SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

DEPARTMENT OF INFORMATICS, SYSTEMS AND COMMUNICATION
PH.D. PROGRAM IN COMPUTER SCIENCE – CYCLE XXXVI

COMPUTATIONAL COLOR CONSTANCY BEYOND RGB IMAGES: MULTISPECTRAL AND TEMPORAL EXTENSIONS

Ph.D. Dissertation of: Ilaria Erba

Registration number: 795774

Supervisor: Ph.D. Marco Buzzelli

Co-Supervisor: Prof. Raimondo Schettini

Tutor: Prof. Francesca Arcelli

Ph.D. Coordinator: Prof. Leonardo Mariani

ACADEMIC YEAR 2022-2023

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Milano Bicocca or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Milano Bicocca or any other University or similar institution except as declared in the Preface and specified in the text.

Ilaria Erba
October, 2023

Abstract

Computational color constancy beyond RGB images multispectral and temporal extensions

Ilaria Erba

When it comes to visual perception, there are notable differences between the ways in which humans and machines interpret and understand images. Unlike image acquisition systems, the human eye can perceive object colors accurately regardless of the light source's color cast. To achieve a similar effect in digital images, a pre-processing step called Computational Color Constancy is used. Its purpose is to render images as if they were captured under a known light source. This problem is also important for those computer vision applications that rely on the coherence of objects' colors. Unfortunately, a unique solution to this problem is unattainable. However, the scientific community has made significant efforts in developing both generalized and environment-specific solutions.

Over the past few years, the cost of spectral sensors has decreased, making them more accessible. So much so, that the first patents for the introduction of low-resolution spectral sensors in smartphone digital cameras have been published. The acquisition of spectral images is still influenced by the light source, and when the acquisition takes place in an uncontrolled environment, the availability of a reliable algorithm for computational color constancy becomes even more important. Therefore, the focus of the scientific community partly shifted towards the estimation of a spectral illuminant, in order to provide a solution for the acquisition of spectral images even in uncontrolled environments. One of the main purposes of this work is to improve the accuracy of color illuminant estimates by utilizing spectral information. To achieve this, two effective strategies have been proposed. The first approach involves employing established statistical-based algorithms to estimate illuminants. Subsequently, four innovative re-elaboration methods have been introduced. They utilize these spectral estimations as input and generate an improved version of the estimations in the color domain. On the other hand, the second strategy involves utilizing both spectral and color information to enhance color illuminant estimation.

The problem of Computation Color Constancy is complex and has many different aspects and applications. However, one area that has not received much attention from the

scientific community is the temporal domain. While computational color constancy aims to accurately render images taken under known light sources, temporal color constancy adds an additional challenge of ensuring that the color of objects remains consistent across all frames. Nonetheless, it also provides a temporal sequence of frames that contain valuable information that can be utilized to improve the illumination estimation. One solution that has been adopted so far is to apply computational color constancy algorithms to each frame individually. However, this approach can lead to the creation of artifacts in which the color of an object changes from frame to frame. To avoid these kinds of issues, it is necessary to define a metric that can identify them. This thesis not only provides such a metric but also analyzes the temporal stability of some of the computational color constancy algorithms when applied in a single-frame manner. Additionally, this thesis also provides a temporal color constancy method. These methods usually estimate the illuminant of a selected frame, also called shot-frame, by exploiting information extracted from previous frames. In this work, this assumption is extended. The method provided takes a window of frames to return an illuminant estimation for each frame. The assumption underlying the method is that not only the previous frames can be beneficial to the estimation of the illuminant of the shot frame, but the assumption is generalized to the adjacent frames.

Contents

Chapter 1

Introduction

1.1 Problem Statement

Photography has undergone a remarkable transformation from its early days of analogical film to the digital age we live in today (1.1). Today, anyone can capture a digital image with ease, thanks to the digital cameras in our smartphones (but also in digital devices such as mirrorless cameras, compact cameras, and so on). As the smartphone market expanded, so did the expectations of buyers who seek to acquire visually appealing digital images with minimal effort. One of the main challenges in making a picture appear more realistic is the color adjustment based on white balancing. For example, the acquisition of a picture of a white object in certain lighting conditions can appear bluer than how it is perceived visually. It is called "white balance" but it actually affects all the colors in the photo. All digital devices come with a white balancing step. In the simplest case, the device will be calibrated to a specific illuminant type, usually D65, which corresponds to the average midday light. As a result, the captured picture will appear well-balanced only when taken under the selected lighting conditions. If the picture is taken with natural light from an overcast sky, it will have a cold color temperature, with more of a blue tone. Shooting in unnatural incandescent light with tungsten light bulbs creates a warm color cast, which appears more yellow or orange. Similarly, shooting under certain types of fluorescent lights can give a green tint to the pictures. To counteract this, it is possible to use a more sophisticated camera's white balance setting or post-production software.

In the realm of photography, it is essential to produce images that are as adherent as possible to human visual perception. This is particularly important in product or food photography, where the accurate representation of colors is crucial. To achieve this level of precision, photographers utilize a tool called a grey card. A grey card is simply a square material that helps to achieve the correct white balance by providing a neutral reference point for the camera to adjust the colors.

Not all environments are controlled and allow for color measurements. Fortunately,

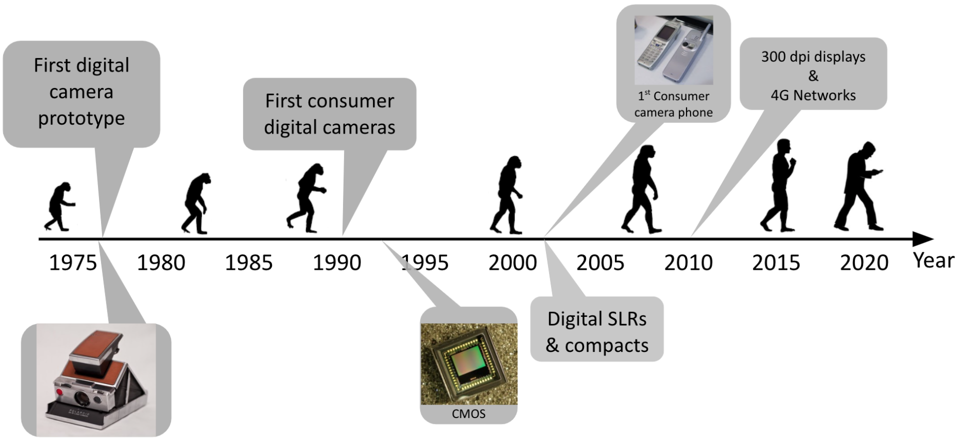

First digital camera prototype
First consumer digital cameras
1st Consumer camera phone
300 dpi displays & 4G Networks
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
Year
CMOS
Digital SLRs & compacts

Figure 1.1: An incomplete timeline of photographic innovations between the years 1975 and 2020. Image taken from [28].

digital photography has transformed the way we capture, edit, and share images. One significant advantage of digital photography over analog photography is the ability to adjust the white balance after the image is taken. In some situations, an automatic white balance algorithm is necessary. For example, in medical imaging, the color representation of tissues and organs must be independent of the illuminant source. Similarly, in security and surveillance, the color representation of objects is vital for identifying individuals and objects.

The process of discounting possible color casts from images is known as color constancy and is commonly used in digital images. Despite being a well-studied problem, there is no single solution that works for all scenarios due to the ill-posed nature of the problem as described in section 1.2. Therefore, the scientific community is still searching for generalized solutions or processing pipelines that can be applied to specific situations. In this work, we focus on two of the possible extensions of the problem: the multispectral and the temporal ones.

Multispectral imaging has become increasingly popular in recent years, thanks to advancements in sensor technology, which have made it more affordable for both researchers and practitioners. As multispectral sensors continue to decrease in cost and become more widely available, the scientific community's interest in this technology continues to grow. Multispectral imaging provides more spectral information as compared to color images. Applications for multispectral imaging span a wide range of fields, including remote sensing and medical imaging. However, multispectral imaging can also be applied to color constancy. So far, the scientific community focused on providing a multispectral color constancy algorithm for automatic white balancing of multispectral imaging. Such a tool would allow for a real-life color acquisition of multispectral imaging in an uncontrolled environment. In this work, we present two methods; one is based on the re-elaboration of multispectral illuminant estimations obtained thanks to multispectral-extended statistical

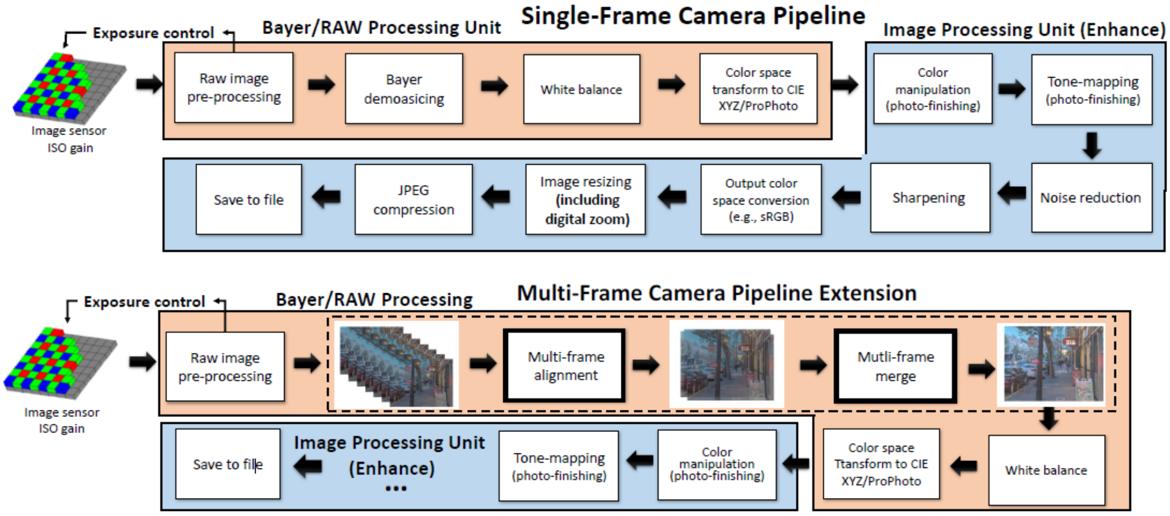

Single-Frame Camera Pipeline
Exposure control
Bayer/RAW Processing Unit
Image Processing Unit (Enhance)
Image sensor
ISO gain
Raw image pre-processing
Bayer demoasicing
White balance
Color space transform to CIE XYZ/ProPhoto
Color manipulation (photo-finishing)
Tone-mapping (photo-finishing)
Noise reduction
Sharpening
Output color space conversion (e.g., sRGB)
Image resizing (including digital zoom)
JPEG compression
Save to file
Multi-Frame Camera Pipeline Extension
Exposure control
Bayer/RAW Processing
Image sensor
ISO gain
Raw image pre-processing
Multi-frame alignment
Mutli-frame merge
White balance
Color space Ttansform to CIE XYZ/ProPhoto
Color manipulation (photo-finishing)
Tone-mapping (photo-finishing)
Image Processing Unit (Enhance)
•••
Save to file

Figure 1.2: The top of this figure shows a standard single-frame camera pipeline. The bottom figure shows the extension to multi-frame (or burst imaging) used by most modern smartphone cameras. Image taken from [28].

methods [29], while the other proposes a neural network for the combination of color and low-resolution spectral information [30]. This thesis proves that it is possible to improve color constancy through spectral images.

With the advancements in digital camera technologies, it has become also possible to capture videos, and as the expectations of customers for better quality images increased so did the ones related to video quality. One of the main constraints that videos need to respect, is the consistency of color between frames. In fact, the acquisition of digital videos is sensitive to this kind of artifact, especially if it is poorly corrected by naive algorithms that operate on a frame-by-frame basis. Most modern digital cameras already have a multi-frame pipeline that performs a color correction frame by frame (Figure 1.2). Therefore, the second part of this thesis focuses on temporal consistency. In particular, a general-purpose procedure to assess the temporal consistency of well-known state-of-the-art methods is provided [17], along with a temporal color constancy method. This method exploits neural networks, and especially convolutional neural networks to extract spatial features, and LSTM to capture spatially recurrent features.

1.2 Computational Color Constancy

The appearance of objects is greatly influenced by several factors including the material they are made of, their shape, and the type of light that illuminates them within the waves of the electromagnetic spectrum. To provide a clearer understanding of this phenomenon, Figure 1.3 shows a visual example. In a real-life scenario, the lights frequently change due to meteorological factors. Therefore, the perception of objects is regularly altered by the

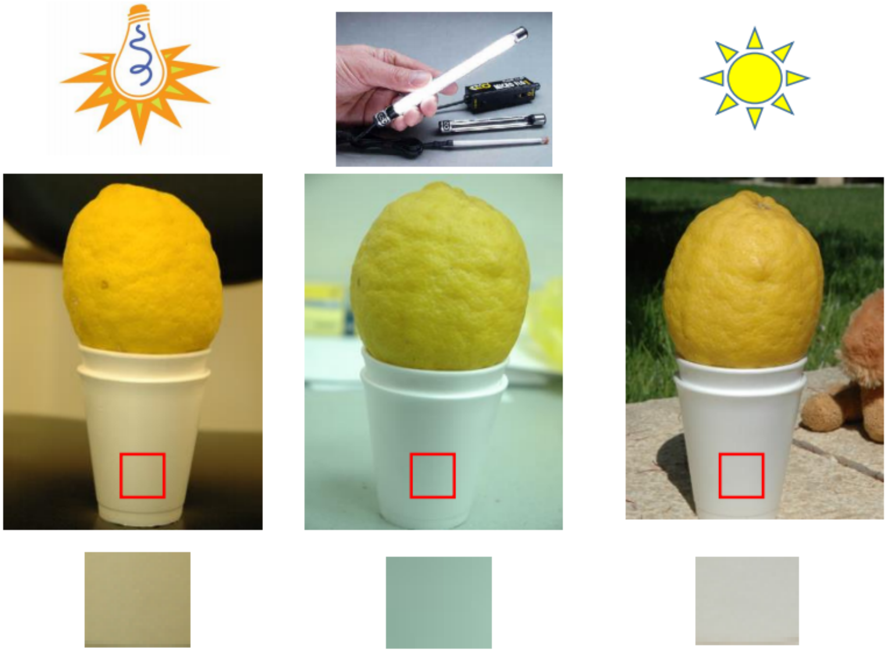


Figure 1.3: Close-up look of the color of a cup under different illuminations. Image taken from [57]

illumination they are exposed to. The human visual system has the ability to perceive the color of objects constantly, despite the change in the illumination. This ability is called color constancy [34]. It is not necessary for the human brain to know the source of the illumination in order to perceive constant the color of objects. This capability is even more fascinating if we consider that, in real-life scenarios with multiple objects, the illumination can result from a mixture of direct and indirect irradiation distributed over a range of incident angles and mutual reflections.

