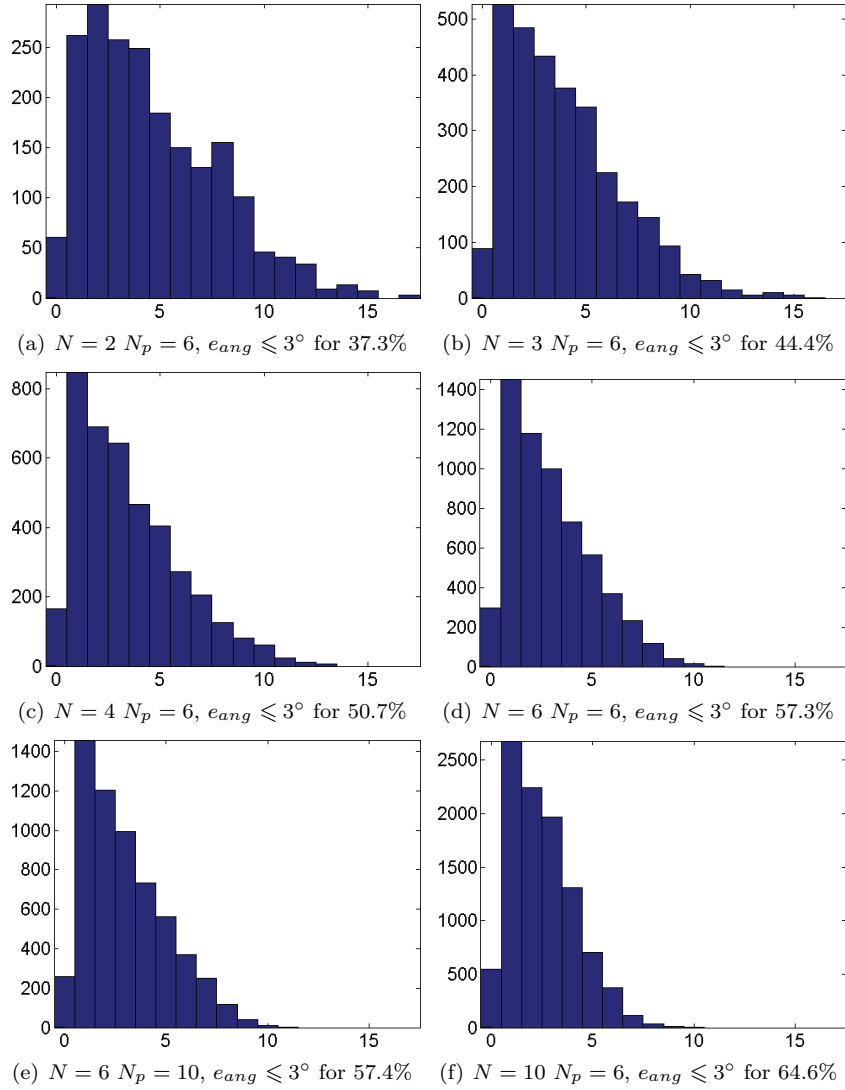
**Figure F.11:** Angular error distributions corresponding to Table 5.7. These results were obtained using Munsell reflectances and real daylight illuminants. The error function is expressed as  $f(\theta_n, \phi_n)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.



**Figure F.12:** Angular error distributions corresponding to Table 5.7. These results were obtained using Natural reflectances and real daylight illuminants. The error function is expressed as  $f(\theta_n, \phi_n)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

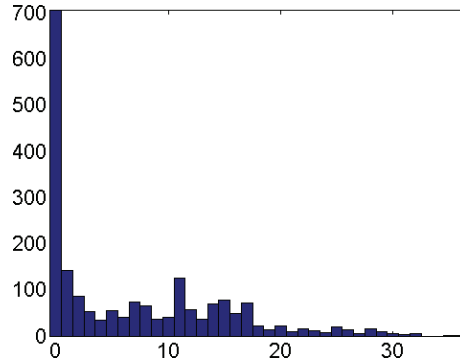


**Figure F.13:** Angular error distributions corresponding to Table 5.7. These results were obtained using Munsell reflectances and real daylight illuminants. The error function is expressed as  $f(\varepsilon_n)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

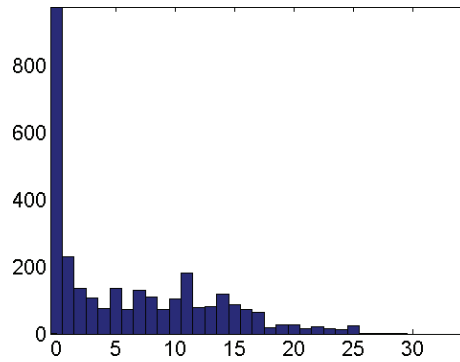


**Figure F.14:** Angular error distributions corresponding to Table 5.7. These results were obtained using Natural reflectances and real daylight illuminants. The error function is expressed as  $f(\varepsilon_n)$ . The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

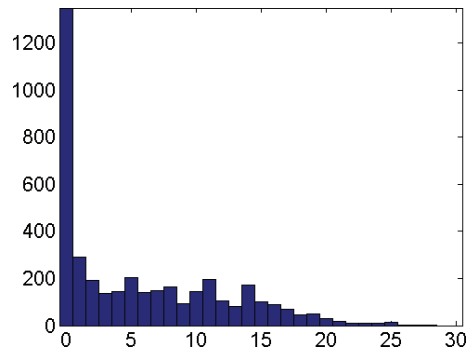
### F.4.3 Angular error distributions for artificial illuminants



(a)  $N = 2$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 47.6%

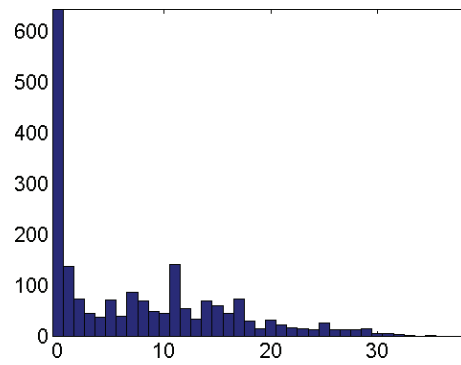
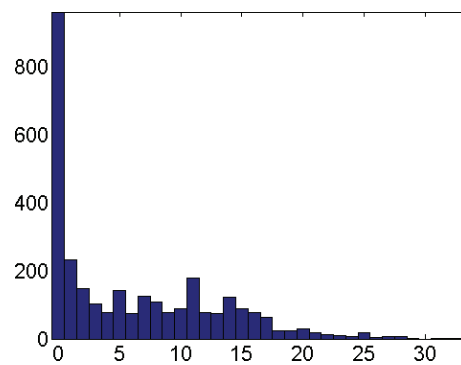
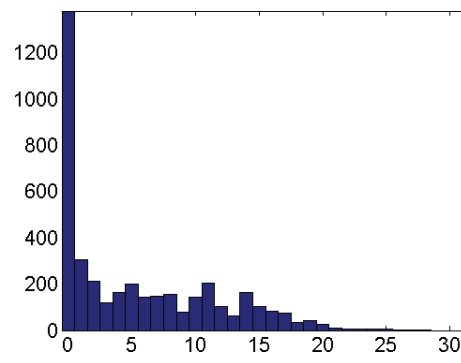


(b)  $N = 3$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 46.6%



(c)  $N = 4$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 47.7%

**Figure F.15:** Angular error distributions corresponding to Table 5.11. These results were obtained using Munsell reflectances and artificial illuminants. The error function is minimized by testing a set of test illuminants. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

(a)  $N = 2$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 43.7%(b)  $N = 3$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 46.6%(c)  $N = 4$   $N_p = 6$ ,  $e_{ang} \leq 3^\circ$  for 49.1%

**Figure F.16:** Angular error distributions corresponding to Table 5.11. These results were obtained using Natural reflectances and artificial illuminants. The error function is minimized by testing a set of test illuminants. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

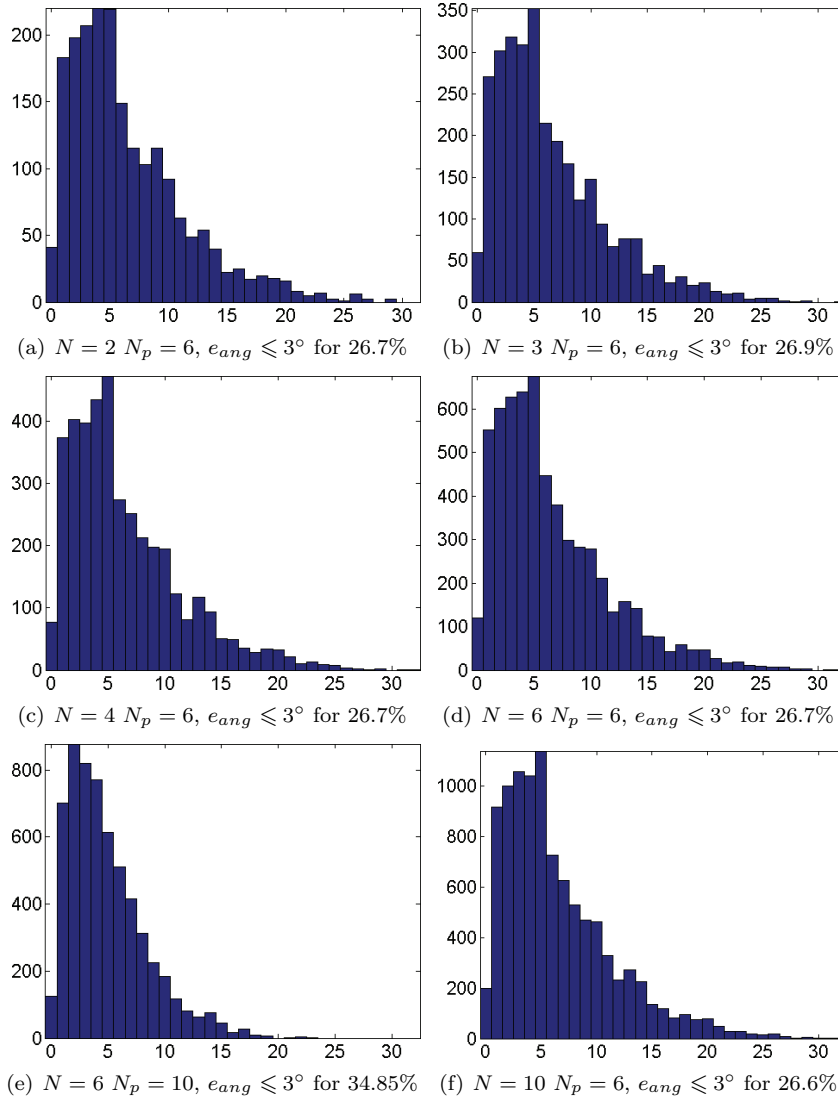
#### F.4.4 Comparison with D’Zmura and Iverson’s (1993a) algorithm

$A_{\text{noise}}$	D’Zmura and Iverson		$f_e(\varepsilon_1, \varepsilon_2, \varepsilon_3)$	
	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
0.0	24.06	153.14	3.42	14.20
0.1	25.70	151.72	3.42	14.48
0.2	25.81	148.83	3.45	14.41
0.3	24.81	147.76	3.39	14.98
0.4	25.66	148.27	3.38	14.06
0.5	23.72	148.41	3.42	14.30
0.6	24.19	154.36	3.41	13.89
0.7	25.12	146.97	3.33	15.34
0.8	23.86	148.27	3.44	14.57
0.9	25.42	149.71	3.48	15.30
1.0	27.46	152.43	3.46	15.47

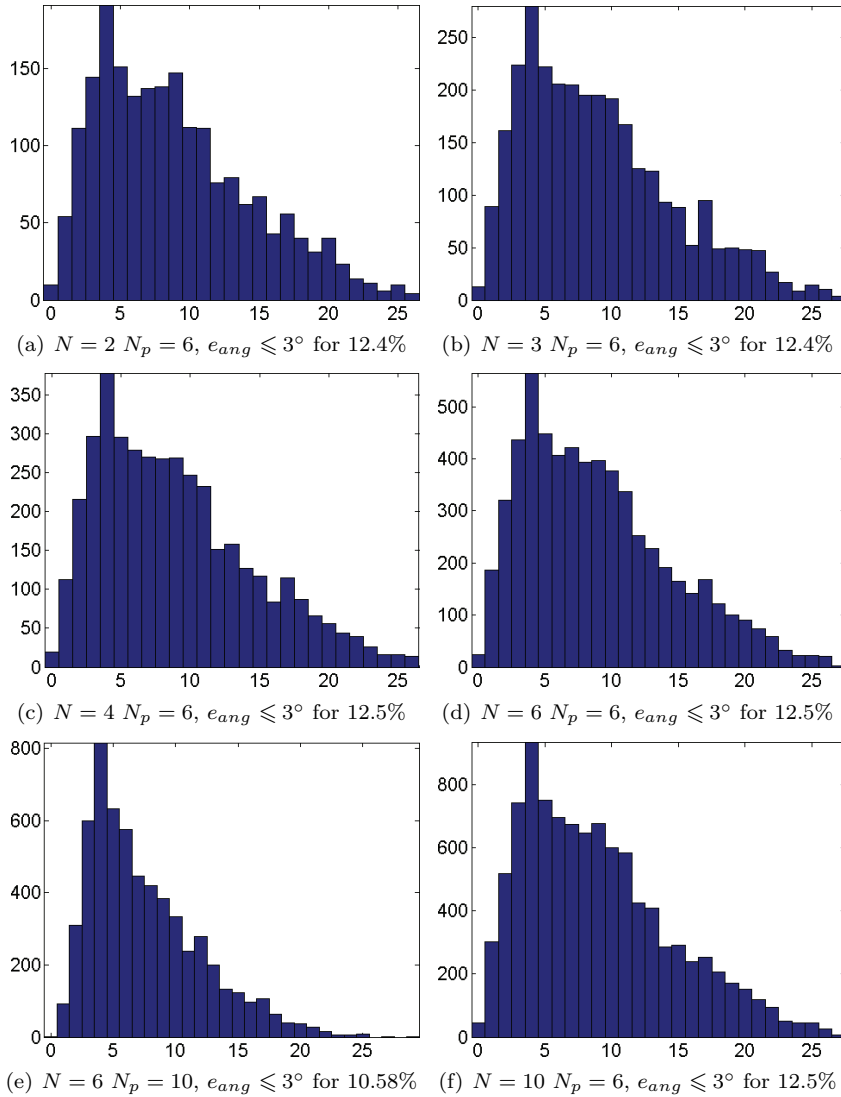
**Table F.5:** The algorithm was run  $10^4$  times using  $N_p = 3$  patches and  $N_E = 2$  standard daylight illuminants. This table reports the median and maximum angular errors  $e_{ang}$  as a function on the amount of shot noise added to the sensor responses for both D’Zmura and Iverson’s (1993a) and our algorithm.



**F.4.5 Angular error distributions for standard daylight illuminants obtained with the Shades of Gray algorithm**



**Figure F.17:** Angular error distributions corresponding to Table 5.10. The algorithm was run using standard daylight illuminants and Munsell reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.



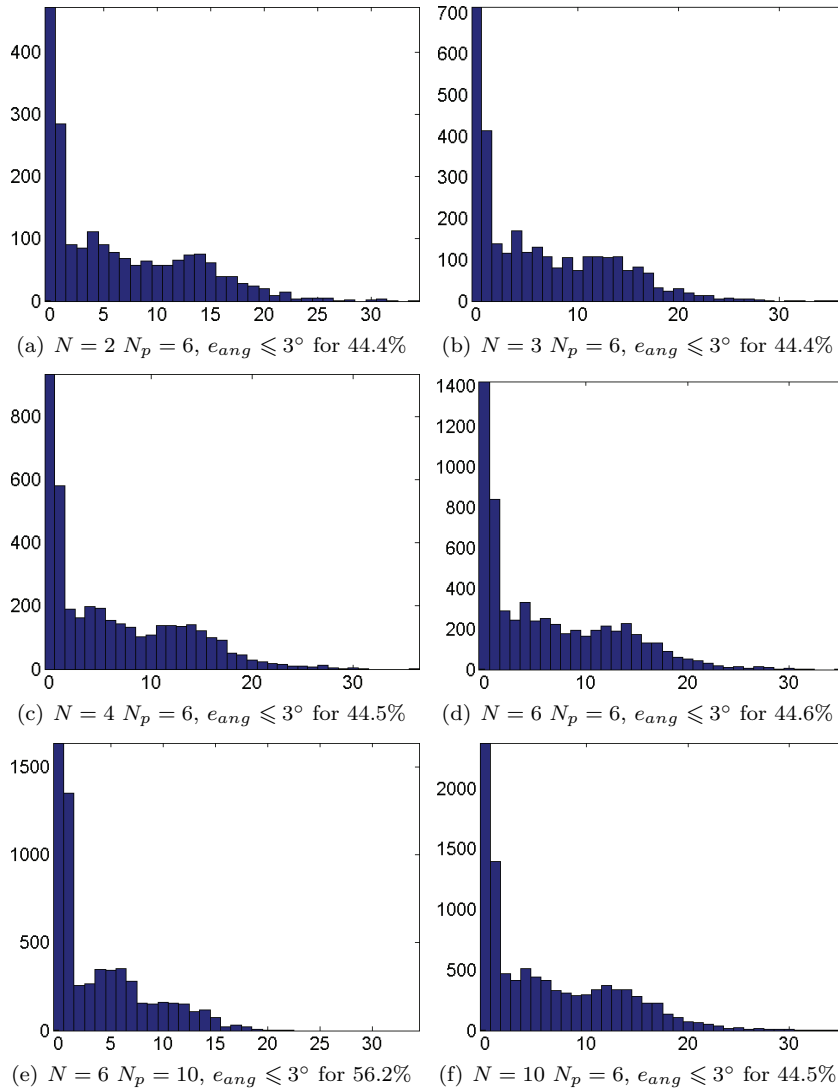
**Figure F.18:** Angular error distributions corresponding to Table 5.10. The algorithm was run using standard daylight illuminants and Natural reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

### F.4.6 Angular errors for real daylight illuminants obtained with the Shades of Gray algorithm

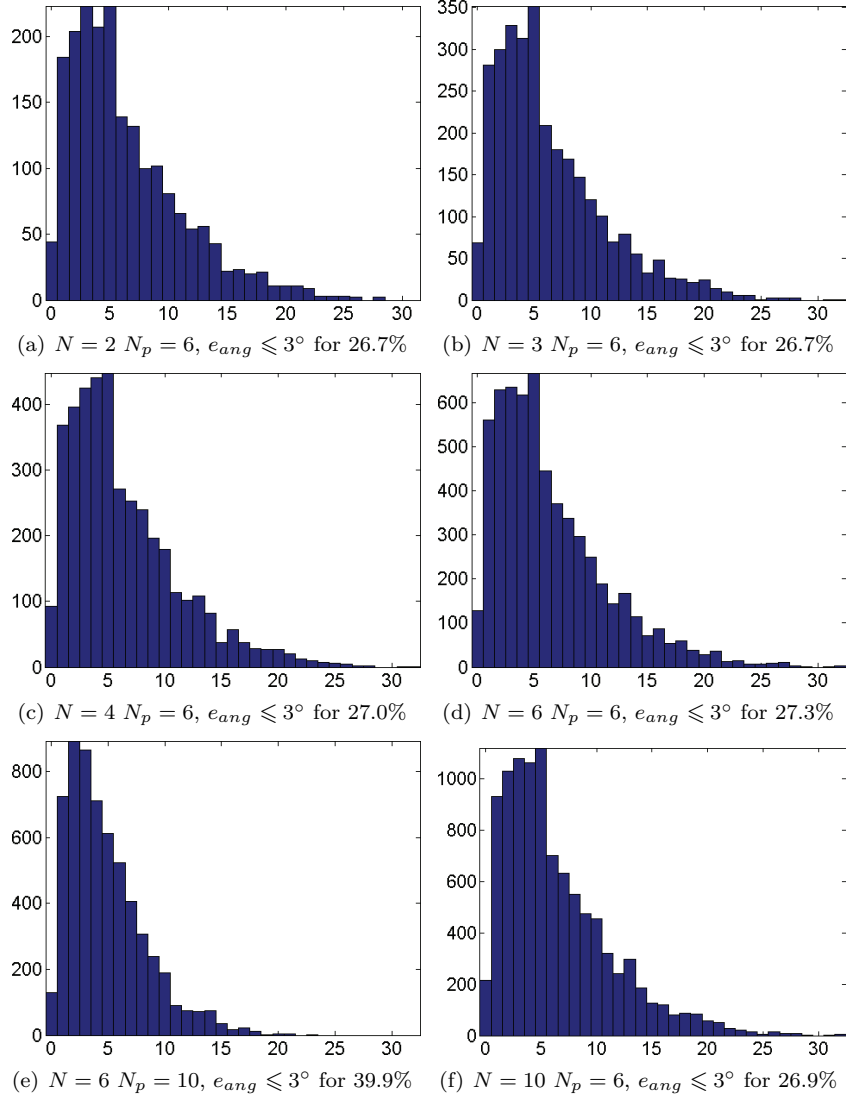
$N$	$N_p$	MacBeth		Munsell		Natural	
		med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
2	6	4.06	34.18	5.09	31.89	7.78	27.19
3	6	4.13	36.03	5.12	32.61	7.75	27.03
4	6	4.16	36.03	5.10	32.74	7.83	26.91
6	6	4.07	35.02	5.09	32.65	7.77	27.19
6	10	1.53	22.26	4.01	22.85	6.30	29.84
10	6	4.13	35.71	5.08	32.86	7.73	27.07

**Table F.6:** The SoG algorithm was run  $10^4$  times using different combinations of  $N_p$  patches and  $N_E$  measured daylight illuminants. This table reports the median and maximum angular errors  $e_{ang}$ .