In the last centuries, many experiments have been performed with the only purpose of understanding how color constancy works. Despite the many efforts we still are not able to accurately answer this question, but theories regarding why this phenomenon happens have been formulated. In particular, two explanations have been advanced [34]. According to the first theory, Color constancy is the result of unconscious inference, which enables a coherent perception of the world, enabling the observer to recognize objects despite the change in illumination. As per the second theory, instead, color constancy is a result of sensory adaptation, allowing the observer to make assumptions about the time of day and weather.

It is important to note that digital devices are unable to discount or ignore illuminants, which results in a significant impact on the color of the images captured or displayed. This leads to discrepancies between how the observer perceives colors in the actual scene versus the captured scene. Therefore, to display an image coherent with the observer's

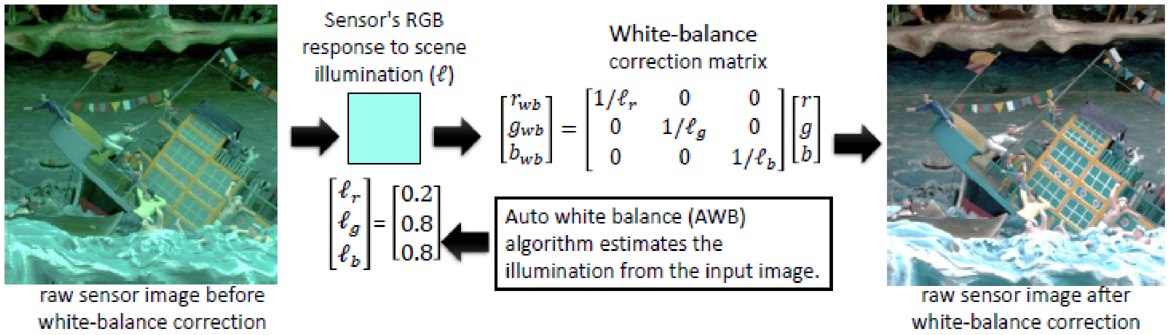

Sensor's RGB response to scene illumination (ℓ)
White-balance correction matrix
$\begin{bmatrix} r_{wb} \\ g_{wb} \\ b_{wb} \end{bmatrix} = \begin{bmatrix} 1/\ell_r & 0 & 0 \\ 0 & 1/\ell_g & 0 \\ 0 & 0 & 1/\ell_b \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$
$\begin{bmatrix} \ell_r \\ \ell_g \\ \ell_b \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.8 \end{bmatrix}$
Auto white balance (AWB) algorithm estimates the illumination from the input image.
raw sensor image before white-balance correction
raw sensor image after white-balance correction

Figure 1.4: Example of color constancy process. The first step estimates the illuminant from the given image, while the second step uses the illuminant to correct the image. Image taken from [28]

perception, we need a pre-processing step to discount the illuminant from the acquired image.

This pre-processing step is called Computational Color Constancy and can be explained as a two-step problem [41]. In the first step, the illuminant is extracted from a scene, and in the second step, the illuminant is used to render the image as if it were acquired under a canonical illuminant. The illuminant estimation problem is formulated as an inverse problem, that aims at reversing the commonly accepted imaging model [4] and separating the reflectance of the object from the illumination

$$I_k(x, y, \lambda) = \int_{\omega} L(\lambda)R(x, y, \lambda)S_k(\lambda)d\lambda, \tag{1.1}$$

where $R(x, y, \lambda)$ is the surface reflectance, $L(\lambda)$ the illumination , and $S_k(\lambda)$ the sensor characteristics, as a function of the wavelength λ, over the visible spectrum ω. The subscript k represents the sensor's response in the k^{th} channel and $I_k(x, y, \lambda)$ is the image corresponding to the k^{th} channel (k = R,G,B).The model being referred to in this context is a simplified version of what actually happens in real-world scenarios. One of the main simplifications made in this model is with regard to the illuminant L. According to the formula used in this model, the illuminant is considered to be global and as such, it is assumed to be constant across all spatial coordinates (x, y).

The following step is the correction process which is often carried out through a von Kries-like transform [101], using a diagonal 3 × 3 matrix to apply independent correction to the response of cone photoreceptors. Although known to be suboptimal and unable to fully handle metameric effects [75], the von Kries transform is commonly adopted due to its simplicity. A visual example of the full process of computational color constancy is provided in Figure 1.4.

Computational Color Constancy is a problem that has been extensively studied and has received a lot of attention in the state of the art. However, it is still an ill-posed problem, which means that there is no unique solution. For a problem to be well-posed, three conditions must be met[27]: a solution must exist, the solution must be unique, and the unique solution must be stable. Moreover, in color constancy, it is difficult to guarantee the uniqueness and stability of the solution because of the strong correlation between the color in the image and the color of the illuminant. This correlation leads to imprecise estimation, and even a slight variation in the correlation can result in a significant variation in the estimation.

The issue of Computational Color Constancy remains significant given the lack of a unique solution due to its ill-posed nature, another important factor is that it plays an important role in color-based computer vision applications, such as digital imaging, object recognition, tracking and image classification.

Over the decades numerous methods [100, 54, 24, 1, 10] have been proposed for Computational Color Constancy, however, no unique solution has been identified. Due to the ill-posed nature of the problem, in fact, color constancy requires the formulation of specific assumptions on the imaged content or the reliance on data-driven biases. To this extent, different methods adopt different strategies. Color constancy methods can be mainly classified into statistical and machine-learning methods. Statistical methods [100, 24] estimate the illuminant of the image by making assumptions about the color features of the image itself. For example, the max-RGB algorithm assumes the presence of a white surface object in the scene [66]. These methods have fast execution time, however, the resulting performance heavily depends on the underlying assumptions. Machine learning algorithms [54, 1, 10], instead, establish the relationship between the image color distribution and the illuminant through a supervised learning process, meaning they do not need to rely on statistical assumptions and therefore are more adaptive. However one of the shortcomings of supervised algorithms is that they are highly dependent on the dataset used to train them [22]. Potentially, they may need to be re-trained on different datasets to overcome data-related bias.

The human visual system is a complex mechanism that relies on three distinct types of cones located in the eye [109, 45]. Each of these cones is sensitive to specific wavelengths of light in the visible spectrum, which are then processed by the brain to form our perception of color. This process involves the brain combining data from the three cones to create the full range of colors that we are able to see. Color cameras work on the same principle, having three filters, named as R, G and B filters. They are sensitive to particular regions of wavelengths and their combination forms a color image.

1.2.1 Evaluation Measures

Computational color constancy algorithms are designed to estimate the original illuminant vector or the expected illuminant of a scene. The performance of these algorithms is evaluated by computing the distance between the original illuminant vector and the estimated one. The most commonly used distance measure is the recovery angular error, which is computed in the normalized RGB color space based on mathematical principles. However, mathematical principles may not always reflect human perception accurately. To determine the correlation between angular error and human perception, Gijsenij et al. [40] conducted a study. They found that the angular error is a reasonably good indicator of the perceptual performance of color constancy algorithms. While other distance measures based on the principles of human vision could be defined, the evaluation benchmark for computational color constancy algorithms is based on the angular error distance measure, which is a crucial factor in ensuring the accuracy of these algorithms.

The recovery error [51, 40] is defined as:

$$e_{rec}(U, V) = \arccos\left(\frac{U \cdot V}{|U||V|}\right) \tag{1.2}$$

where "·" indicates the dot product, $|x|$ is the euclidean norm, U denotes the RGB illuminant target, and V is the RGB estimated illuminant.

Finlayson et al. [32] proposed a new metric for evaluating illuminant estimation algorithms, called the reproduction angular error, which measures the angle between the reproduction of a true achromatic surface under a white light with the actual reproduction of an achromatic surface when an estimated illuminant color gets discounted. The reproduction angular error is defined as follows:

$$e_{rep}(U, V) = \arccos\left(\frac{\frac{U}{V} \cdot (1, 1, 1)}{|\frac{U}{V}|\sqrt{3}}\right) \tag{1.3}$$

again, "·" indicates the dot product, $|x|$ is the euclidean norm, U denotes the RGB illuminant target, and V is the RGB estimated illuminant.

1.3 Related Works For Single Frame RGB Color Constancy

Over the decades numerous methods ([100, 54, 24, 1, 10]) have been proposed for RGB illuminant estimation, however, no unique solution has been identified. Due to the ill-posed nature of the problem, in fact, color constancy requires the formulation of specific assumptions on the imaged content or the reliance on data-driven biases. To this extent, different methods adopt different strategies. Color constancy methods can be mainly classified into statistical and machine-learning methods.

Statistical methods, such as Grey-World [15], White-Patch [66], Shades-of-Grey [31], and Gray-Edge [100] estimate the illuminant of the image by making assumptions about the statistical properties of the scene. For example, the gray world algorithm is one of the oldest and simplest color constancy algorithms. It is based on the assumption that the color in each sensor channel averages to gray over the entire image. The gray world algorithm [15] estimates the deviation from the assumptions and is given by a simple expression:

$$\begin{aligned} l_r &= mean(E_R), \\ l_g &= mean(E_G), \\ l_b &= mean(E_B) \end{aligned} \tag{1.4}$$

where l_r, l_g, l_b are the mean value in each channel respectively and E_R, E_G, E_B are individual image channels. In the White-Patch [66] algorithm, the estimate of the illuminant is obtained by measuring the maximum of the responses in each channel. The estimation formulation is very similar to that of the grey world algorithm in equation 1.4, except for the fact that the mean is replaced by the maximum of the sensor responses in each channel. These methods have fast execution time, however, the resulting performance is heavily dependent on the underlying assumptions.

More recently, machine learning algorithms have been proposed for illuminant estimation [54, 1, 10]. Unlike statistical-based algorithms, that relies on statistical assumptions to estimate the illuminant, machine learning algorithms establish the relationship between the image color distribution and the illuminant through a supervised learning process, meaning they do not need to rely on statistical assumptions and therefore they are more adaptive.

Bianco et al. [13] proposed a color constancy method using Convolutional Neural Networks (CNNs). They trained a CNN on a large dataset of images with ground truth illuminant color information to estimate the illuminant color from an input image. Hu et al. [54] proposed a fully convolutional color constancy method called FC4. Their method uses a fully convolutional neural network to estimate the illuminant color spatial distribution of an input image that is used to correct the input image for color constancy. More recently, alternative approaches for convolution-free deep learning have been applied to illuminant estimation as well Li et al. [68] proposed a transformer-based multiple illuminant color constancy method called TransCC. The method uses a transformer-based network to estimate the illuminant color distributions of an input image under multiple illuminants. The generative nature of the method is what enables the handling of multiple illuminant sources, at the same time however introducing potential artifacts in the output white-balanced images. However one of the shortcomings of supervised algorithms is that they are highly dependent on the dataset used to train them [21]. Potentially, they may need to be re-trained on different datasets to overcome data-related bias.

Forsyth et al. [33], and later Gijsenij et al. [41] introduced gamut-based methods for color constancy. These are based on the assumption that in real-world images, for a given illuminant, one observes only a limited number of colors. Consequently, any variations in the colors of an image (i.e., colors that are different from the colors that can be observed under a given illuminant) are caused by a deviation in the color of the light source. This limited set of colors that can occur under a given illuminant is called the canonical gamut image. Gamut-based methods have a sensitivity to the scene content similar to that of methods based on lower-level statistics, combined with a non-negligible computational complexity, especially when handling large-resolution images.

Chapter 2

Multispectral Color Constancy

Spectral imaging has been rapidly advancing over the past two decades and is divided into two distinct imaging methods: multispectral and hyperspectral imaging. Although these terms are often used interchangeably, they each have their own application spaces. Both technologies offer advantages over conventional machine vision imaging methods that are limited to acquiring only three spectral bands from the visible spectrum (400-700 nm). However, the benefits of spectral imaging come with an increased system complexity in terms of optical design.