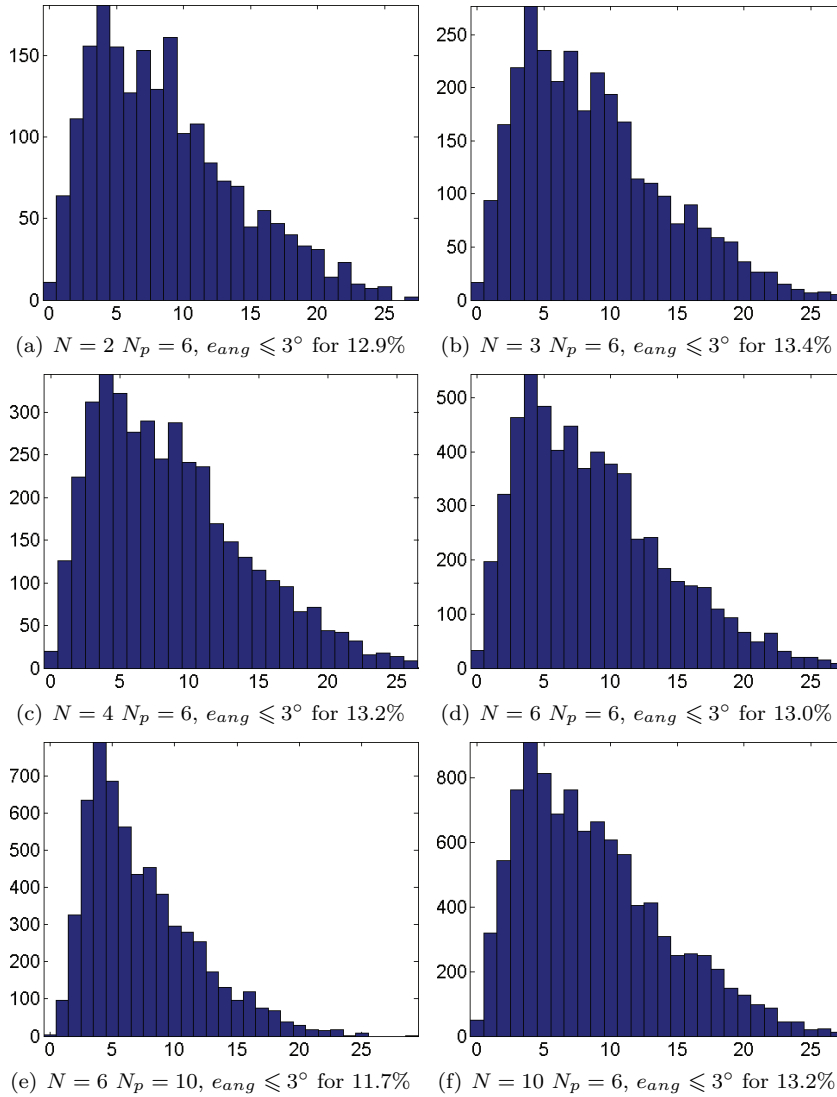
### F.4.7 Angular error distributions for real daylight illuminants obtained with the Shades of Gray algorithm



**Figure F.19:** Angular error distribution corresponding to Table F.6. The algorithm was run using measured daylight illuminants and MacBeth reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count. The error distributions for the Munsell and Natural reflectance are in Section 5.3.5



**Figure F.20:** Angular error distribution corresponding to Table F.6. The algorithm was run using measured daylight illuminants and Munsell reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.



**Figure F.21:** Angular error distribution corresponding to Table F.6. The algorithm was run using measured daylight illuminants and Natural reflectances. The x-axis represents the angular error  $e_{ang}$  in sRGB expressed in degrees. The y-axis represents the image count.

### F.4.8 Angular errors for Shades of Gray in presence of shot noise

$A_{\text{noise}}$	$N$	$N_p$	MacBeth		Munsell		Natural	
			med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
0	2	6	4.15	33.41	5.19	31.91	8.06	26.10
	3	6	4.26	34.17	5.19	32.54	8.02	27.07
	4	6	4.13	34.88	5.17	32.63	8.03	26.57
	6	6	4.16	33.72	5.17	32.63	7.98	27.07
	6	10	1.51	22.24	4.09	22.74	6.43	29.36
	10	6	4.15	34.88	5.17	32.72	8.04	27.07
0.1	2	6	4.15	33.54	5.20	31.98	8.05	26.06
	3	6	4.24	33.99	5.20	33.17	8.00	27.27
	4	6	4.13	34.65	5.21	32.52	8.02	26.85
	6	6	4.21	33.91	5.20	33.08	7.99	27.43
	6	10	1.63	22.45	4.09	22.84	6.44	29.58
	10	6	4.18	34.72	5.20	33.01	8.05	27.05
0.2	2	6	4.23	33.31	5.27	31.31	8.01	26.06
	3	6	4.25	34.20	5.17	32.13	7.94	27.30
	4	6	4.17	35.48	5.23	33.69	8.05	26.67
	6	6	4.31	33.68	5.25	32.24	7.98	27.68
	6	10	1.77	22.82	4.11	22.76	6.48	28.74
	10	6	4.23	35.36	5.22	33.55	8.04	27.31
0.3	2	6	4.19	33.84	5.26	31.52	7.97	26.45
	3	6	4.39	34.41	5.24	33.56	8.02	28.17
	4	6	4.25	35.74	5.29	32.34	7.99	27.17
	6	6	4.24	34.13	5.27	33.08	8.03	27.88
	6	10	1.98	23.29	4.15	23.31	6.51	28.81
	10	6	4.30	34.55	5.27	33.02	8.03	27.83
0.4	2	6	4.24	33.51	5.28	32.60	8.05	26.53
	3	6	4.30	35.33	5.34	32.94	8.12	27.84
	4	6	4.36	33.66	5.34	33.81	8.04	26.91
	6	6	4.35	34.24	5.31	33.71	7.98	26.99
	6	10	2.19	22.50	4.23	23.47	6.51	29.19
	10	6	4.26	33.77	5.38	33.56	8.05	27.67
0.5	2	6	4.43	32.13	5.41	32.95	8.11	27.98
	3	6	4.48	35.70	5.41	33.43	8.17	28.53
	4	6	4.31	34.98	5.43	32.67	8.00	27.27
	6	6	4.41	33.43	5.41	35.27	8.07	27.91
	6	10	2.56	24.18	4.29	23.55	6.62	29.83
	10	6	4.42	35.39	5.42	34.22	8.05	29.20

**Table F.7:** Median and maximum angular errors  $e_{ang}$  obtained for the shades of gray (SoG) algorithm with shot noise. The simulations were run on standard daylight illuminants and for  $A_{\text{noise}} = 0 - 0.5$ .

$A_{\text{noise}}$			MacBeth		Munsell		Natural	
			med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
0.6	2	6	4.45	34.86	5.52	30.27	7.93	28.00
	3	6	4.50	33.22	5.55	33.12	8.21	28.52
	4	6	4.48	36.32	5.52	36.82	8.11	27.79
	6	6	4.54	34.61	5.38	32.26	8.09	29.24
	6	10	2.74	23.98	4.33	22.99	6.61	28.74
	10	6	4.49	35.09	5.51	35.42	8.13	28.64
0.7	2	6	4.64	35.06	5.58	36.12	8.15	27.38
	3	6	4.60	34.87	5.64	30.65	8.05	30.31
	4	6	4.59	36.06	5.67	36.08	8.14	28.15
	6	6	4.60	34.55	5.53	33.51	8.10	27.78
	6	10	3.06	24.42	4.43	24.64	6.70	29.44
	10	6	4.66	35.06	5.59	35.03	8.09	29.04
0.8	2	6	4.69	33.58	5.77	30.41	8.14	28.51
	3	6	5.00	34.84	5.71	31.24	8.18	29.38
	4	6	4.80	35.79	5.76	34.69	8.20	30.98
	6	6	4.63	33.35	5.74	38.01	8.14	28.24
	6	10	3.32	25.41	4.51	24.85	6.82	29.15
	10	6	4.83	34.98	5.73	34.80	8.15	28.65
0.9	2	6	4.97	34.78	5.82	29.84	8.27	30.16
	3	6	5.06	35.80	5.86	32.24	8.13	29.35
	4	6	4.93	35.09	5.89	34.54	8.18	28.99
	6	6	4.91	34.68	5.73	32.65	8.19	31.50
	6	10	3.67	24.25	4.64	24.42	6.91	28.31
	10	6	5.02	37.20	5.84	36.14	8.22	29.81
1	2	6	5.25	35.46	5.91	31.59	8.26	28.48
	3	6	5.14	37.14	5.90	35.88	8.40	29.54
	4	6	5.24	34.42	5.89	31.41	8.36	29.96
	6	6	5.17	35.62	5.91	34.03	8.27	31.24
	6	10	3.96	26.97	4.76	23.15	7.00	29.05
	10	6	5.10	38.36	5.94	33.86	8.28	30.43

**Table F.8:** Median and maximum angular errors  $e_{ang}$  obtained for the shades of gray (SoG) algorithm with shot noise. The simulations were run on standard daylight illuminants and for  $A_{\text{noise}} = 0.6 - 1.0$ .



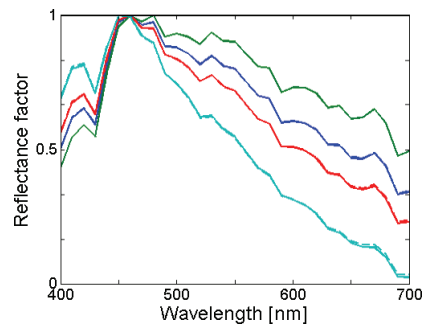
$A_{\text{noise}}$	$N$	$N_p$	MacBeth		Munsell		Natural	
			med $e_{ang}$	max $E_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
0	2	6	4.06	34.18	5.09	31.89	7.78	27.19
	3	6	4.13	36.03	5.12	32.61	7.75	27.03
	4	6	4.16	36.03	5.10	32.74	7.83	26.91
	6	6	4.07	35.02	5.09	32.65	7.77	27.19
	6	10	1.53	22.26	4.01	22.85	6.30	29.84
	10	6	4.13	35.71	5.08	32.86	7.73	27.07
0.1	2	6	4.06	34.51	5.08	31.96	7.77	26.90
	3	6	4.14	35.55	5.11	31.98	7.76	27.26
	4	6	4.21	36.06	5.11	32.87	7.83	27.13
	6	6	4.09	34.59	5.09	33.16	7.75	27.58
	6	10	1.63	22.57	4.03	22.72	6.31	29.54
	10	6	4.15	36.05	5.10	33.82	7.73	27.52
0.2	2	6	4.14	34.47	5.17	30.26	7.85	27.32
	3	6	4.29	35.78	5.14	32.95	7.77	27.61
	4	6	4.20	37.05	5.17	33.27	7.87	27.24
	6	6	4.14	34.19	5.15	32.98	7.78	28.12
	6	10	1.79	22.37	4.06	23.07	6.34	29.72
	10	6	4.16	36.26	5.12	33.30	7.76	27.87
0.3	2	6	4.21	34.13	5.25	29.45	7.88	27.35
	3	6	4.15	35.10	5.25	32.66	7.81	27.11
	4	6	4.22	34.90	5.18	32.29	7.89	27.62
	6	6	4.25	34.11	5.21	34.69	7.81	27.92
	6	10	1.99	22.89	4.10	22.71	6.38	30.01
	10	6	4.19	35.46	5.21	33.95	7.79	27.77
0.4	2	6	4.36	34.70	5.21	28.90	7.86	28.79
	3	6	4.15	36.36	5.30	32.71	7.87	28.45
	4	6	4.28	35.88	5.28	33.65	7.90	27.72
	6	6	4.21	35.19	5.24	35.09	7.86	28.19
	6	10	2.24	22.41	4.15	23.21	6.41	30.23
	10	6	4.27	35.90	5.25	33.44	7.84	28.63
0.5	2	6	4.52	33.72	5.48	32.00	7.85	26.89
	3	6	4.38	37.71	5.37	33.65	7.92	28.85
	4	6	4.29	36.96	5.40	32.17	7.88	28.02
	6	6	4.29	34.52	5.38	33.72	7.92	27.73
	6	10	2.52	22.63	4.14	24.59	6.48	31.10
	10	6	4.39	36.11	5.35	34.58	7.84	28.39

**Table F.9:** Median and maximum angular errors  $e_{ang}$  obtained for the shades of gray (SoG) algorithm with shot noise. The simulations were run on measured daylight illuminants and for  $A_{\text{noise}} = 0 - 0.5$ .

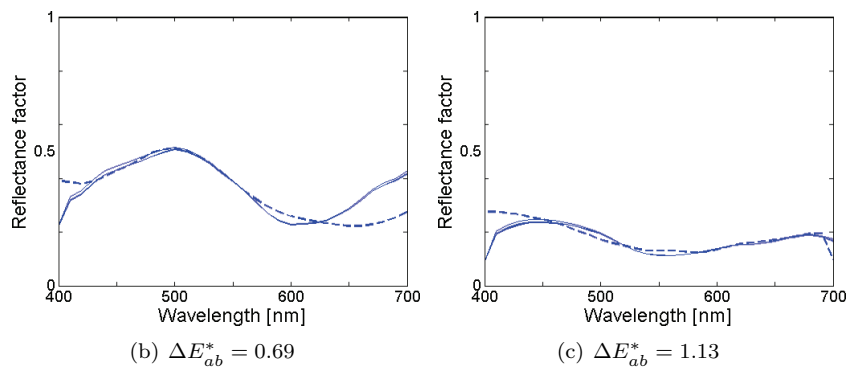
$A_{\text{noise}}$			MacBeth		Munsell		Natural	
			med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$	med $e_{ang}$	max $e_{ang}$
0.6	2	6	4.49	35.42	5.37	31.80	8.04	26.72
	3	6	4.45	35.84	5.38	34.53	8.06	28.51
	4	6	4.42	35.89	5.44	31.50	7.93	27.43
	6	6	4.45	36.57	5.48	35.11	7.88	29.62
	6	10	2.74	23.69	4.26	23.92	6.49	29.59
	10	6	4.48	37.55	5.41	35.76	7.81	28.79
0.7	2	6	4.76	36.35	5.61	31.32	8.00	29.70
	3	6	4.56	38.67	5.50	33.61	7.95	29.89
	4	6	4.49	33.77	5.48	33.50	7.94	29.06
	6	6	4.55	35.65	5.47	34.92	8.00	29.36
	6	10	3.04	23.44	4.31	23.74	6.53	30.19
	10	6	4.64	36.76	5.56	34.05	7.97	30.07
0.8	2	6	4.72	34.76	5.67	34.63	7.98	28.29
	3	6	4.73	36.29	5.68	35.95	8.01	30.26
	4	6	4.79	33.18	5.62	34.77	7.96	28.13
	6	6	4.71	35.25	5.66	33.53	8.07	29.49
	6	10	3.32	24.26	4.45	22.19	6.68	32.48
	10	6	4.76	36.48	5.65	37.48	7.97	28.71
0.9	2	6	5.14	36.14	5.81	32.44	8.03	30.64
	3	6	4.93	38.92	5.77	33.47	8.23	29.39
	4	6	4.88	36.36	5.70	37.44	7.99	30.13
	6	6	5.02	38.57	5.82	37.27	8.05	30.81
	6	10	3.61	24.86	4.53	21.73	6.76	27.31
	10	6	4.96	37.65	5.69	35.38	8.14	30.31
1	2	6	5.17	35.91	5.81	32.84	8.11	28.14
	3	6	5.21	37.53	5.84	35.97	8.13	30.75
	4	6	5.22	33.18	5.78	31.82	8.16	30.16
	6	6	5.13	37.01	5.96	38.81	8.03	30.59
	6	10	3.90	24.05	4.65	21.99	6.75	31.56
	10	6	5.16	38.09	5.85	37.01	8.05	29.64

**Table F.10:** Median and maximum angular errors  $e_{ang}$  obtained for the shades of gray (SoG) algorithm with shot noise. The simulations were run on measured daylight illuminants and for  $A_{\text{noise}} = 0.6 - 1.0$ .

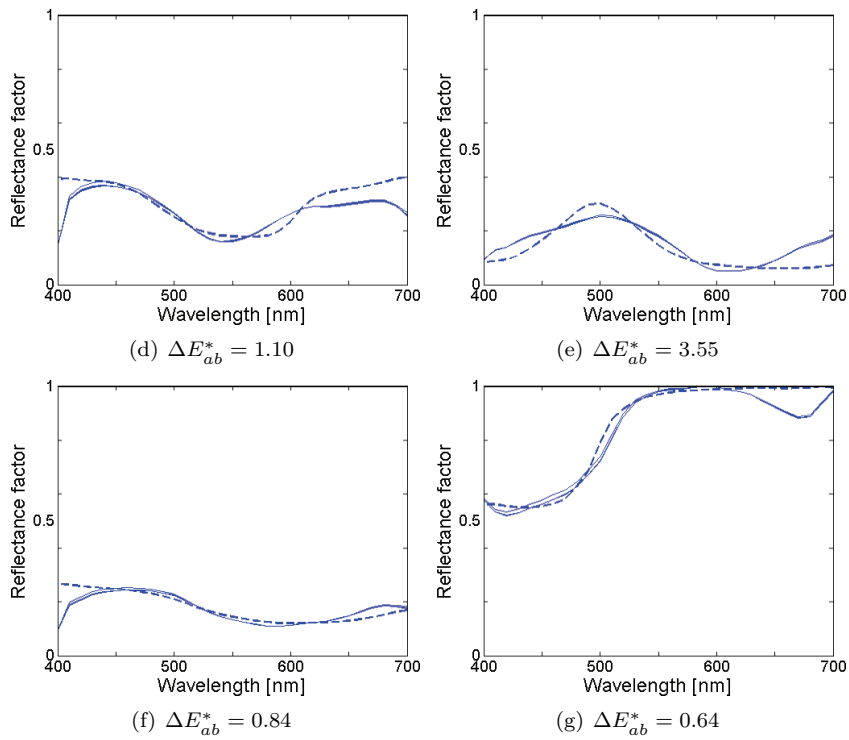
## F.4.9 Reflectances estimates



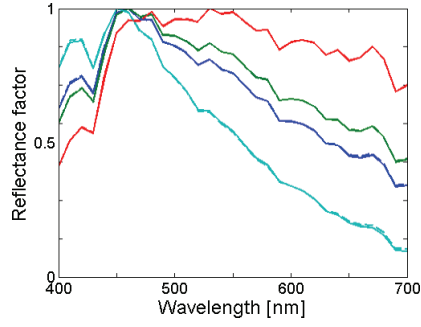
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



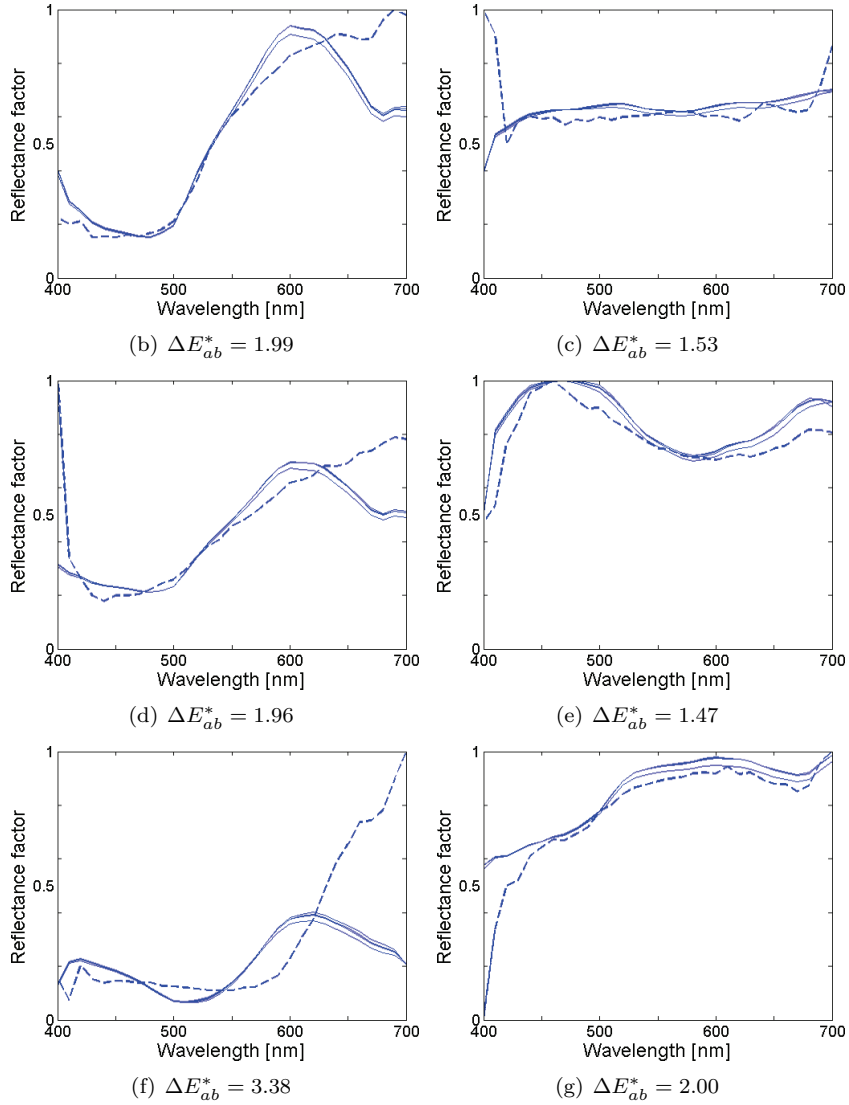
**Figure F.22:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Munsell reflectances and standard illuminants. They correspond to the lowest angular error  $e_{ang}$ .