Spectrometers are essential instruments in spectral imaging, as they allow for the measurement of the spectral content of light and electromagnetic radiation. This measurement results in a spectrally resolved image of an object or scene, which is often referred to as a datacube due to the three-dimensional nature of the data. There are various types of imaging spectrometers, such as filtered cameras, whisk-broom scanners, and push-broom scanners, among others. However, these instruments will not be analyzed further in this thesis since they diverge from the purpose of this work [36]. Multispectral and hyperspectral cameras differ primarily by the number of bands they record and the width of these bands. Hyperspectral imaging (HSI) captures and analyzes a large number of narrow, contiguous bands across the electromagnetic spectrum, resulting in a high-resolution spectrum for each pixel in the image. On the other hand, a multispectral imaging (MSI) system captures several preselected and discrete wavebands, as opposed to the continuous wavelength data collection in HSI, as shown in Figure 2.1. Both imaging systems result in the acquisition of a data cube. While a hyperspectral data cube has more than 100 channels, a multispectral one has fewer ones, in the order of tenths.

Hyperspectral imaging provides more detailed data than multispectral imaging, allowing for more specific analysis and accurate identification of materials and substances. While some consider MSI to be inferior to HSI due to lower spectral resolution, the two technologies each have their own advantages that make them a preferred tool for different tasks.

HSI is best suited for applications that require sensitivity to subtle differences in signal

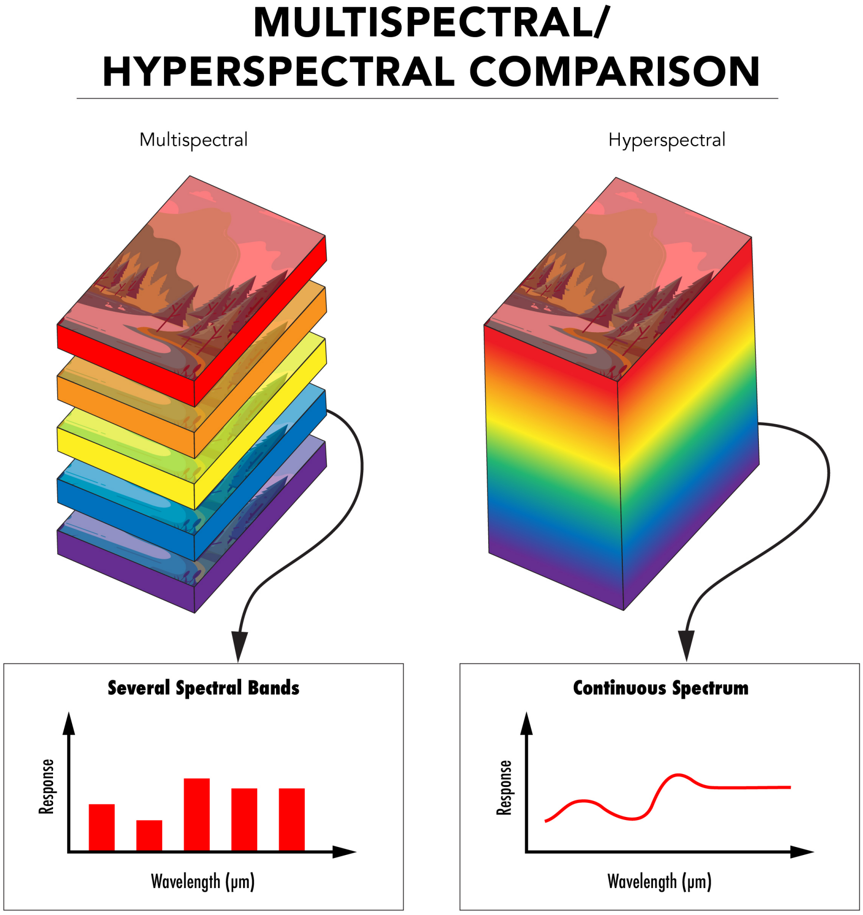


Figure 2.1: The acquisition techniques of MSI and HSI are different from each other. In the MSI technique, the system captures only a limited number of wavebands in a discrete manner. On the other hand, in the HSI technique, the system acquires a larger number of wavelengths in a contiguous manner, resulting in a higher resolution spectrum for each pixel in the image. Image taken from [26].

along a continuous spectrum, which could be missed by a system that samples larger wavebands. However, if there is less spectral information to acquire, the image capture, processing, and analysis can happen more quickly, allowing the imaging technique to avoid a noisy acquisition.

The application spaces that require the uses of HSI and MSI continue to grow in number. Remote sensing, aerial imaging of the earth's surface with the use of unmanned aerial vehicles (UAVs) and satellites, has relied on both HSI and MSI for decades. Spectral photography can penetrate through Earth's atmosphere and different cloud cover for an unobscured view of the ground below. This technology can be used to monitor changes in population, observe geological transformations, and study archeological sites [99]. The same is true in the medical field. Non-invasive scans of skin to detect diseased or malignant cells can now be performed by doctors with the help of hyperspectral imaging. Certain wavelengths are better suited for penetrating deeper into the skin, allowing a more detailed understanding of a patient's condition. Cancers and other diseased cells are now easily distinguishable from healthy tissue, as they will fluoresce and absorb light under the correct stimulation [93]. Although application spaces that benefit from HSI and MSI are large and increasing, limitations in the current technology have led to slow industry adoption. Currently, these systems are significantly more expensive compared to other machine vision components. The sensors need to be more complex, have broader spectral sensitivity, and must be precisely calibrated.

2.1 Related Works

Computational color constancy is a deeply explored field. A large section of the scientific literature approaches the problem in the RGB domain. However, since the works presented in the following sections aim at exploiting multispectral information, this section specifically addresses works that connect spectral imaging (including multispectral and hyperspectral) with illuminant estimation.

The use of spectral imaging techniques increased in the last years and their involvement has proven to be beneficial for several fields related to computer vision.

Lenz et al. [67] investigated the tasks of illuminant estimation and color correction with the aid of multispectral representation. Specifically, they approximate the spectral description of the scene pixels with a linear combination of bases from a dataset of known spectra. They then characterize the image through the mode of such combination coefficients, which is assumed to represent the global illuminant change.

Li et al. [69] proposed an end-to-end unrolling network architecture to estimate both single and multiple illuminants in the input image, casting the problem as a constrained matrix factorization. They also constructed a large spectral image dataset for training and evaluation.

Su et al. [97] proposed a separation of the reflectance and illumination components using a weighting scheme, factorizing the weighted specular-contaminated pixels to estimate the illumination spectrum. Despite the demonstrated robustness in both simulation and real experiments, it is computationally expensive, since this approach requires a number of iterations for the spectral illuminant estimation of a single image.

Zheng et al. [110] proposed a method that models the separation of illuminant from reflectance as a low-rank matrix factorization task, and developed a scalable algorithm that works in the presence of model error and image noise. They demonstrated that taking advantage of the greater color variety offered by hyperspectral images can improve separation accuracy, and relax the restrictive subspace illumination assumption in the existing literature, thus providing supporting evidence for the method proposed in this work.

Tong et al. [98] proposed a general framework to estimate the spectrum of the illumination from specular information in a single hyperspectral image. By utilizing a specular independent subspace they separated the reflectance components and shaped a weighting scheme in order to find specular-contaminated pixels so that the illumination can be directly estimated by factorizing them.

Khan et al. [59, 60] investigated the use of illuminant estimation algorithms for multispectral imaging systems to overcome the difficulty in the calibration of multispectral devices. To address this problem, the authors propose directly extending computational color constancy algorithms to multispectral imaging, including edge-based methods [100] as well as highlight-based methods [42]. In subsequent works [58, 61] they developed a spectral adaptation transform to bring the multispectral image data into a canonical representation, effectively performing illuminant correction. They illustrated the potential benefits of using multispectral imaging in computer vision applications but also acknowledged that multispectral imaging can still be sensitive to changes in illumination.

Robles-Kelly et al. [89] presented a convolutional neural network to recover pixel-wise illuminant in multispectral images. The network takes in input a tensor which is constructed by making use of an image patch at different scales in order to allow the network to predict the pixel-wise illuminant using locally-supported multiscale information.

More recently, Kitanovski et al. [62] developed an imaging pipeline for a spectral filter array camera to estimate scene reflectances in the absence of knowledge about the scene illuminant. The proposed approach involves estimating the illuminant's spectral power radiance, which is shown to stabilize and marginally improve the estimation accuracy compared to the method that estimates the illuminant in the RGB domain only.

Unlike more traditional computational color constancy algorithms, that estimate illuminants from RGB images, Koskinen et. al. [63] propose to use the average color spectra of a scene. They tested several regression functions (such as Kernel Ridge, Random Forest, and Multilayer Perceptron) to map the spectral pixel to the white point. They demonstrate that the method is effective even with as few as 10-14 spectral channels.

2.2 RGB Color Constancy Using Multispectral Pixel Information

In this work [29], has been addressed the problem of RGB color constancy by exploiting richer multispectral input image data, motivated by the ever-increasing availability of multispectral imaging devices [83, 80, 50, 35, 84, 108, 23, 95, 81, 6, 7, 8], which are applied in a variety of computer vision, remote sensing, and medical imaging applications [76, 74]. Multispectral reconstruction methods [3, 73] are also gaining traction in the scientific community, further incentivizing working in this domain. However, spectral reconstruction is not yet considered to be a mature field, with several open issues [72, 71]. In order

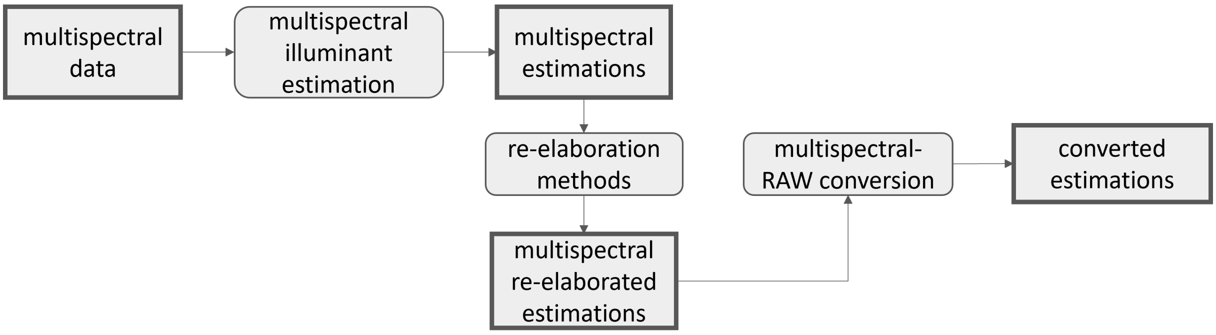

multispectral data
multispectral illuminant estimation
multispectral estimations
re-elaboration methods
multispectral re-elaborated estimations
multispectral-RAW conversion
converted estimations

Figure 2.2: Pipeline used to retrieve the multispectral estimations and their re-elaborated version starting from the multispectral data provided by the dataset.

to investigate the practical utility of multispectral information, this research has been developed under the assumption of data that either come from a multispectral acquisition device, or from a hypothetical perfect multispectral reconstruction method. Khan et al. [59, 61] demonstrated the advantages of estimating and processing illuminants in the multispectral domain. Taking this work as inspiration and as a starting point, our final color correction takes place in the RGB domain instead, since this is the underlying color model of many viewing devices for end-user consumption, and the von Kries-like transform is among the most basic and widely-supported correction models in color imaging pipelines and in color management systems [28, 11].

Inspired by the works of Khan et al. [59, 61], this work is posed as an investigation to verify whether multispectral information is useful to improve traditional RGB color constancy methods. To this end, has been investigated how to extend a selected number of sensor-independent color constancy methods from the RGB domain to an N-dimensional multispectral domain. Showing that this extension is not sufficient to achieve computational color constancy, the work proposes to convert the resulting multispectral estimations to the RGB domain under the assumption that re-elaborating the multispectral illuminant estimation may improve the raw RGB converted result, an investigation on several re-elaboration methods is conducted. The proposed method for color constancy using multispectral pixel information reaches an improvement of 60% in mean reproduction error on the NUS dataset by Nguyen et al. [83], when compared to the corresponding RGB methods. In Figure 2.2 the pipeline for the extraction and the re-elaboration of the multispectral and raw RGB estimations is shown.

2.2.1 Proposed method for multispectral-based RGB illuminant estimation

This section is dedicated to the description of the method proposed to exploit multispectral information for raw RGB illuminant estimation. This procedure is carried out by extending a set of camera-independent color constancy algorithms, originally devised for the raw RGB domain, to the multispectral domain. This has been done consistently with what was proposed by Khan et al. [59]. Multispectral estimation of the illumination should be then mapped into raw RGB for performing color correction. To this end, a straightforward solution for the multispectral-to-raw conversion consists in exploiting the camera sensitivity function to the multispectral illuminant estimation.

Alternatively, the work proposes that an unequal contribution of the estimated multispectral illuminant bands may benefit the eventual raw RGB estimation. This process, which from now on will be referred to as "multispectral illuminant estimations re-elaboration" is discussed mainly in Section 2.2.1.2. The concept that is shared among the multispectral illuminant estimations re-elaboration methods is that there exists a mapping (e.g. in the form of weights or biases) that, applied to the multispectral estimation, reduces the distance between the converted multispectral-based raw RGB estimation and the expected raw RGB illuminant.