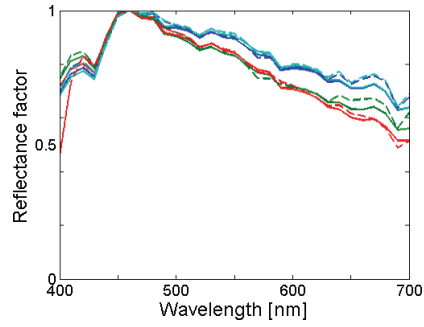
**Figure F.22:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Munsell reflectances and standard illuminants. They correspond to the lowest angular error  $e_{ang}$ .



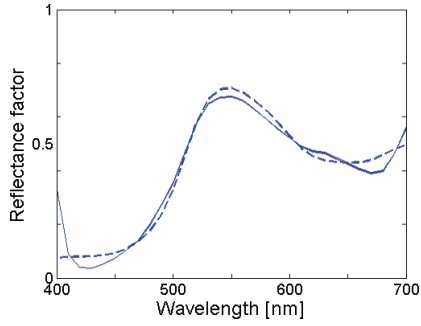
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



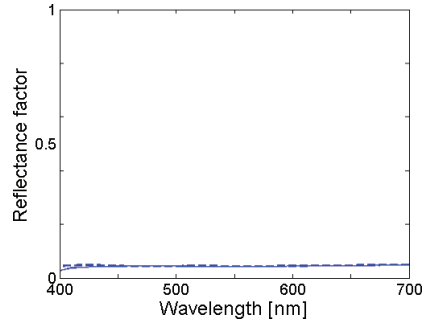
**Figure F.23:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Natural reflectances and standard illuminants. They correspond to the lowest angular error  $e_{ang}$ .



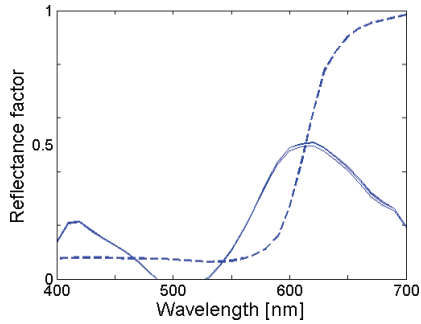
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



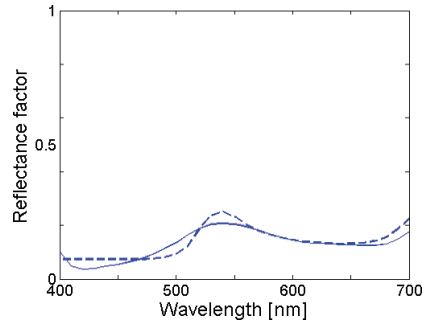
(b)  $\Delta E_{ab}^* = 1.55$



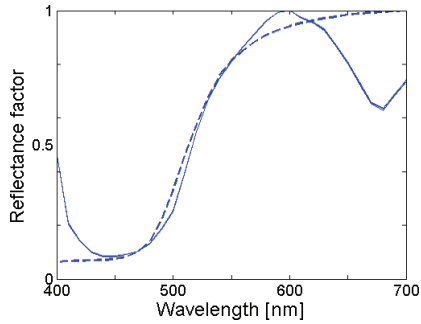
(c)  $\Delta E_{ab}^* = 0.31$



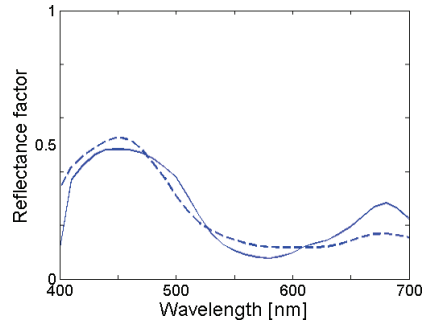
(d)  $\Delta E_{ab}^* = 5.89$



(e)  $\Delta E_{ab}^* = 3.10$

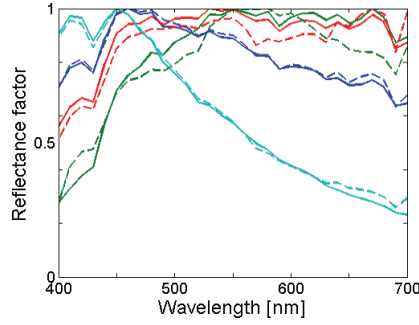


(f)  $\Delta E_{ab}^* = 3.55$

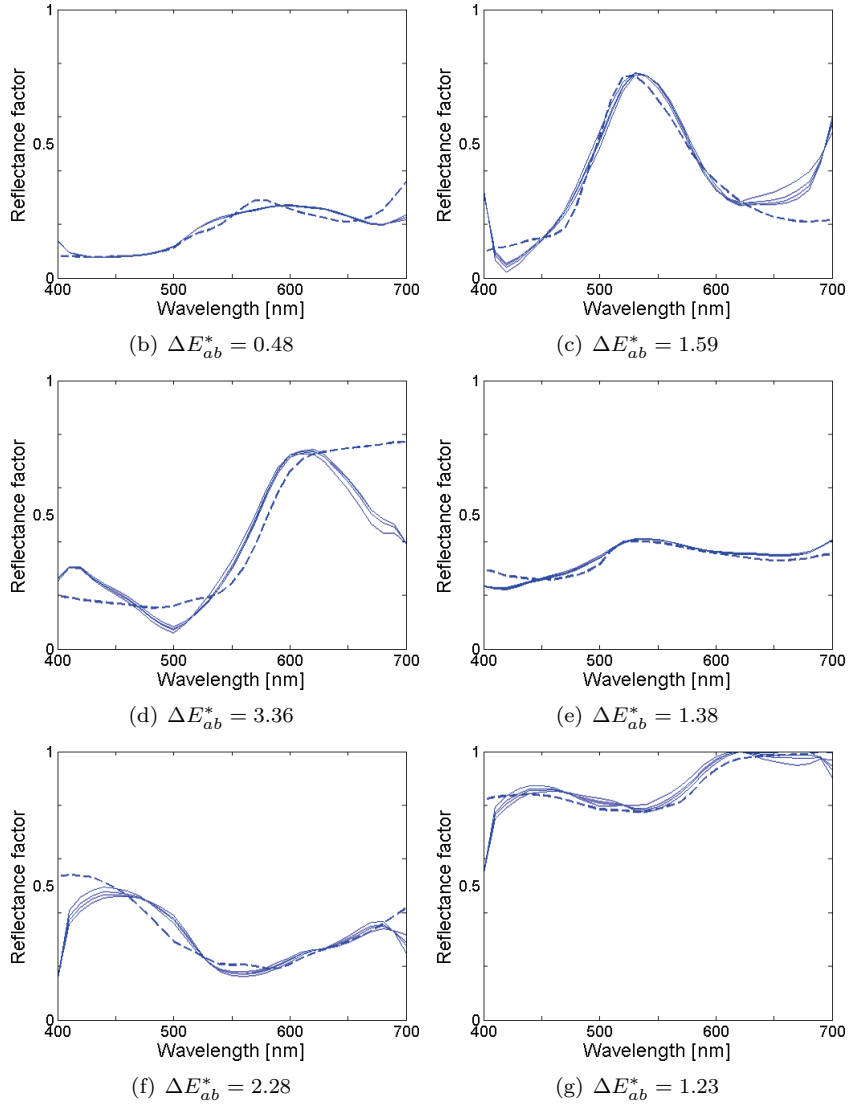


(g)  $\Delta E_{ab}^* = 1.81$

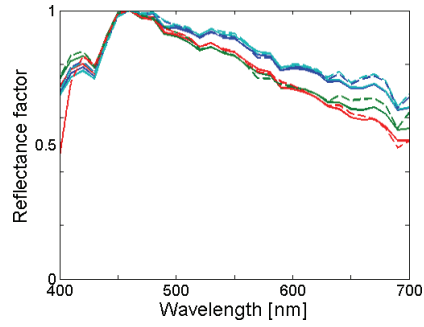
**Figure F.24:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using MacBeth reflectances and real daylight illuminants. The error function is expressed as  $f_e(\theta_n, \phi_n)$ . They correspond to the lowest angular error  $e_{ang}$ .



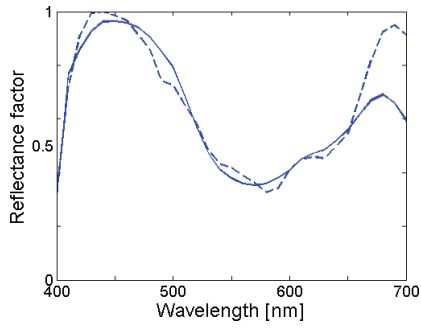
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



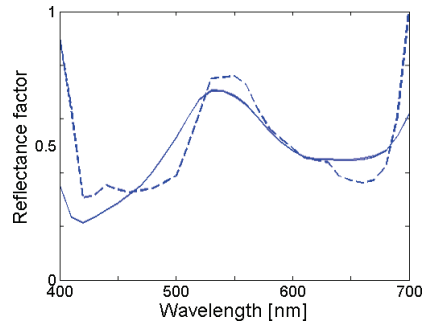
**Figure F.25:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Munsell reflectances and real daylight illuminants. The error function is expressed as  $f_e(\theta_n, \phi_n)$ . They correspond to the lowest angular error  $e_{ang}$ .