2.2.1.1 Multispectral Illuminant Estimation Algorithms

This work takes into consideration six algorithms belonging to the edge-based color constancy framework (EB), introduced in 2007 by van de Weijer et al. [100] as a generalization of multiple algorithms based on low-level image statistics. The general equation for estimation of the illuminant, according to this framework, is

$$\left(\int \left|\frac{\partial^n f^\sigma(x)}{\partial x^n}\right|^p dx\right)^{\frac{1}{p}} = ke^{n,p,\sigma}. \tag{2.1}$$

This operation is executed on the separate RGB channels

$$\begin{aligned}\left(\int \left|\frac{\partial^n f^\sigma(x)}{\partial x^n}\right|^p dx\right)^{\frac{1}{p}} = \Bigg(&\left(\int \left|\frac{\partial^n R^\sigma(x)}{\partial x^n}\right|^p dx\right)^{\frac{1}{p}}, \\ &\left(\int \left|\frac{\partial^n G^\sigma(x)}{\partial x^n}\right|^p dx\right)^{\frac{1}{p}}, \\ &\left(\int \left|\frac{\partial^n B^\sigma(x)}{\partial x^n}\right|^p dx\right)^{\frac{1}{p}}\Bigg)\end{aligned} \tag{2.2}$$

This framework generates different estimations for the illuminant color, based on three variables

- n identifies the spatial derivatives order, which is typically set between 0 (no derivative, as in the case of the grey world algorithm), and 2 (for second order derivative).
- Minkowski norm p determines the relative weights of the multiple measurements from which the final illuminant color is estimated. For example, with $p = 1$, the

illuminant is derived by an averaging operation over the derivatives of the channels. For $p = \infty$, the illuminant is computed from the maximum of the derivatives in the scene.

- σ denotes the scale of the local measurements, specifying the intensity of the smoothing operation via Gaussian filtering.

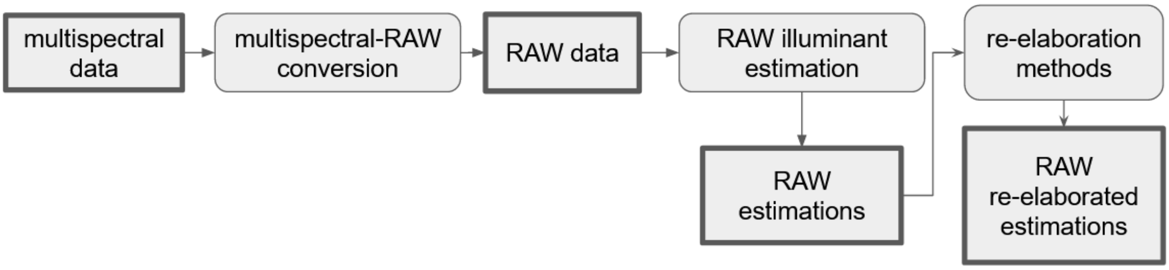
The three parameters of these methods (the spatial derivatives order n, the Minkowski norm p, and standard deviation σ) have been set as proposed in [12]

- grey world (GW) $n = 0$, $p = 1$, $\sigma = 0$.
- white point (WP) $n = 0$, $p = \infty$, $\sigma = 0$.
- shades of grey (SoG) $n = 0$, $p = 4$, $\sigma = 0$.
- general grey world (GGW) $n = 0$, $p = 9$, $\sigma = 9$.
- 1st order grey edge (GE1) $n = 1$, $p = 1$, $\sigma = 6$.
- 2nd order grey edge (GE2) $n = 2$, $p = 1$, $\sigma = 1$.

The RGB color constancy algorithms described in Equation 2.2 is extended to operate on an arbitrary number of dimensions N, so that they can be applied to multispectral images, producing a multispectral illuminant estimation. In the following, these algorithms will be referred to as the spectral counterpart of EB algorithms (e.g. grey World becomes Spectral grey World).

2.2.1.2 Multispectral and RGB Illuminant Estimations Re-elaboration

Anticipating the experimental results, it is notable that the sole multispectral extension of raw RGB illuminant estimation algorithms is not sufficient to achieve color constancy. A hypothesis is formulated as follows the multispectral-to-raw conversion may improve the raw RGB estimation by adopting an unequal contribution from the N multispectral bands. The approach consists of learning a N-channel modifier to apply to the multispectral illuminant estimation, before converting it through the camera sensitivity function. To verify that the improvement derived from the re-elaboration of the multispectral estimation is due to the multispectral information and not to the re-elaboration method itself, re-elaboration methods are also applied to the raw RGB estimations. In order to make the process steps more understandable, Figure 2.3 shows the baseline of re-elaboration for raw RGB illuminant estimation.



multispectral data
multispectral-RAW conversion
RAW data
RAW illuminant estimation
re-elaboration methods
RAW estimations
RAW re-elaborated estimations

Figure 2.3: Pipeline used to retrieve the raw RGB estimations and their re-elaborated version starting from the multispectral data provided by the dataset.

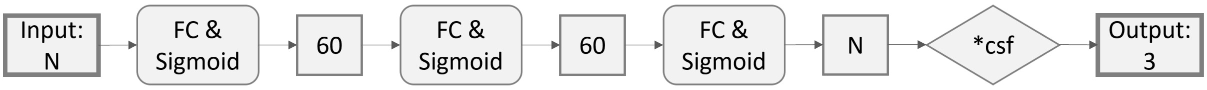


Figure 2.4: The Feed Forward Neural Network architecture is composed of three hidden layers. The rectangles with round edges indicate the layers used, in this specific case "FC" stands for fully connected layers, which are then followed by a sigmoid activation function. The network re-elaborates the given estimation to better fit the expected illuminant.

Average Multiplicative Weight (AMW) Let $IE_{RAW} \in \mathbb{R}^3$ be the illuminant estimation in raw RGB for a single image, and $IE_{MS} \in \mathbb{R}^N$ be the illuminant estimation in the multispectral domain. Let $GT_{\{RAW,MS\}}$ be the corresponding ground truth information.

The relationship between estimated illuminant (IE) and ground truth illuminant (GT) can be expressed by means of a multiplicative weight (W)

$$\begin{aligned} IE_{MS} * W_{MS} = GT_{MS}, \\ IE_{RAW} * W_{RAW} = GT_{RAW} \end{aligned} \tag{2.3}$$

From this relationship, given an image, the W factor can be defined as the division between the ground truth and the illuminant estimation

$$\begin{aligned} W_{MS} = \frac{GT_{MS}}{IE_{MS}}, \\ W_{RAW} = \frac{GT_{RAW}}{IE_{RAW}} \end{aligned} \tag{2.4}$$

Given a training set having cardinality C, a $C \times N$ multispectral weights matrix is obtained and $C \times 3$ raw RGB weights, and subsequently they have been averaged along the cardinality dimension. This process is replicated for each one of the six selected color constancy algorithms, estimating in total $6 \times N$ multispectral weights and 6×3 raw RGB weights.

Average Additive Bias (AAB) The Average Additive Bias method is based on a similar idea to the one of the Average Multiplicative Weight. In this case, instead of a multiplicative weight, the re-elaboration has been modeled through an additive bias (B). The relationship between estimated and ground truth illuminant is expressed as

$$\begin{aligned} IE_{MS} + B_{MS} &= GT_{MS}, \\ IE_{RAW} + B_{RAW} &= GT_{RAW} \end{aligned} \tag{2.5}$$

The bias B is calculated by simply subtracting the illuminant estimation from the ground truth

$$\begin{aligned} B_{RAW} &= GT_{RAW} - IE_{RAW}, \\ B_{MS} &= GT_{MS} - IE_{MS} \end{aligned} \tag{2.6}$$

As for the previous approach, given a training set of cardinality C, the $C \times N$ multispectral and the $C \times 3$ raw RGB biases matrices are averaged, and the process is repeated for each algorithm, resulting again in $6 \times N$ multispectral biases and 6×3 raw RGB biases. Each bias is then applied to the testing set estimations of the corresponding algorithm coherently with the relationship expressed in Equation 2.6.

Optimization-Driven Multiplicative Weight (ODMW) The search of the weight W from Equation 2.3 is here carried out through a direct search method for multidimensional unconstrained minimization [64]. This is obtained by optimizing the end result in terms of raw RGB recovery angular error, between the expected RGB illuminant and the estimated illuminant after the application of the weights

$$\begin{aligned} W_{MS} &= \text{argmin}_w \{ e_{rec}(GT_{RAW}, csf \times (IE_{MS} * w)) \}, \\ W_{RAW} &= \text{argmin}_w \{ e_{rec}(GT_{RAW}, IE_{RAW} * w \} \end{aligned} \tag{2.7}$$

where e_{rec} is the recovery error 1.2, csf is the camera sensitivity function, w denotes a possible set of weights and × is the matrix multiplication. The direct search method for multidimensional unconstrained minimization method is known as the Nelder–Mead simplex algorithm [64], for which the MATLAB implementation known as “fminsearch” has been utilized. All weights are initialized as ones, indicating no re-elaboration.

Feed Forward Neural Network (FFNN) The approach selected for the fourth multispectral illuminant re-elaboration method is learning-based. A topologically simple neural network has been designed, based on a feed-forward multilayer perceptron [44], consisting of only three fully connected layers with a sigmoid activation function, as shown in Figure 2.4. The first fully-connected layer maps the N-band multispectral input to a 60-dimensional latent space, while the last layer maps the result back to N values. Assuming that the input consists of 31-band multispectral data, the dimension of the latent space has been heuristically defined. The multispectral output of the third fully connected layer is converted into raw RGB exploiting the camera sensitivity function. The conversion to raw RGB makes it possible to use, as the loss, function the recovery error between the raw-converted illuminant estimation and the expected raw illuminant, as defined in Equation 1.2.

For the sake of comparison, the same re-elaboration method is also applied to the estimated raw RGB illuminants. In this case, the input dimension N is 3, the dimensionality of the latent space has not been modified, instead, it has been kept to 60. The output of the last fully-connected layer is already in raw RGB format and can be directly used in the computation of the loss function.

2.2.2 Experimental setup

2.2.2.1 Dataset

The focus of our investigation resides in the use of multispectral imaging to improve raw RGB illuminant estimation, therefore it has been selected the NUS [83] dataset by Nguyen et al. from the National University of Singapore, which contains multispectral data along with the ground-truth of their radiance information and the camera sensitivity functions. Having this information it is possible to compute raw RGB data by multiplying the multispectral data by the camera sensitivity functions. The same process of raw RGB computation is also applied to the ground truth multispectral illuminant data. The dataset contains 64 multispectral images along with the corresponding illuminant spectra, that have been acquired using Specim's PFD-CL-65-V10E (400 nm to 1000 nm) spectral camera with Specim OLE23 fore lens. For light sources, the dataset varies from natural sunlight to shade conditions, additionally considering artificial wide-band lights obtained from metal halide lamps of different color temperatures (2500 K, 3000 K, 3500 K, 4300 K, 6500 K) and a commercial off-the-shelf LED E400. The subjects of the scenes include both outdoor and indoor images and both natural and man-made objects. Furthermore, a few images of buildings at very long focal lengths were also included. For each spectral image, a total of 31 bands were considered at 400 nm to 700 nm, with a spacing of about 10 nm. Of the total 64 multispectral images in the dataset, 24 are reserved for testing and the remaining 40 for training. One or multiple color targets are present in the acquired scenes.

In order to reduce computational complexity, the dataset has been processed by rescaling the multispectral images to a tenth of their original size, resulting in 24 testing images of $132 \times W \times 31$, and in 40 training images of $132 \times W \times 31$, where W is the width size that varies between 95 and 237.

2.2.3 Experimental Results

This section is divided into four parts. The first part is dedicated to the analysis of the performance of multispectral extension color constancy algorithms as defined in Section 2.2.1.1, compared to the performance of their raw RGB counterpart. The second part is reserved for the assessment of the four re-elaboration techniques extensively discussed in Section 2.2.1.2. In order to return a clear idea of how the estimations resulting from these methods and pipelines perform in the color correction step visual examples for a visual comparison are shown. The third part is dedicated to the comparison of the first-section results with two of the main works from the state of the art, namely those from Khan et al. [59] and Robles-Kelly et al. [89]. In the fourth and final part, the contribution of each input spectral band in our best solution for multispectral-based illuminant estimation is measured, so as to provide a form of model explainability.

To compare the aforementioned methods,the reproduction angular error [32] is selected. The rationale is to evaluate the final effect of applying color constancy in the RGB domain, as measured via reproduction error, in addition to the intermediate step of illuminant estimation, as measured via recovery error.

2.2.3.1 Raw RGB vs Multispectral Illuminant Estimation

This section is dedicated to the assessment of the performance of multispectral extended color constancy algorithms, compared with the original raw RGB edge-based color constancy algorithms. The results are reported in Table 2.1. It has been observed that only three multispectral extended algorithms perform better than their RGB version spectral white point, spectral first order grey edge and spectral second order grey edge. While spectral white point mean reproduction error improves only by 0.08° compared to its raw RGB version (corresponding to a 1% improvement), spectral first order grey edge and spectral second order grey edge improve respectively by 0.38° (6%) and 0.65° (9%). The results for the grey world algorithm are the same for both the multispectral and the raw RGB input. Additional statistics are reported in the supplementary materials.