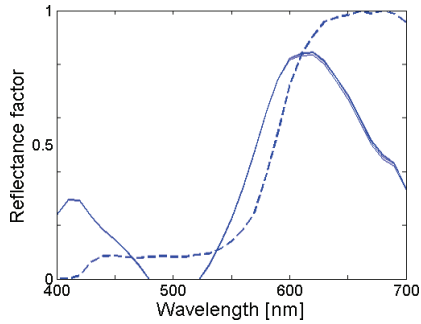
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



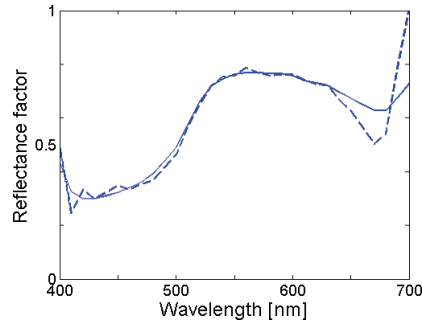
(b)  $\Delta E_{ab}^* = 0.78$



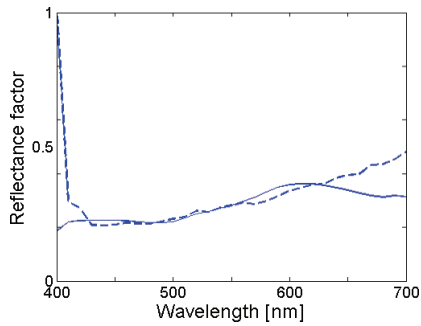
(c)  $\Delta E_{ab}^* = 4.31$



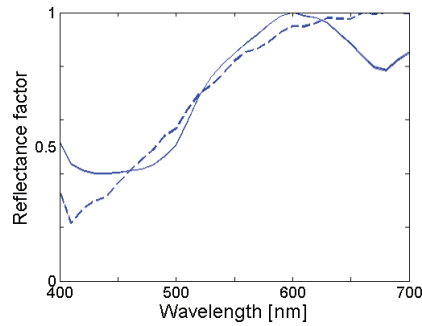
(d)  $\Delta E_{ab}^* = 9.36$



(e)  $\Delta E_{ab}^* = 0.41$



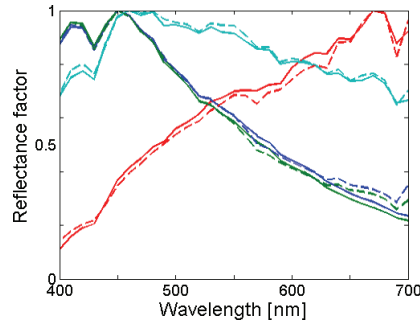
(f)  $\Delta E_{ab}^* = 0.85$



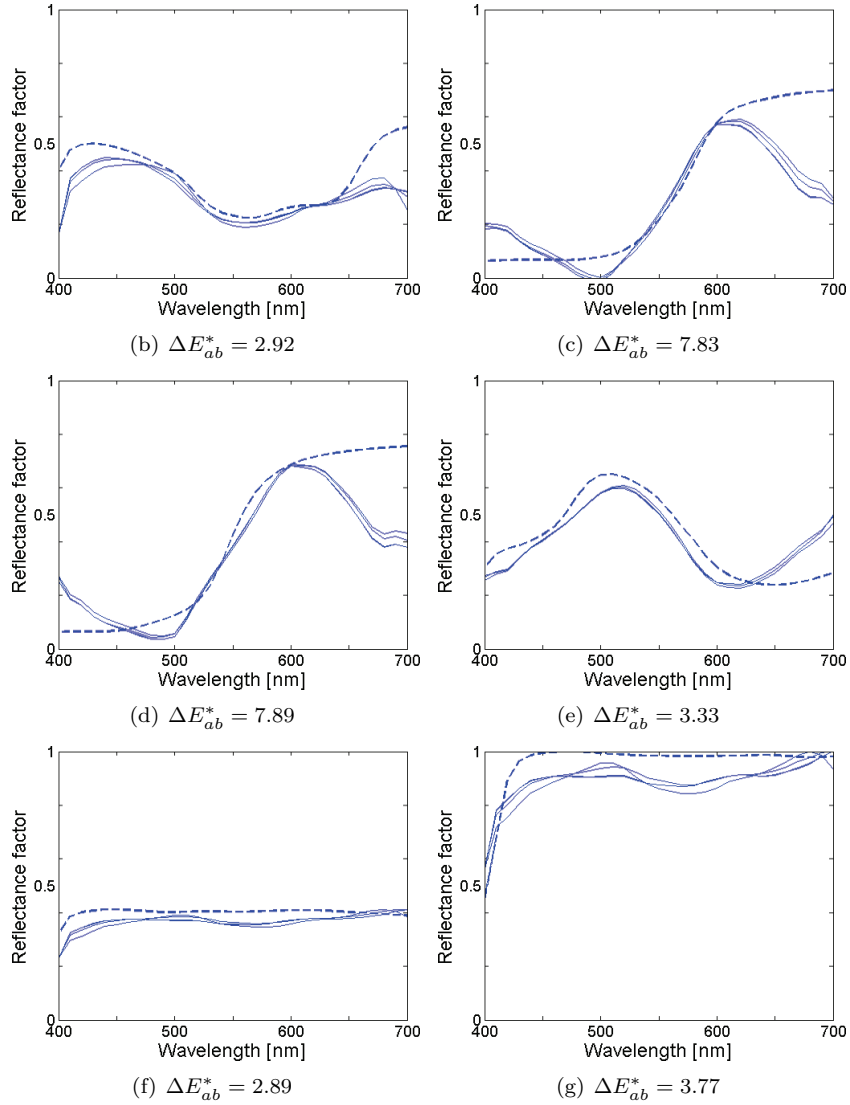
(g)  $\Delta E_{ab}^* = 2.68$

**Figure F.26:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Natural reflectances and real daylight illuminants. The error function is expressed as  $f_e(\theta_n, \phi_n)$ . They correspond to the lowest angular error  $e_{ang}$ .

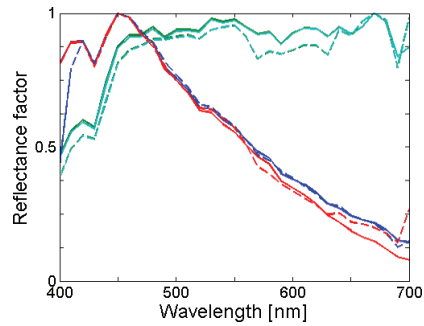




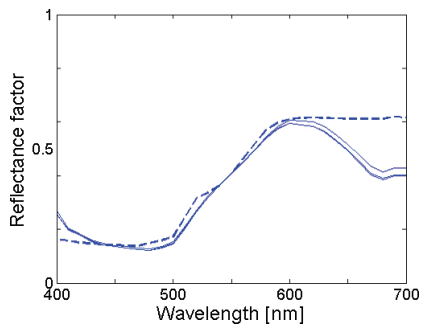
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



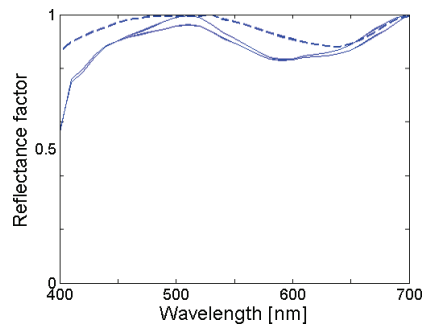
**Figure F.27:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using MacBeth reflectances and real daylight illuminants. The error function is expressed as  $f_e(\epsilon_n)$ . They correspond to the lowest angular error  $e_{ang}$ .



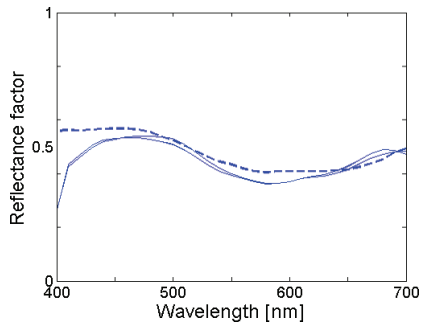
(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



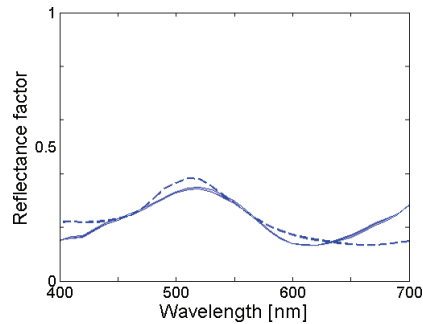
(b)  $\Delta E_{ab}^* = 2.21$



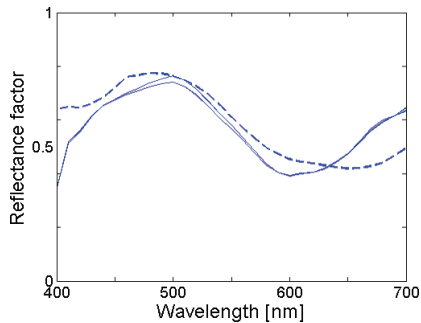
(c)  $\Delta E_{ab}^* = 2.74$



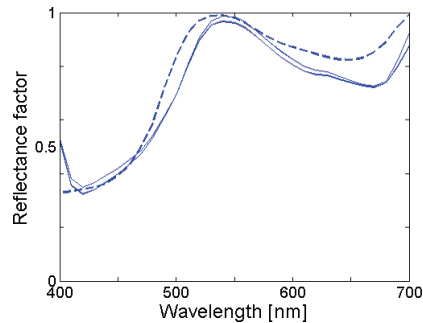
(d)  $\Delta E_{ab}^* = 2.45$



(e)  $\Delta E_{ab}^* = 1.29$

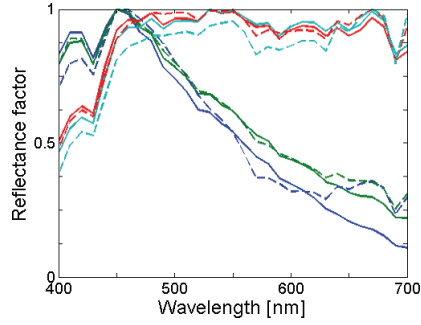


(f)  $\Delta E_{ab}^* = 2.38$

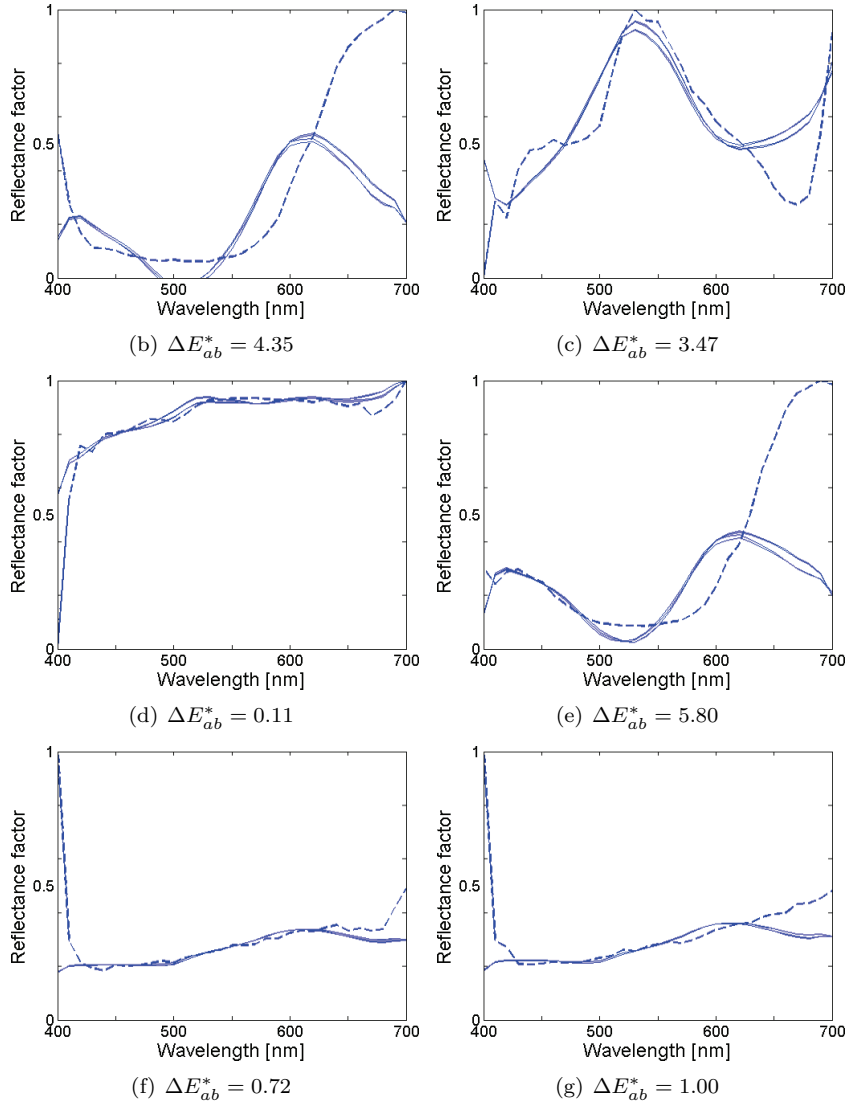


(g)  $\Delta E_{ab}^* = 2.76$

**Figure F.28:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Munsell reflectances and real daylight illuminants. The error function is expressed as  $f_e(\varepsilon_n)$ . They correspond to the lowest angular error  $e_{ang}$ .



(a) The real and retrieved illuminants are represented by the dotted and solid lines, respectively.



**Figure F.29:** Plots (b) to (g) represent the reflectances estimated through the illuminant retrieval. These results were obtained using Natural reflectances and real daylight illuminants. The error function is expressed as  $f_e(\epsilon_n)$ . They correspond to the lowest angular error  $e_{ang}$ .



# Bibliography

- Baer, R. L., Holland, W. D., Holm, J. and Vora, P. (1999), A comparison of primary and complementary color filters for CCD based digital photography, *Sensors, Cameras, and Applications for Digital Photography* **3650(1)**, 16–25.
- Barel, A. O., Clarys, P., Alewaeters, K., Duez, C., Hubinon, J.-L. and Mommaerts, M. (2001), The Visi-Chroma VC-100: a new imaging colorimeter for dermatocosmetic research, *Skin Research and Technology* **7**, 24–31.
- Barnard, K. and Funt, B. (2002), Camera characterization for color research, *Color Research and Application* **27(3)**, 153–164.
- Barnard, K., Cardei, V. and Funt, B. (2002a), A comparison of computational color constancy algorithms - Part I: Methodology and experiments with synthesized data, *IEEE Transactions on Image Processing* **11(9)**, 972–983.
- Barnard, K., Martin, L. and Funt, B. (2000), Colour by correlation in a three-dimensional colour space, *Proceedings of the 6th European Conference on Computer Vision* **1**, 275–289.
- Barnard, K., Martin, L., Coath, A. and Funt, B. (2002b), A comparison of computational color constancy algorithms - Part II: Experiments with image data, *IEEE Transactions on Image Processing* **11(9)**, 985–996.
- Bartelson, J. (1960), Memory colors of familiar objects, *Journal of the Optical Society of America A* **50(1)**, 73–77.
- Brainard, D. H. and Wandell, B. A. (1992), Asymmetric color matching: How color appearance depends on the illuminant, *Journal of the Optical Society of America A* **9(9)**, 1433–1448.
- Buchsbaum, G. (1980), A spatial processor model for object color perception, *Journal of the Franklin Institute* **310(1)**, 1–26.
- Cai, Y. (2002), A novel imaging system for tongue inspection, *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference* **1**, 159–164.
- Caisey, L., Grangeat, F., Lemasson, A., Talabot, J. and Voirin, A. (2006), Skin color and makeup strategies of women from different ethnic groups, *International Journal of Cosmetic Science* **28(6)**, 427–437.

- 
- Chevreul, M. E. (1839), *De la loi du contraste simultané des couleurs (The Principles of Harmony and Contrast of Colours)*, Paris, Pitois-Levrault et Cie.
- CIE (1978), CIE Supplement No 2 to Publication No 15. Recommendations on uniform color spaces, color differences, and psychometric terms.
- CIE (1986), CIE Publication No 15.2. Colorimetry.
- CIE (1987), CIE Publication No 17.4. International Lighting Vocabulary.
- CIE (2004), CIE Publication No 15:2004 Colorimetry.
- Clarys, P., Alewaeters, K., Lambrecht, R. and Barel, A. O. (2000), Skin color measurements: comparison between three instruments: the Chromameter, the DermaSpectrometer and the Mexameter, *Skin Research and Technology* **6**, 230–238.
- Cohen, J. (1964), Dependency of spectral reflectance curves of the Munsell color chips, *Psychonomic Science* **1**, 369–370.
- DiCarlo, J. M., Xiao, F. and Wandell, B. A. (2001), Illuminating illumination, *Proceedings of the 9th IS&T/SID Color Imaging Conference* **1**, 27–34.
- D’Zmura, M. and Iverson, G. (1993a), Color constancy. I. Basic theory of two-stage linear recovery of spectral descriptions for lights and surfaces, *Journal of the Optical Society of America A* **10(10)**, 2148–2165.
- D’Zmura, M. and Iverson, G. (1993b), Color constancy. II. Results for two-stage linear recovery of spectral descriptions for lights and surfaces, *Journal of the Optical Society of America A* **10(10)**, 2166–2180.
- D’Zmura, M. and Iverson, G. (1994), Color constancy. III. General linear recovery of spectral descriptions for lights and surfaces, *Journal of the Optical Society of America A* **11(9)**, 2389–2400.
- Ebner, M. (2007), *Color Constancy*, John Wiley & Sons, Ltd.
- IEC 61966-2-1 (1999), Multimedia systems and equipment - Colour measurement and management - Part2-1:Colour management - Default RGB colour space - sRGB.
- Fairchild, M. (2005), *Color Appearance Models (Second Edition)*, John Wiley & Sons, Ltd.
- Farrell, J., Sherman, D. and Wandell, B. (1994), How to turn your scanner into a colorimeter, *Proceedings of the 10th IS&T’s International Congress on Advances in Non-Impact Printing Technologies* **1**, 579–581.
- Finlayson, G. (1994), Color constancy and a changing illumination, *Human Vision, Visual Processing, and Digital Display* **1**, 6–10.
- Finlayson, G. and Trezzi, E. (2004), Shades of gray and colour constancy, *Proceedings of the 12th IS&T/SID Color Imaging Conference* **1**, 37–41.

- 
- Finlayson, G. D. (1996), Color in perspective, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(10), 1034–1038.
- Finlayson, G. D. and Hordley, S. (2000), Improving gamut mapping color constancy, *IEEE Transactions on Image Processing* **9**(10), 1057–1149.
- Finlayson, G. D., Drew, M. S. and Funt, B. V. (1994a), Color constancy: Generalized diagonal transforms suffice, *Journal of the Optical Society of America A* **11**(11), 3011–3019.
- Finlayson, G. D., Drew, M. S. and Funt, B. V. (1994b), Spectral sharpening: sensor transformations for improved color constancy, *Journal of the Optical Society of America A* **11**(5), 1553–1563.
- Finlayson, G. D., Hordley, S. and Hubel, P. (2001a), Color by correlation: A simple, unifying framework for color constancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(11), 1209–1221.
- Finlayson, G. D., Hordley, S. and Hubel, P. (2001b), Illuminant Estimation for Object Recognition, *Color Research and Application* **27**, 260–270.
- Finlayson, G. D., Hordley, S. and Morovic, P. (2005), Colour constancy using the chromagenic constraint, *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition* **1**, 1079–1086.
- Finlayson, G. D., Hordley, S. and Morovic, P. (2006), CMP-C04-06: Chromagenic Colour Constancy, Technical report, School of Computing Sciences, University of East Anglia.
- Finlayson, G., Funt, B. and Barnard, K. (1995), Color constancy under varying illumination, *Proceedings of the 5th IEEE International Journal of Computer Vision* **1**, 720–725.
- Forsyth, D. (1990), A novel algorithm for color constancy, *International Journal of Computer Vision* **5**(1), 5–36.
- Funt, B., Barnard, K. and Martin, L. (1998), Is machine colour constancy good enough?, *Proceedings of the 5th European Conference on Computer Vision* **1**, 445–459.
- Funt, B., Cardei, V. and Barnard, K. (1995), Learning color constancy, *Proceedings of the 4th IS&T/SID Color Imaging Conference* **1**, 58–60.
- Gershon, R., Jepson, A. and Tsotsos, J. (1988), From [R,G,B] to surface reflectance: Computing color constant descriptors in images, *Perception* **1**, 755–758.
- Goethe, J. W. (1810), *Zur Farbenlehre (Theory of Colors)*, Tübingen: J.G. Cotta.
- Green, K. (2007), Color Matching by Phone, MIT Technology Review, <http://www.technologyreview.com/business/19034>.
- Harville, M., Baker, H., Bhatti, N. and Süssstrunk, S. (2005), Consistent image-based measurement and classification of skin color, *Proceedings of the 2005 IEEE International Conference on Image Processing* **2**, 374–377.

- 
- Hering, E. (1878), *Zur Lehre vom Lichtsinn (On the Theory of Sensibility to Light)*, Vienna.
- Ho, J., Funt, B. V. and Drew, M. S. (1990), Separating a color signal into illumination and surface reflectance components: Theory and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12(10)**, 966–977.
- Hordley, S. (2006), Scene illuminant estimation: Past, present, and future., *Color Research and Application* **31(4)**, 303–314.
- Hordley, S. and Finlayson, G. D. (2004), Re-evaluating colour constancy algorithms, *Proceedings of the 17th IEEE International Conference on Pattern Recognition* **1**, 76–79.
- Hsu, R.-L., Abdel-Mottaleb, M. and Jain, A. K. (2002), Face detection in color images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24(5)**, 696–706.
- Hubel, P. M., Holm, J., Finlayson, G. D. and Drew, M. S. (1997), Matrix calculations for digital photography, *Proceedings of the 5th IS&T/SID Color Imaging Conference* **1**, 105–111.
- Hung, P. and Berns, R. (1995), Determination of constant hue loci for a CRT gamut and their prediction using color appearance spaces, *Color Research and Application* **20**, 285–295.
- Hunt, R. W. G. and Pointer, M. R. (1985), A colour-appearance transform for the CIE 1931 standard colorimetric observer, *Color Research and Application* **10(3)**, 165–179.
- Igarashi, T., Nishino, K. and Nayar, S. K. (2005), The Appearance of Human Skin, Technical report, Department of Computer Science Columbia University New York, USA.
- ISO (2004), ISO 22028-1: Photography and graphic technology - Extended colour encodings for digital image storage, manipulation and interchange.
- Itten, J. (1961), *Kunst der Farbe (The Art of Color)*, Ravensburg : O. Maier.
- Jain, J., Bhatti, N., Baker, H., Chao, H., Dekhil, M., Harville, M., Lyons, N., Schettino, J., and Süssstrunk, S. (2008), Color Match: An imaging based mobile cosmetics advisory service, *Proceedings of the 10th international conference on Human computer interaction with mobile devices and service* **1**, 331–334.
- Judd, D. B. and Wyszecki, G. (1975), *Color in Business, Science, and Industry*, John Wiley & Sons Inc.
- Judd, D. B., MacAdam, D. L. and Wyszecki, G. (1964), Spectral distribution of typical daylight as a function of correlated color temperature, *Journal of the Optical Society of America A* **54(8)**, 1031–1036.