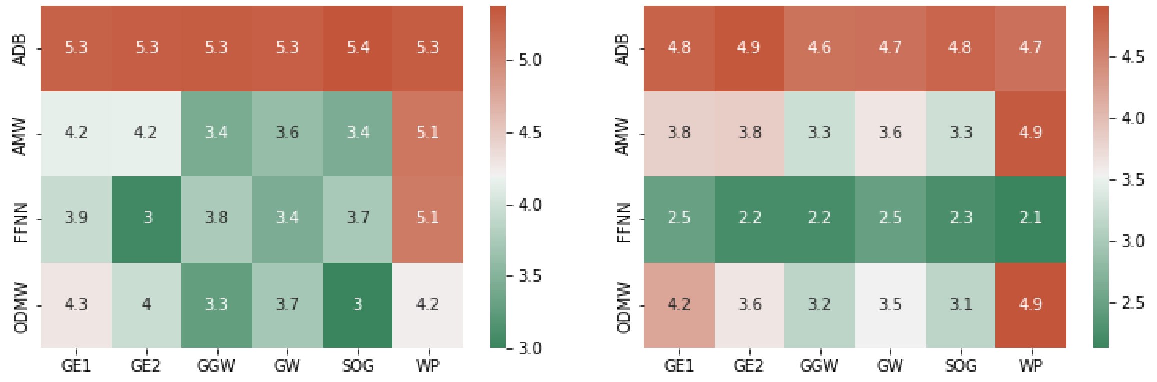
2.2.3.2 Iluminant Estimations Re-elaboration

The illuminant estimation re-elaboration will be assessed method by method, comparing the performance of the re-elaborated raw RGB input and the performance of the re-elaborated multispectral information. As for the previous assessment, the estimations are compared with the reproduction error, and the performance is reported in terms of mean and median errors. Additionally, it shows the percent improvement with respect to the

input	awb algorithm	mean recovery	median recovery	mean reproduction	median reproduction	% mean reproduction improvement
multispectral	**GE1**	5.09	4.42	6.38	5.23	6%
raw RGB		5.46	4.72	6.76	5.67	
multispectral	**GE2**	5.03	4.49	6.32	5.13	**9%**
raw RGB		5.55	5.26	6.97	6.29	
multispectral	**GGW**	3.70	3.15	**4.42**	3.37	-2%
raw RGB		**3.67**	2.92	4.33	3.09	
multispectral	**GW**	3.81	**2.55**	**4.42**	**2.91**	0%
raw RGB		3.81	**2.55**	**4.42**	**2.91**	
multispectral	**SOG**	3.84	3.15	4.63	3.74	-1%
raw RGB		3.84	3.16	4.57	3.71	
multispectral	**WP**	4.68	3.93	5.60	4.99	1%
raw RGB		4.81	4.29	5.68	5.26	

Table 2.1: Evaluation of raw RGB and multispectral color constancy algorithms performance divided by method. Both recovery and reproduction angular errors have been reported, expressed in degrees (°). The lower the better. The last column shows the percentual improvement in mean reproduction angular error of multispectral vs. raw RGB algorithms. Best results per metric are highlighted in bold.

traditional raw RGB pipeline from Table 2.1. The results for all re-elaboration methods are presented in Table 2.2 and in Figure 2.5 for a clearer view.



(a) raw input

(b) multispectral input

Figure 2.5: Visual representation of Table 2.2. We show the mean recovery angular error value in the form of a heatmap, one for the raw RGB input and one for the multispectral input.

Average Multiplicative Weight All illuminant estimation algorithms benefit from the use of the Average Multiplicative Weight method, both for the multispectral and raw RGB input. Results in Table 2.2 also show that not only the re-elaboration methods improve the illuminant estimation accuracy but also that the multispectral input improves the performance with respect to the raw RGB input. The best improvement is achieved by

input	relaboration method	awb algorithm	mean recovery	median recovery	mean reproduction	median reproduction	% improvement mean reproduction error
raw RGB	AAB	GE1	5.28	4.07	6.46	4.75	4%
raw RGB	AAB	GE2	5.31	4.20	6.49	4.91	7%
raw RGB	AAB	GGW	5.32	4.39	6.50	5.25	-50%
raw RGB	AAB	GW	5.30	4.51	6.48	5.50	-47%
raw RGB	AAB	SOG	5.38	4.24	6.58	5.08	-44%
raw RGB	AAB	WP	5.32	4.02	6.51	4.82	-15%
multispectral	AAB	GE1	4.78	3.63	5.91	4.27	13%
multispectral	AAB	GE2	4.88	3.93	6.01	4.57	14%
multispectral	AAB	GGW	4.64	3.60	5.75	4.39	-33%
multispectral	AAB	GW	4.66	3.73	5.77	4.52	-31%
multispectral	AAB	SOG	4.76	3.58	5.89	4.46	-29%
multispectral	AAB	WP	4.66	3.35	5.78	4.10	-2%
raw RGB	AMW	GE1	4.15	3.36	4.79	4.42	29%
raw RGB	AMW	GE2	4.16	3.98	4.84	4.30	31%
raw RGB	AMW	GGW	3.42	2.21	3.82	2.55	12%
raw RGB	AMW	GW	3.61	3.08	3.95	3.31	11%
raw RGB	AMW	SOG	3.44	2.28	3.84	2.54	16%
raw RGB	AMW	WP	5.12	4.79	5.57	5.24	2%
multispectral	AMW	GE1	3.82	2.88	4.51	3.70	33%
multispectral	AMW	GE2	3.84	3.30	4.54	3.82	35%
multispectral	AMW	GGW	3.26	2.23	3.69	2.34	15%
multispectral	AMW	GW	3.62	2.86	4.03	3.28	9%
multispectral	AMW	SOG	3.28	2.22	3.73	2.60	18%
multispectral	AMW	WP	4.86	4.44	5.33	5.00	6%
raw RGB	ODMW	GE1	4.29	3.46	5.25	4.36	22%
raw RGB	ODMW	GE2	3.95	3.54	4.80	4.16	31%
raw RGB	ODMW	GGW	3.27	2.87	3.81	3.45	12%
raw RGB	ODMW	GW	3.71	3.48	4.36	4.08	1%
raw RGB	ODMW	SOG	3.00	1.94	3.50	2.40	23%
raw RGB	ODMW	WP	4.22	3.97	4.84	4.46	15%
multispectral	ODMW	GE1	4.20	3.41	5.15	4.07	24%
multispectral	ODMW	GE2	3.62	3.04	4.35	3.61	38%
multispectral	ODMW	GGW	3.15	2.25	3.59	2.31	17%
multispectral	ODMW	GW	3.52	2.85	3.99	3.17	10%
multispectral	ODMW	SOG	3.14	2.18	3.65	2.58	20%
multispectral	ODMW	WP	4.91	4.01	5.61	4.86	1%
raw RGB	FFNN	GE1	3.93	3.76	5.13	5.40	24%
raw RGB	FFNN	GE2	3.05	2.16	3.81	3.19	45%
raw RGB	FFNN	GGW	3.75	3.69	4.48	3.76	-3%
raw RGB	FFNN	GW	3.44	2.40	4.02	2.68	9%
raw RGB	FFNN	SOG	3.74	2.79	4.62	3.86	-1%
raw RGB	FFNN	WP	5.09	2.75	6.20	3.19	-9%
multispectral	FFNN	GE1	2.48	2.18	3.15	2.63	53%
multispectral	FFNN	GE2	2.24	1.33	2.77	1.93	**60%**
multispectral	FFNN	GGW	2.24	1.43	2.71	1.55	37%
multispectral	FFNN	GW	2.47	1.53	2.91	1.63	34%
multispectral	FFNN	SOG	2.26	1.11	2.71	1.43	41%
multispectral	FFNN	WP	**2.13**	**1.04**	**2.58**	**1.28**	55%

Table 2.2: Evaluation of the re-elaboration of the multispectral and raw RGB illuminant estimations. All values are expressed in degrees (°), the lower the better. The last column shows the percentage of improvement of the mean reproduction angular error, between the selected method and the traditional illuminant estimation algorithm for the raw RGB input.

multispectral second order grey edge (GE2) but the best performance overall is achieved by multispectral general grey world (GGW) with a 3.69° reproduction error.

Average Additive Bias The only illuminant estimation algorithms that benefit from the Average Additive Bias re-elaboration are first order grey edge (GE1) and second order grey edge (GE2), even in their multispectral extension.

However, with this re-elaboration, the best performing pipeline is the multispectral general grey world (GGW) with a 5.75° mean reproduction error value, which is still performing worse than the 3.69° achieved by the raw RGB general grey world (which is the best traditional performing algorithm in this investigation).

Optimization-Driven Multiplicative Weight The re-elaboration of the multispectral estimations obtained with the Optimization-Driven Multiplicative Weight improves the mean reproduction error value for each algorithm. However, the raw RGB shades of grey (SoG) estimations actually achieve the best result for the Optimization-Driven Multiplicative Weight method with a mean reproduction error of 3.5°, which improves the value for that metric with respect to the non-re-elaborated raw RGB estimations by 23%. While the best improvement is achieved by multispectral second order grey edge (GE2) with a 38% improvement compared to the traditional raw RGB estimation.

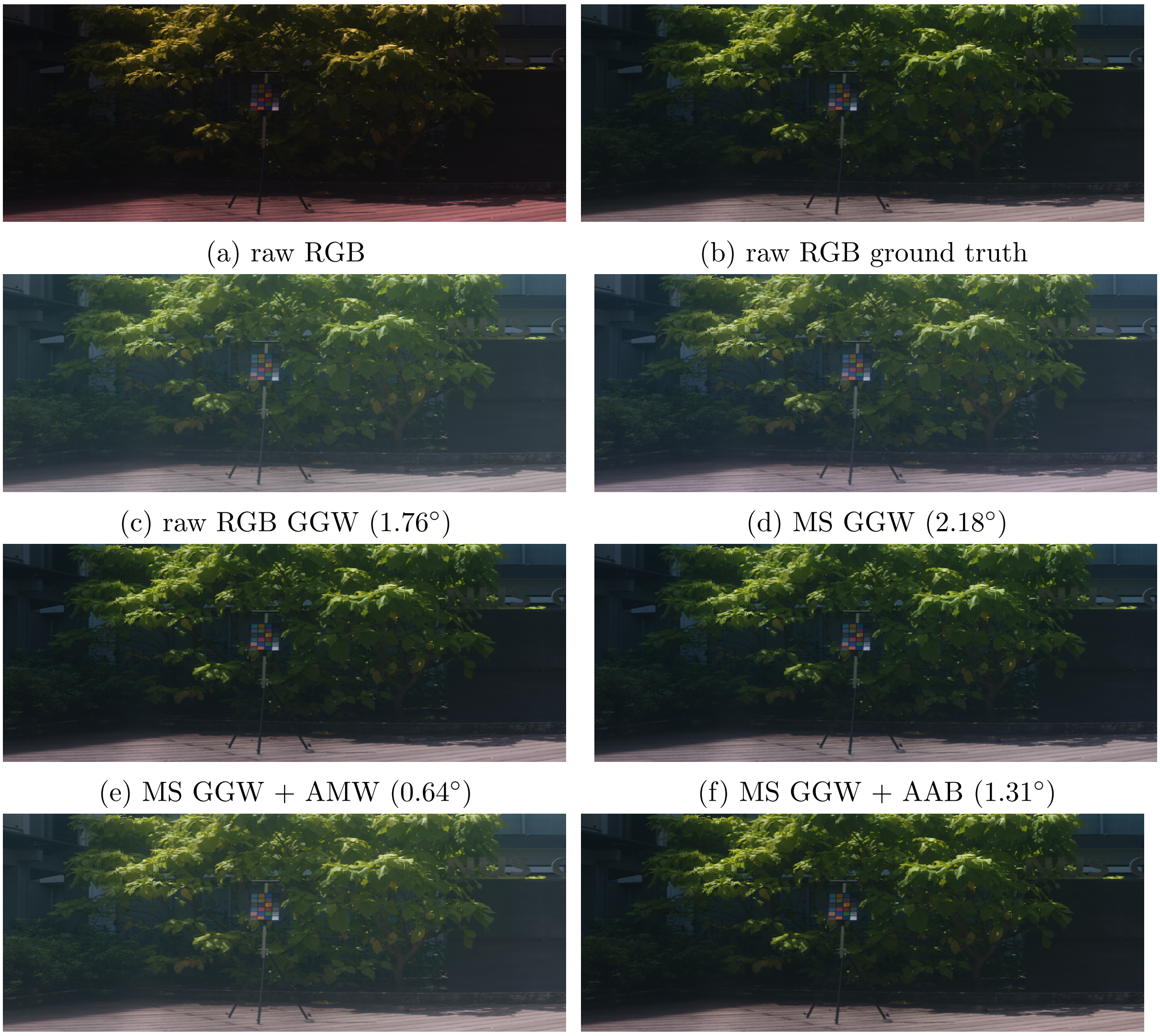
Feed-Forward Neural Network The performance of the Feed-Forward re-elaboration method led to more noticeable improvements in processing multispectral information with respect to traditional raw RGB data, as can be easily appreciated from Table 2.2. The mean reproduction error for the multispectral inputs ranges from 3.15° for the first order grey edge (GE1) down to 2.58° for the spectral white point (WP), achieving the best performance overall in our analysis. The white point (WP) re-elaboration method improves by 55% compared with the traditional raw RGB method for the same illuminant estimation algorithm.

Figure 2.6 offers a visual representation of the effect of color constancy using our proposed method in different configurations, including raw RGB or multispectral input, and the four re-elaboration methods.

2.2.3.3 Comparison with the State of the Art

Among the methods in the state of the art, Robles-Kelly et al. [89] and Khan et al. [59] are the most similar to the hereby proposed method in terms of approach and final goal, i.e. RGB color constancy by exploiting multispectral information.