- 
- Kawakami, R., Tan, R. T. and Ikeuchi, K. (2004), A robust framework to estimate surface color from changing illumination, *Proceedings of the Asian Conference on Computer Vision* **2**, 1026–1031.
- Land, E. H. and McCann, J. J. (1971), Lightness and Retinex theory, *Journal of the Optical Society of America A* **61(1)**, 1–11.
- Lee, H.-C. (1986), Method for computing the scene-illuminant chromaticity from specular highlights, *Journal of the Optical Society of America A* **3(10)**, 1694–1699.
- Lu, C. and Drew, M. S. (2006), Practical scene illuminant estimation via flash/no flash pairs, *Proceedings of the 14th IS&T/SID Color Imaging Conference* **1**, 84–89.
- Malacara-Hernandez, D. (2002), *Color Vision and Colorimetry: Theory and Applications*, SPIE Press, Bellingham, WA.
- Maloney, L. T. (1986), Evaluation of linear models of surface spectral reflectance with a small number of parameters, *Journal of the Optical Society of America A* **3(10)**, 1673–1683.
- Maloney, L. T. and Wandell, B. A. (1986), Color constancy: A method for recovering surface spectral reflectance, *Journal of the Optical Society of America A* **3(1)**, 29–33.
- Marguier, J. and Süsstrunk, S. (2008), Illuminant retrieval for fixed location cameras, *Proceedings of the 16th IS&T/SID Color Imaging Conference* **1**, 227–231.
- Marguier, J., Bhatti, N., Baker, H. and Süsstrunk, S. (2009), A Home Décor expert in your camera, *Proceedings of the 17th IS&T/SID Color Imaging Conference* **1**, 85–90.
- Marguier, J., Bhatti, N., Baker, H., Harville, M. and Süsstrunk, S. (2007a), Assessing human skin color from uncalibrated images, *International Journal of Imaging Systems and Technology* **17(3)**, 143–151.
- Marguier, J., Bhatti, N., Baker, H., Harville, M. and Süsstrunk, S. (2007b), Color correction of uncalibrated images for the classification of human skin color, *Proceedings of the 15th IS&T/SID Color Imaging Conference* **1**, 331–335.
- Marimont, D. H. and Wandell, B. A. (1992), Linear Models of surface and illuminant spectra, *Journal of the Optical Society of America A* **9(11)**, 1905–1913.
- Moon, P. and Spencer, D. E. (1944), Geometric formulation of classical color harmony, *Journal of the Optical Society of America A* **34**, 46–50.
- Moroney, N. (2000), Usage guidelines for CIECAM97s, *Proceedings of IS&T 2000 PICS Conference* **1**, 164–168.
- Moroney, N. (2003), A hypothesis regarding the poor blue constancy of CIELAB, *Color Research and Application* **28(5)**, 371–378.

- Moroney, N., Fairchild, M., Hunt, R., Li, C., Luo, M. and Newmann, T. (2002), The CIECAM02 color appearance model, *Proceedings of the 10th IS&T/SID Color Imaging Conference* **1**, 23–27.
- Munsell, A. H. (1905), *A Color Notation*, Boston, G.H. Ellis Co.
- Nakai, H., Manabe, Y. and Inokuchi, S. (1998), Simulation and analysis of spectral distributions of human skin, *Proceedings of the 14th IEEE International Conference on Pattern Recognition* **2**, 1065–1067.
- Nikkanen, J., Gerasimow, T. and King, L. (2008), Subjective effect of white-balancing errors in digital photography, *Optical Engineering* **47(11)**, 113201–1–15.
- Ostwald, F. W. (1916), *Die Farbenfibel (The Color Primer)*, Leiptzig, Unesma.
- Parkkinen, J., Hallikainen, J. and Jaaskelainen, T. (1989), Characteristic spectra of Munsell colors, *Journal of the Optical Society of America A* **6(2)**, 318–322.
- Piérard, G. (1998), EEMCO guidance for the assessment of skin colour, *Journal of the European Academy of Dermatology and Venereology* **10(1)**, 1–11.
- Romero, J., García-Beltrán, A. and Hernández-Andrés, J. (1997), Linear bases for representation of natural and artificial illuminants, *Journal of the Optical Society of America A* **14(5)**, 1007–1014.
- Smith, B., Spiekermann, C. and Sember, R. (1992), Numerical methods for colorimetric calculations: Sampling density requirements, *Color Research and Application* **17(6)**, 394–401.
- Stokes, M., Anderson, M., Chandrasekar, S. and Motta, R. (1996), A standard default color space for the Internet - sRGB.
- Störring, M., Andersen, H. J. and Granum, E. (2001), Physics-based modelling of human skin colour under mixed illuminants, *Robotics and Autonomous Systems* **35(3-4)**, 131–142.
- Swain, M. J. and Ballard, D. H. (1991), Color Indexing, *International Journal of Computer Vision* **7(1)**, 11–32.
- Taylor, S., Westerhof, W., Im, S., and Lim, J. (2006), Noninvasive techniques for the evaluation of skin color, *Journal of American Academy of Dermatology* **54**, S282–S290.
- Tsumura, N., Haneishi, H. and Miyake, Y. (1999), Independent-component analysis of skin color image, *Journal of the Optical Society of America A* **16(9)**, 2169–2176.
- van de Weijer, J. and Gevers, T. (2005), Color constancy based on the gray-edge hypothesis, *Proceedings of the 2005 IEEE International Conference on Image Processing* **2**, 722–5.
- Viola, P. and Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition* **1**, I511–I518.

- 
- von Helmholtz, H. L. (1867), *Handbuch der Physiologischen Optik (Handbook of Physiological Optics)*, Leiptzig, Leopold Voss.
- von Kries, J. (1902), Theoritische Studien über die Umstimmung des Sehorgans, *Festschrift der Albrecht-Ludwig Universität* pp. 145–158.
- Westland, S., Laycock, K., Cheung, V., Henry, P. and Mahyar, F. (2007), Colour harmony, *Colour: Design & Creativity* **1(1)**, 1–15.
- Westland, S., Shaw, J. and Owens, H. (2000), Colour statistics of natural and man-made surfaces, *Sensor Review* **20(1)**, 50–55.
- Whitley, E. and Ball, J. (2002), Statistics review 6 : Nonparametric methods, *Critical Care* **6**, 509–513.
- Wu, W., Allebach, J. P. and Analoui, M. (2000), Imaging colorimetry using a digital camera, *Journal of Imaging Science and Technology* **44(4)**, 267–279.
- Wyszecki, G. and Stiles, W. (2000), *Color Science: Concepts and Methods, Quantitative Data and Formulae (Second Edition)*, John Wiley & Sons, Ltd.



# Curriculum Vitae

**Joanna Marguier**

Audiovisual Communications Laboratory II  
Ecole Polytechnique Fédérale de Lausanne (EPFL)  
1015 Lausanne, Switzerland  
joanna.marguier@a3.epfl.ch

## Personal

Date of birth: September 4, 1978.  
Nationality: Swiss and French.  
Civil status: Single.

## Education

2005 - 2009 **PhD candidate, Computer & Communication Sciences**

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland  
Research topics: Color constancy, Illuminant retrieval

1997 - 2003 **Master's degree, Physics**

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

1993 - 1997 **Science high school diploma with honors**

College de Saussure, Geneva, Switzerland

## Professional Experience

2003-2005 **Swatch Group Research and Development (Asulab)**

Neuchâtel, Switzerland

- Process engineer
- Lead project from process development to industrialization
- Clean room techniques, photolithography, evaporation, vacuum techniques

***Internships:***

- 2008 **Hewlett-Packard Laboratories** (2 months)  
Palo Alto, California, USA
- Processing of photographs to assign object color
  - Application to a Home Décor service
- 2006 **Hewlett-Packard Laboratories** (3 months)  
Palo Alto, California, USA
- Analysis and characterization of color across different capture devices
  - Processing of photographs of people's faces to classify skin color
- 2001 **ABB Research Center** (3 months)  
Baden, Switzerland, Electro-technologies department
- Crystallographic and temperature characterization of superconductors for a fault current limiter

***Academic:***

- 2005-2009 **Teaching Assistant, EPFL**
- Color Imaging for master's students
  - Digital Photography for master's students
  - Student project supervision

**PUBLICATIONS**

- J. Marguier, N. Bhatti, H. Baker, M. Harville and S. Süsstrunk, *Assessing human skin color from uncalibrated images*, International Journal of Imaging, Systems and Technology, special issue on Applied Color Image Processing, Vol. 17, Nr. 3, pp. 143 - 151, 2007.
- J. Marguier, N. Bhatti, H. Baker and S. Süsstrunk, *A Home Décor expert in your camera*, Proceedings of the IS&T/SID 17th Color Imaging Conference, Vol. 1, pp. 85-90, 2009.
- J. Marguier and S. Süsstrunk, *Illuminant retrieval for fixed location cameras*, Proceedings of the IS&T/SID 16th Color Imaging Conference, pp. 227-231, 2008.
- J. Marguier, N. Bhatti, H. Baker, M. Harville and S. Süsstrunk, *Color correction of uncalibrated images for the classification of human skin color*, Proceedings of the 15th IS&T/SID Color Imaging Conference, pp. 331-335, 2007.
- J. Marguier and S. Süsstrunk, *Color matching functions for a perceptually uniform RGB space*, Proceedings of the ISCC/CIE Expert Symposium, 2006.
- D. Alleysson, S. Süsstrunk and J. Marguier, *Influence of Spectral Sensitivity Functions on color demosaicing*, Proceedings of the IS&T/SID 11th Color Imaging Conference, Vol. 11, pp. 351-357, 2003.

## **LANGUAGES**

Native French  
Fluent English  
Conversational German

## **COMPUTER SKILLS**

Programming: MatLab  
Operating systems: Windows, Linux, Unix  
Others: Latex, Photoshop, AutoCad

## **ACTIVITIES & INTERESTS**

Aikido, Cycling, and Hiking