Robles-Kelly presented a method that employs a convolutional neural network to estimate pixel-wise illuminant in the scene for both trichromatic and spectral images. Khan et al. [59] proposed to extend statistical illuminant estimation methods (applied also here, and described in Section 2.2.1.1) to N dimensions, and subsequently developed a spectral adaptation transform to bring the multispectral image data into canonical, or target, multispectral representation [61]. In order to enable a direct comparison with our method it has been necessary to apply a consensus-based strategy for raw RGB illuminant



(a) raw RGB

(b) raw RGB ground truth

(c) raw RGB GGW (1.76°)

(d) MS GGW (2.18°)

(e) MS GGW + AMW (0.64°)

(f) MS GGW + AAB (1.31°)

(g) MS GGW + ODMW (0.25°)

(h) MS GGW + FFNN (0.19°)

Figure 2.6: Visual example of some of the most significant methods. (a) acquired scene; (b) scene corrected with the expected illuminant in raw RGB; (c-h) scene corrected with the illuminant estimated via general grey world (GGW) on either raw RGB or multispectral (MS) input, using different re-elaboration methods. For all corrections, angular errors are reported in parentheses (the lower, the better).

estimation. The strategy consists in

1. converting the input multispectral radiance image into the RGB domain, to obtain a raw RGB image that is not white-balanced.

2. dividing the same input multispectral radiance image by the estimated multispectral illuminant to obtain a multispectral reflectance image.

3. multiplying the obtained multispectral reflectance image for the target multispectral illuminant, to obtain a new multispectral radiance image.

Method	**mean recovery angular error**	**median recovery angular error**
Robles-Kelly et al. [89]	12.56	4.62
Khan et al. (multispectral GGW) [61]	3.96	2.94
Our baseline (raw GGW)	3.67	2.92
Our baseline (multispectral GGW)	3.70	3.15
Our best method (multispectral WP + FFNN)	**2.13**	**1.04**

Table 2.3: Mean and median recovery angular error (in degrees °, the lower the better) for Khan's, Robles-Kelly's and our method on NUS dataset [83]. In bold the best results.

4. converting the multispectral image resulting from step 3 into the RGB domain, to obtain a raw RGB image that is white-balanced.

5. obtaining a per-pixel RGB illuminant estimation by dividing the result of point 4 by the result of point 1.

6. generating a global raw RGB illuminant estimation of the input image by consensus through average per channel.

In Table 2.3 a comparison of the results of the previously cited methods is shown against our solution. Two baselines of the presented method without re-elaboration (based on raw RGB and multispectral data) have been considered, using general grey world as a reference due to its optimal performance as reported in Table 2.1. Our best-performing configuration has also been taken into consideration, using multispectral white point and FFNN re-elaboration as reported in Table 2.2. The comparison is performed in terms of mean and median recovery angular error in order to allow for a direct comparison with the results reported by Robles-Kelly et al. [89].

The comparison between the proposed method's baselines and existing methods from the state of the art highlights that a simple approach without re-elaboration achieves similar performance as the solution by Khan et al. Additionally, our complete method based on output re-elaboration allows for achieving superior performance in raw RGB illuminant estimation.

2.2.3.4 Further analysis

In order to further study the best model for illuminant estimation from multispectral data, an investigation regarding the relationship between input and output wavelength bands has been conducted as a form of model explainability. Specifically, the relevance of each input spectral band i has been measured by selectively feeding to the trained feed forward neural network FF a set of band-specific impulses (setting band i to 1, and all the other bands to 0). The assessment consists in the absolute difference in each of the network's

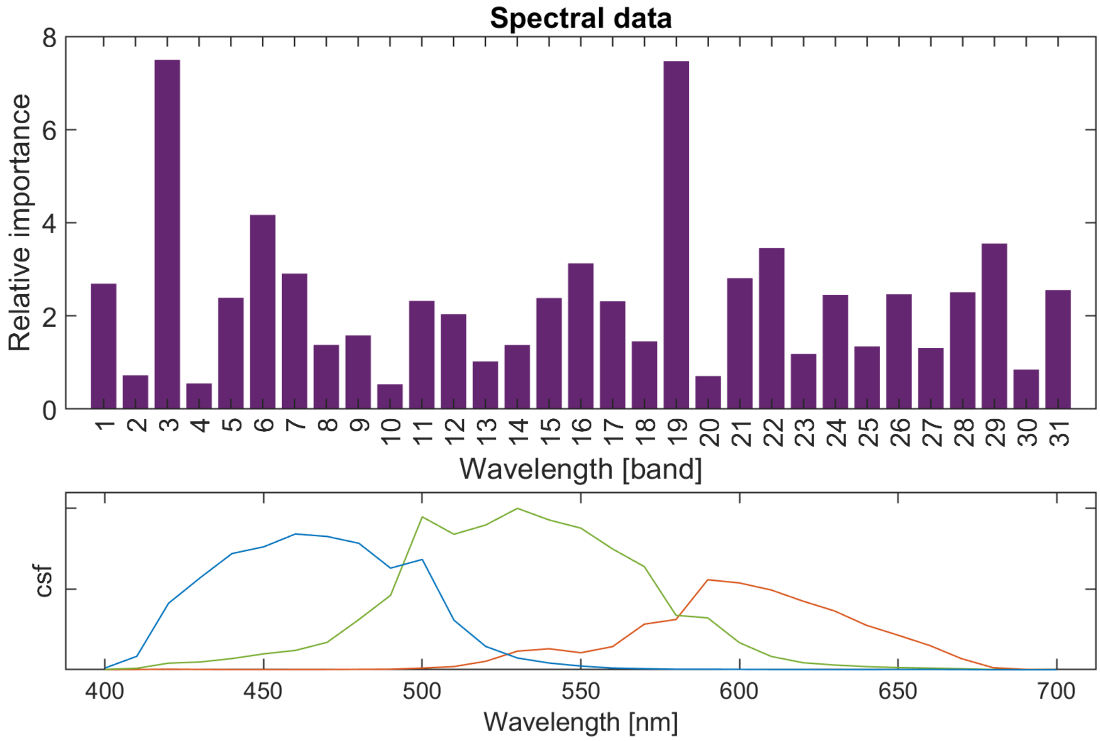


Figure 2.7: Relative importance of wavelength bands for our neural model for illuminant estimation from multispectral data. Camera sensitivity functions reported for reference.

output bands j compared to the average network's output A

$$rel_i = \sum_j \left| FF\left(impulse(i)\right)_j - A_j \right| \tag{2.8}$$

$$A_j = \frac{1}{N} \sum_i FF\left(impulse(i)\right)_j \tag{2.9}$$

The result of this analysis is reported in Figure 2.7. By supporting the visualization with the camera sensitivity function curves used in the optimization process, three band clusters emerge, roughly corresponding to the central sections of the cameras color filters, with local minima corresponding to the overlap between two color channels, where information is partially redundant (the estimate on the green channel is partially informed by the information from the blue and red channels). Furthermore, the model presents two outlying peaks and a generally oscillating behavior.

These observations on band relevance could potentially inform the definition of feature reduction techniques for hardware optimization, where fewer and selected wavelength bands are considered in the construction of a multispectral sensor.

2.2.4 Conclusions

An investigation to assess whether multispectral information can be beneficial for the raw RGB color constancy problem has been conducted. The work can be separated into two main steps. 1) In the first step it is shown an evaluation of the multispectral illuminant estimations and a comparison of the results with the traditional raw RGB illuminant estimations. 2) The second step suggested to re-elaborate multispectral estimations to better fit the expected raw RGB illuminant. Four re-elaboration methods have been proposed and evaluated, not only by comparison to the traditional raw RGB approach but also with raw RGB re-elaborated performance.

It has been proved that multispectral information can be used to improve raw RGB color constancy. In fact, it has been shown that some methods (first-order grey edge, second-order grey edge and white point) improve with our multispectral-based methods. Results show that re-elaboration methods improve performance both for multispectral and raw RGB illuminant estimation respectively with an overall performance increment of 60% and 50%, for the mean reproduction angular error, with respect to the traditional raw RGB pipeline. Of great relevance is the result achieved by the multispectral white point with feed-forward neural network re-elaboration, which achieves a mean recovery error of 2.13°.

Future development may involve the extension of the work to other illuminant estimation algorithms, especially machine learning-based algorithms.

2.3 Illuminant estimation exploiting spectral average radiance

This section is dedicated to investigating whether combining spectral and color information can improve traditional color constancy [30].

Nowadays smartphones may embed spectral sensors that are able to capture the spectral average radiance of the scene. A recent patent from Apple [55], for example, describes an electronic device that includes control circuitry that gathers ambient light measurements using a color ambient light sensor. Sensor responses are processed to generate a color rendering index for the ambient light, which is used to correct the color of the captured images via a color correction matrix. This leads to more accurate and faithful color reproduction in the captured images. Hybrid-Resolution Spectral Imaging Systems have also been proposed [77, 78, 79], where a conventional high-resolution RGB color camera is combined with a low-resolution spectral imaging sensor, producing an high-resolution spectral image. The focus of this work is to investigate how low-resolution spectral radiance can be combined with high-resolution RGB color information to produce a properly white-balanced RGB color image.

Unlike previous works, that only consider either the RGB or the spectral information, this work proposes to combine them both to improve the accuracy of illuminant estimation. This chapter shows that, by incorporating both the RGB and spectral domains, it is possible to capture a more comprehensive set of features related to the illuminant, which improves the accuracy of the estimation. In particular, this work poses itself as an investigation divided into three points 1) the first point concerns which resolution of color and spectral information brings the higher benefit, 2) the second point regards whether it is more beneficial to predict the illuminant in the spectral or color domain, and finally, 3) The third point focuses on whether to use color or spectral domain for illuminant target during training.

The chapter is structured as follows: Section 2.3.1 describes our proposed method for illuminant estimation exploiting both RGB and spectral average radiance. Section 2.3.2 presents the experimental setup and results.

2.3.1 Method

The work presented in this chapter poses itself with the purpose of providing a color illuminant estimation method that combines RGB color and spectral information. Assuming the availability of an RGB color image and the spectral average distribution of its corresponding radiance scene, the combination is performed by means of a suitably-designed neural network. According to this method, the input image has been divided into patches, and the designed neural network has been trained having as input the RGB color image patch, and its corresponding spectral average distribution. The size of the patch may vary, and its tuning is discussed in the experimental results section. The process for each single patch is visually depicted in Figure 2.8, which illustrates in parallel the process of RGB color illuminant estimation and spectral illuminant estimation. These two options will be compared in the experimental results section (2.3.2). Given an input image, the individual patch estimations are combined with a suitable selection module, as described in Section 2.3.1.1.

The neural network architecture, depicted in Figure 2.9 is composed of two branches. The first branch takes as input the RGB color patch having size $w \times h \times 3$, and a suitably designed Convolutional Neural Network (CNN) extracts a feature vector of size N, where N is the same as the spectral resolution. The second branch takes as input an N-dimensional vector of the spectral average distribution, and a Feed-Forward Neural Network (FFNN) extracts a feature vector of the same size N. The two vectors are then concatenated into a vector of size $2N$, which is fed to the final block of the neural network, which differs in structure and in terms of the final output

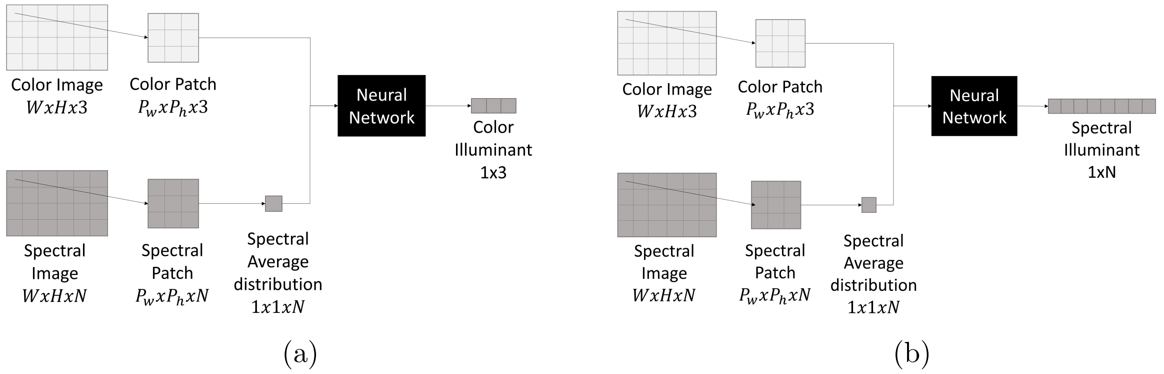


Figure 2.8: The RGB color and N-dimensional spectral images are divided into patches of $P_w \times P_h$ pixels. The RGB color patch and the spectral average distribution are fed to the neural network. In figure (a) the network returns an RGB color illuminant estimation while in figure (b) it returns a spectral illuminant estimation.

- an encoder for the case of RGB color illuminant estimation produces as output a three-dimensional vector corresponding to the RGB coordinates of the illuminant;
- a Feed-Forward Neural Network (FFNN) for the case of spectral illuminant estimation produces an N-dimensional vector corresponding to the spectrum of the illuminant.

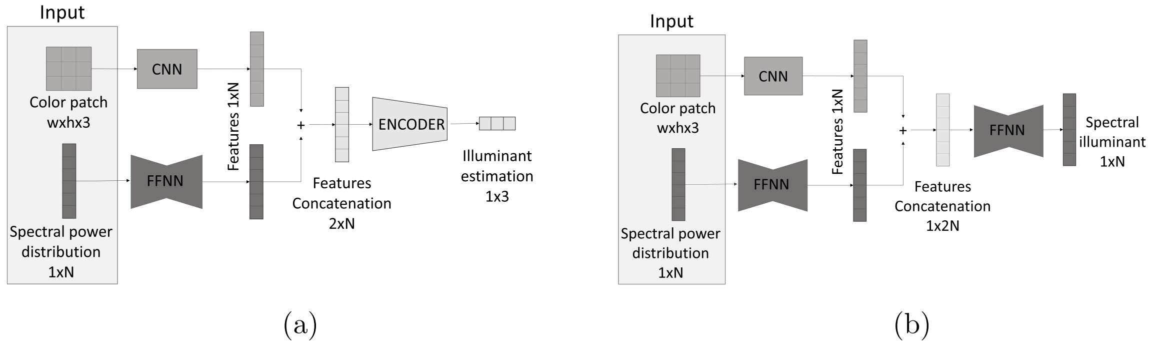


Figure 2.9: Figure (a) represents the spectral architecture while figure (b) represents the color one. The two architectures are identical except for the last block. The network gets a color patch as input (of dimension $h \times w \times 3$) and its spectral average distribution. The color patch is fed to the convolutional neural network which returns a $1 \times N$ feature map. In the same way, the spectral average distribution is fed to a feed-forward neural network that elaborates it and returns a $1 \times N$ feature map. The two feature maps get concatenated into a unique feature map. In Figure A the resulting feature map gets fed to another feed-forward neural network which finally returns a $1 \times N$ spectral illuminant estimation. In Figure B, instead, the resulting future map gets fed to an encoder, which returns the color illuminant estimation.

The choice of the convolutional part of the network architecture takes into consideration the scarce availability in the state of the art of spectral datasets that provide illuminant targets of spectral radiance images. Due to this circumstance, the choice fell on a shallow convolutional neural network architecture with a small number of trainable parameters.

More precisely, the selected architecture is the Convolutional Mean architecture [43] which consists of two convolutional layers (the first being $3 \times 3 \times 3 \times 7$ and the second one $3 \times 3 \times 7 \times 14$, both of them having stride and padding set to 1) each followed by a Max Pooling layer (2×2) and the activation function, next the Weighted Global Average Pooling Layer which in turn is composed by the third convolutional layer ($1 \times 1 \times 14 \times 3$ with stride and padding set to 1) followed by a ReLu and a Per-Channel Global Average Pooling. While the Per-Channel Global Average Pooling layer returns the feature map averaged by channel, in this work the feature map dimension is only reduced by half with an average pooling layer and then the resulting feature map is fed to a fully connected layer to return a vector equal in size to the number of channels of the spectral input. The ReLu layers have been replaced with leaky ReLu [105] which has proven to be capable of solving the "Dead Neuron" problem and is more effective than ReLu. The diagram of the CNN block is shown in Figure 2.10.

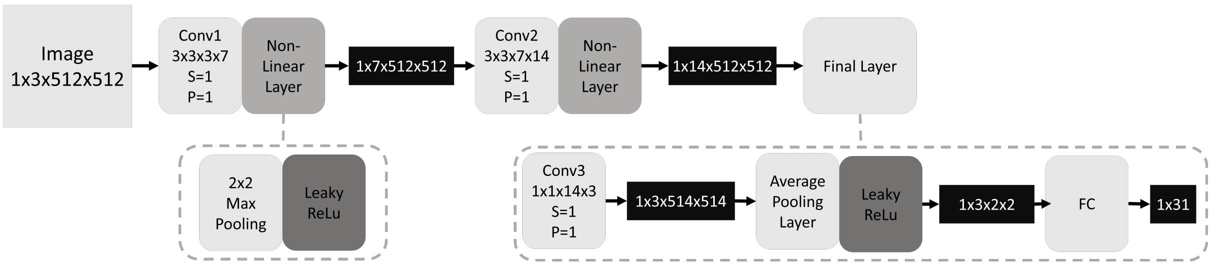


Figure 2.10: The Convolutional Neural Network block architecture contains two 3×3 filter convolutional layers (Conv1/2) which are followed by a $2{\times}2$ max pooling and a Leaky ReLU. In the end, there is a final layer which is implemented as a $1{\times}1$ convolutional layer (Conv3) with Leaky ReLU, an average pooling layer that halves the feature maps dimension and finally a fully connected layer (FC). In this diagram, P and S denote padding and stride respectively. The other four numbers shown in the Conv box represent "Filter Size 1 × Filter Size 2 × #Input Channel × #Output Channel" whose product is the total number of filter parameters. The activation functions are displayed assuming a patch size of 512.

Figure 2.11 shows the FFNN structures identified in dark grey in Figure 2.9. This consists of three fully connected layers followed by the leaky ReLu activation function. The first and second layer of the first structure maps the N-band spectral input (or feature map) to 60 values, while, the first and second layer of the second structure maps a 2N-band spectral vector to 60 values. Finally, the last layer in both structures maps them back to N values.

Finally, Figure 2.12 shows the encoder architecture that produces the RGB illuminant estimation for an input patch. It consists of three fully connected layers the first one maps the N input values to 15 values, the second one maps them to 5 and the final layer maps them to 3, obtaining, therefore, a color illuminant estimation. For the activation function, the leaky ReLu has been selected, for the same reasons previously explained.

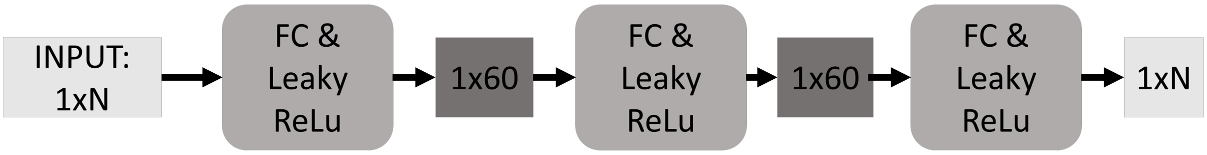


Figure 2.11: The Feed Forward Neural Network block architecture consists of three hidden layers. The rectangles with round edges indicate the layers used, in this specific case "FC" stands for fully connected layers, which are then followed by a Leaky ReLu activation function.

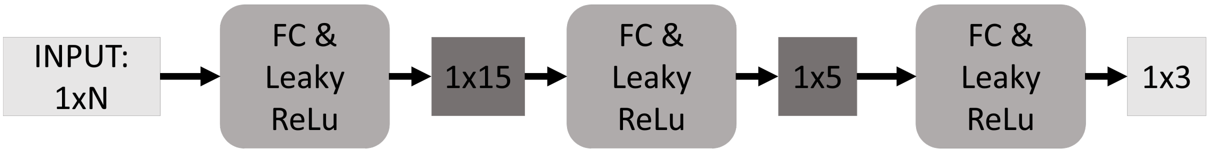


Figure 2.12: The encoder block architecture consists of three hidden layers. As for the figure 2.11, the rectangles with round edges indicate the fully connected layers, which are again followed by a Leaky ReLu activation function.

2.3.1.1 Selection Module

The model hereby proposed, provides several illuminant estimations corresponding to the patches of a single input image. Assuming that the illumination is uniform across the scene, the work proposed an exploitation of a further module in the inference phase with the purpose of selecting the best color estimation among the several color estimates corresponding to the analyzed patches. The intuition behind this module is that, among the suggested estimations, some are to be considered outliers, which must be eliminated, resulting in a subset of estimations supposedly closer to the actual illuminant in the scene. In order to implement this intuition, a k-mean clustering [106] has been exploited to assess a consensus among the estimations. The process is carried out in the inference phase, after the multispectral-RGB conversion step, therefore, the clustering is computed in the color domain. The number of clusters is automatically determined by computing the silhoutte coefficient [111]. The populousness of the clusters determines which are to be considered outliers and which one is to be taken into consideration to determine the most likely solution. Two alternative selection strategies have been proposed to determine such a solution the cluster centroid, and the individual patch estimation that is closest to the cluster centroid. These two strategies are compared in the experimental results section.

2.3.2 Experiments

This work focuses on the development of a neural network able to estimate the RGB color of the illuminant of a scene by combining spectral and RGB color information. To this end, the image is divided into patches, and for each patch, the average radiance has been used and the RGB data of the patch itself. In the inference phase, the patches color estimations are further processed to produce a single illuminant estimation.

The purpose of the following experiments is, not only to investigate whether the average spectral distribution of the scene can improve the RGB illuminant estimation but also to investigate the influence of the input patch size. The experiment section also revolves around another turning point whether predicting the illuminant in the RGB color or spectral domain is more valuable as input for RGB color correction. The second option, which consists of predicting a spectral illuminant, brings up two training strategies 1) training the spectral prediction with the spectral expected illuminant or 2) converting the prediction to RGB color and then training it with the RGB color expected illuminant (ground truth). This section will show and analyze the results obtained from these three ”implementation choices”. From now on they will be referred to as

1. RGB color architecture (CA).
2. spectral architecture trained on spectral (SATOS).
3. spectral architecture trained on RGB color (SATOC).

The performance of the model is evaluated through the recovery angular error metric [52] 1.2.

2.3.2.1 Data set

This work requires a dataset that contains both spectral and RGB color images in full resolution and the corresponding target illuminant in both representations. The dataset selected fulfills all our requirements and it captures real-world scenes. The NUS dataset [82] contains 64 spectral radiance images, of which 24 are reserved for testing and 42 for training. The images have dimensions of $1312 \times W \times 31$ pixels, where W varies from 951 to 2374. For each spectral image, a total of 31 bands were captured at 400 nm to 700 nm, with a spacing of 10 nm. The scenes subjects include outdoor and indoor images and natural and man-made objects. For illumination sources, the dataset varies from natural sunlight and shade conditions, additionally considering artificial wide-band lights obtained from metal halide lamps of different color temperatures (2500 K, 3000 K, 3500 K, 4300 K, 6500 K) and a commercial off-the-shelf LED E400. Furthermore, the dataset is provided with the spectral radiance of the scenes and the camera sensitivity function, which allows for spectral-to-color conversion. The target illuminant is retrieved from color-checker targets present in the spectral radiance images. The dataset is provided under the assumption that the illuminant is global over the entire scene.

According to the proposed method, the input RGB and spectral images are divided into patches. Given the unequal width dimensions of the dataset elements, to prove the effectiveness of our approach it has been decided to limit the analysis to a 512 × 512 central crop of the image. These images are further divided into patches of sizes 4 × 4, 8 × 8, 16 × 16, 32 × 32, 64 × 64, 128 × 128, 256 × 256 and finally 512 × 512. The RGB color patches and the corresponding average spectral radiance are used to train our model, as shown in Figure 2.9. The performance for the different patch sizes is discussed in the next sections.

2.3.2.2 Experimental Results

The investigation hereby conducted mainly focuses on the resolution of color and spectral information, and which combination of these two domains best benefits the color illuminant estimation problem.

2.3.2.3 Patch illuminant estimation

Given the 24 test images, we will assess the network's capability to accurately estimate the illuminant of the single image patches The estimated illuminants are compared with the target illuminants (i.e the illuminant associated with the image and the resulting recovery angular errors across all patches from all test images are synthesized in 2.4 with several statistics min, mean, median, percentile 95 percentile and max. As explained in the method section 2.3.1, this work proposes two different architectures (one that estimates the illuminant in the RGB color domain, and one in the spectral domain) and two training strategies (one provides a color target illuminant, while the other provides a spectral target illuminant). For the sake of simplicity, the training strategies have been summed up as 1) color architecture (trained on color target) (CA) 2) spectral architecture trained on spectral target (SATOS) and 3) spectral architecture trained on color target (SATOC).

Table 2.4 shows the mean recovery error values go from 1.67° to 2.9°, respectively achieved by the SATOC on patch size 64 and the SATOS for patch size 512. It is possible to notice from the metrics illustrated in Table 2.4, and even more evidently in Figure 2.13, that spectral and color resolution greatly impact the performance of the methods. The patch of 512×512, which has the highest color resolution and the lowest spectral resolution, performs the worst. At the same time, though, the best-achieving patch sizes are not the ones with the highest spectral resolution (from 4×4 to 16×16), but the mid-size ones (from 32×32 to 128×128). A clear comparison between the three proposed methods is shown in Figure 2.13. The figure shows that there is not a clear winner between CA and SATOC, while SATOS is the worst-performing method except for patch size 8x8 and patch size 128×128. CA, instead, performs best for patch sizes 8×8 and 512×512, while from patch size 64×64 to 256×256 the SATOC method performs the best, the performance for the remaining patch sizes is very similar.

method	patch size	min	mean	median	ple95	max
CA	4	0.03	1.92	0.68	9.37	26.85
	8	0.03	**1.71**	0.37	7.16	15.41
	16	0.03	1.93	**0.18**	7.52	16.96
	32	0.03	1.80	0.20	7.58	17.25
	64	0.03	1.81	0.55	6.72	15.91
	128	0.03	2.04	1.12	6.34	15.53
	256	0.06	1.98	1.09	6.36	15.96
	512	0.08	2.52	1.94	**4.90**	**13.86**
SATOS	4	0.03	2.34	0.32	9.44	39.72
	8	0.03	2.10	0.47	8.24	29.14
	16	0.03	2.29	**0.21**	12.16	17.26
	32	0.03	**1.93**	0.22	9.17	15.50
	64	0.03	1.97	0.31	10.20	15.52
	128	0.03	1.84	1.06	7.65	15.02
	256	0.03	2.30	1.65	**6.81**	16.74
	512	0.76	2.90	1.79	6.92	12.31
SATOC	4	0.03	1.90	**0.04**	10.07	45.25
	8	0.03	2.21	0.10	9.45	21.22
	16	0.03	1.91	0.13	8.20	17.39
	32	0.03	1.84	0.37	7.60	15.46
	64	0.03	**1.67**	0.73	6.56	14.07
	128	0.03	1.77	0.80	6.42	15.40
	256	0.03	1.80	0.83	7.09	16.26
	512	0.69	2.80	1.87	**6.40**	**11.80**

Table 2.4: Performance of the three methods measured in terms of recovery error in degrees (the lower the better). In bold it is shown the best results based on the method.

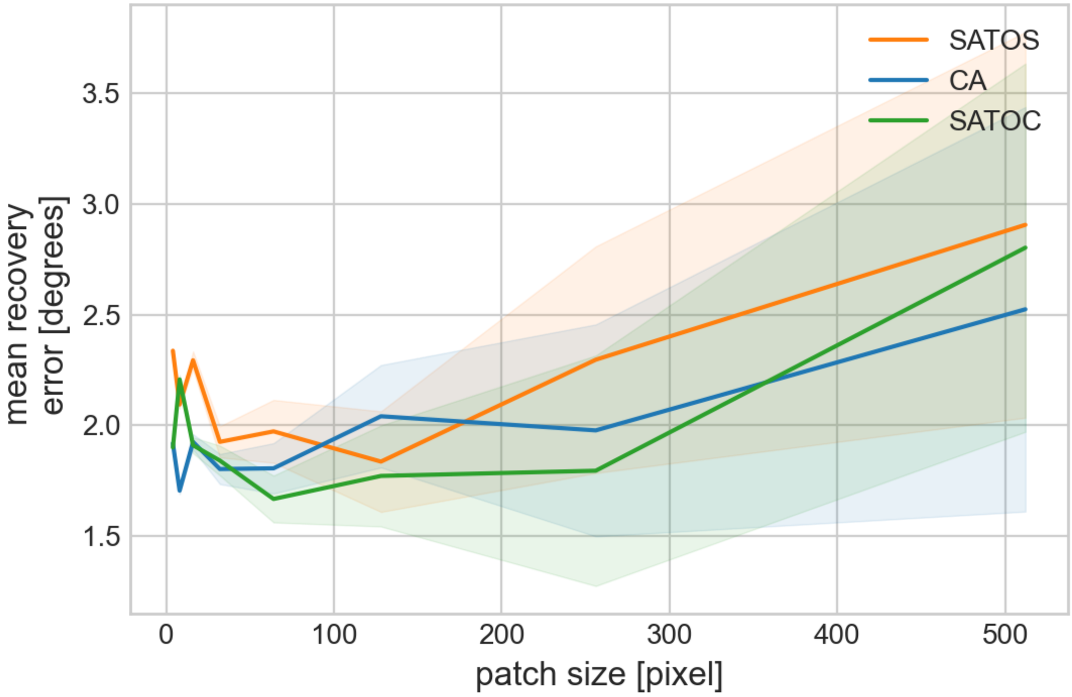


Figure 2.13: The plot shows the accuracy of the CA, SATOS, and SATOC methods, in terms of mean recovery angular error in degrees (°) varying the patch size. The lower the result the better.

2.3.2.4 Image illuminant estimation

According to the proposed method, there are several candidates for illuminant correction in a given image, corresponding to the image patches. For this reason, the method of an additional selection module has been provided, which has the purpose of identifying a single illuminant estimation among the ones estimated from the patches. As explained in section 2.3.2.4 this has been achieved by relying on a simple clustering technique.

method	patch size	cluster label accuracy	centroid recovery error	closest to centroid recovery error	closest to gt recovery error	best-selected distance
CA	256	0.64	1.82	1.82	1.37	0.45
	128	0.56	1.98	2.00	1.00	1.00
	64	0.76	1.57	1.58	0.43	1.15
	32	0.72	1.53	1.52	0.25	1.28
	16	0.64	1.76	1.75	0.14	1.61
	8	**0.88**	**1.33**	1.33	0.14	1.19
	4	0.80	1.58	1.58	**0.08**	1.50
SATOC	256	0.76	1.78	1.80	1.33	0.47
	128	0.56	1.83	1.83	0.72	1.11
	64	0.68	1.56	1.57	0.43	1.13
	32	0.72	1.75	1.74	0.27	1.48
	16	0.80	1.46	**1.46**	0.16	1.31
	8	0.76	1.90	1.90	0.13	1.78
	4	0.80	1.63	1.64	0.11	1.53
SATOS	256	0.44	2.28	2.22	1.95	0.27
	128	0.48	1.78	1.76	0.72	1.04
	64	0.60	1.62	1.63	0.77	0.86
	32	0.64	1.89	1.89	0.31	1.58
	16	0.64	2.31	2.31	0.21	2.09
	8	0.68	1.86	1.86	0.16	1.70
	4	0.64	2.04	2.05	0.15	1.89

Table 2.5: Selection module performance. The cluster label accuracy metric is considered in the range 0-1 where 1 is the best result. While centroid recovery error, closest to centroid recovery error, and closest to gt recovery error are expressed in degrees, the lower the better. The best-selected distance indicates the potential for improvement if it were possible to always select the best estimation, therefore, the lower the better. The best results based on the selection method are highlighted in bold.

In Table 2.5 the results of the proposed selection module are shown. The evaluation of the module concerns three different aspects. First, the two illuminant selection strategies (centroid, and estimation closest to the centroid) are evaluated in terms of mean recovery error. Second, the section presents an investigation regarding whether the most populated centroid is the one closest to the ground truth. Finally, the recovery error between the estimation closest to the ground truth (i.e. the best possible estimation) and the estimation closest to the centroid. Then, the two errors get subtracted as a way to show how much the problem could improve. The rationale behind this method is that the closer the two errors are, the less space for improvement there is, and vice versa. This metric helps understand the potential for improvement in the selection module. After analyzing the results and the graphs presented in 2.15, it becomes clear that the smaller patch sizes present more opportunities for improvement. Moreover, the metric indicated that the degree of improvement could range from 25% (for larger patch sizes) to as high as 95% (for smaller patch sizes).

The assumption behind the selection module is that the majority of estimations proposed by the network are close to the target illuminant. The clustering process serves as a voting mechanism, therefore the cluster containing the majority of the estimations also supposedly contains the estimation closest to the ground truth. Therefore, an accuracy metric has been included to investigate the veracity of this assumption. Table 2.5 shows that the selection cluster accuracy goes from 44% to 88%. The SATOS model is confirmed to have the worst performance. Overall the assumption is verified in more than 60% of the cases.

We then evaluate the performance of the two different proposed illuminant selections centroid and estimation closest to the centroid. As also shown in Figure 2.14, the performance of the two selection strategies is almost identical. From the graph, it is easy to see that the CA model overall has the best performance, except for patch sizes 4, 8, and 32. The best performance overall is achieved by the CA model for patch size with a recovery error of 1.33. The graph also confirms that the SATOS model is the worst-performing among the proposed ones.

The final metric adopted to evaluate this selection module is the distance between the selected estimation and the estimation closest to the ground truth. This metric stands for the improvement that the mean recovery error would have if it were always possible to select the best estimation. Overall it is possible to see that the improvement ranges from 0.5° to 2°. As expected, there is a tendency, where the difference increases as the patch size decreases, and therefore, as the number of estimations increases. Figure 2.15 shows the potential for improvement of the three models if the module were able to always select the best estimation.

Table 2.6 shows the comparison between our best-performing solution (SATOC on image patches of size 16×16 pixels) and Khan's and Robles-Kelly's works [59], [89]. Our method outperforms Khan's and Robles-Kellys' work, respectively, by 63% and 88% for the mean recovery angular error metric.

2.3.2.5 Considerations and observations

The hereby presented work shows that the patch size that leads to the best performance is patch size 8×8, and overall, the mid-size patches are the most suited for the problem, indicating that the problem benefits the most from mid-resolution both for color and spectral information. The network that provides estimations in the spectral domain and receives the target in the spectral domain (SATOS) is the worst-performing one. While

Method	mean recovery angular error	% improvement mean recovery error
Our Baseline ($SATOC_{16}$)	**1.46**	\
Roble's Kelly et. al. [89]	12.56	88%
Khan et. al. [61]	3.96	63%
Grey-Edge [100]	5.46	73%

Table 2.6: Mean recovery angular error (in degrees $\circ$, the lower the better) for Khan's, Robles-Kelly's and this method on NUS dataset [82]. For the method presented $SATOC_{16}$ has been selected, being the best-performing one. In bold the best results.

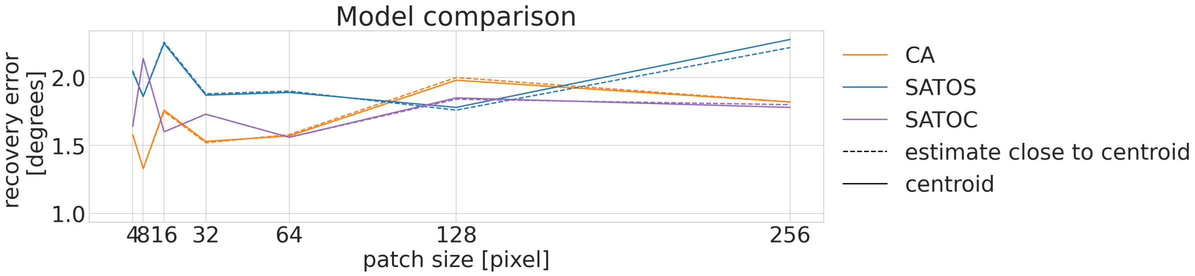


Figure 2.14: Comparison between the centroid and the estimation closest to the centroid performance. The comparison is performed in terms of recovery error in degrees (°), the lower the better.

there is no clear winner between the color approach (CA) and the spectral with color targets approach (SATOC). The CA approach performs better with smaller patch sizes and SATOC performs better for the larger patch sizes. It is also proven that the selection module performs better than the average of the result, meaning that it is able to extract an illuminant estimation closer to the target illuminant most of the time. It is also shown that the potential for improvement is very large, in fact, the error of the models with patch size 4 × 4 is close to zero. This result also proves how spectral information may be beneficial for the color illuminant estimation problem.

2.3.3 Conclusions

Spectral sensors are becoming every day cheaper and more available on the market, so much so that they're making their first appearances in digital imaging acquisition tools. This work poses itself as an investigation to verify if the combination of spectral and color information can improve the result for the RGB color constancy problem. The investigation has been performed on the standard NUS dataset. The best results, as identified in this experimental setup, are obtained with a model trained to predict the illuminant in the spectral domain using an RGB color loss function. We observed that, in general, by processing data in the form of mid-size image patches it is possible to achieve better results as compared to using the whole image or smaller image patches. In

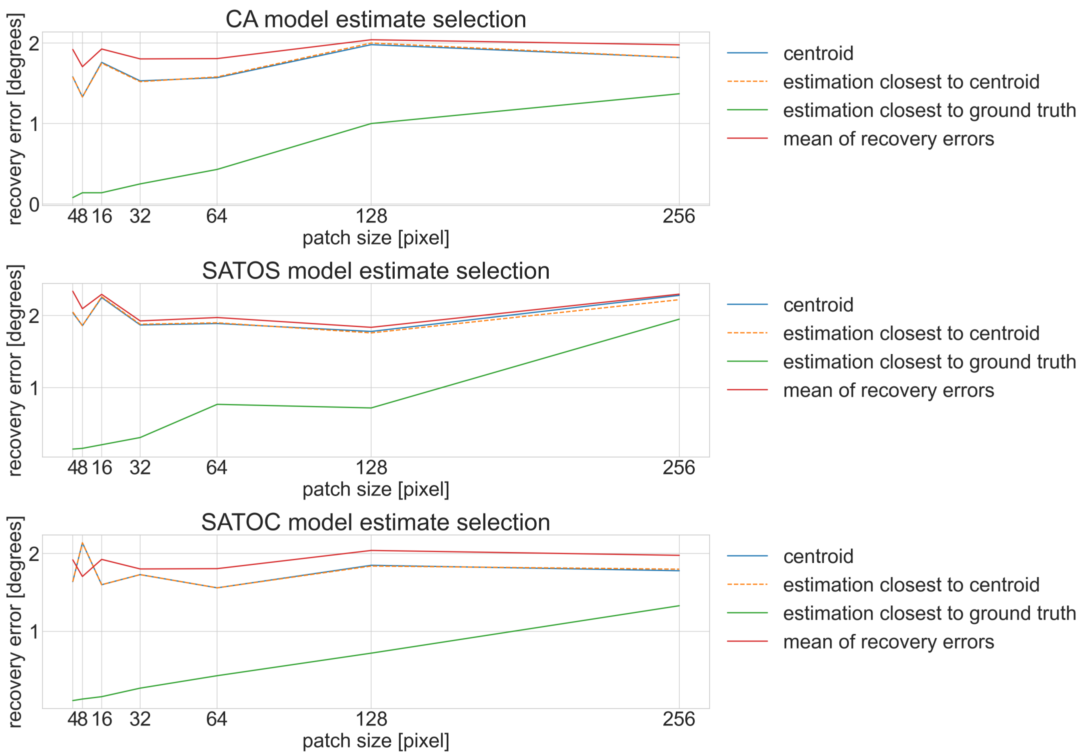


Figure 2.15: Comparison of performance for the three models (CA, SATOS, and SATOC) with centroid selection, estimation closest to the centroid, estimation closest to ground truth, and performance before the estimate selection. The comparison is performed in terms of recovery error (in °, the lower the better) based on the patch size.

the future, these results should be confirmed by extending the investigation to a larger properly annotated dataset. Nonetheless, our experiments show in practice the potential of combining spectral and RGB color information to improve RGB illuminant estimation.

Future developments may focus on neural network architectures that not only provide an illuminant estimation but also a level of confidence, as well as spatially varying multi-illuminant estimation.

Chapter 3

Temporal Color Constancy

As discussed in the previous sections, color constancy is a crucial aspect of digital image processing. While there are existing solutions available, they have limitations, and there is still room for improvement. The previous chapter focused on utilizing multispectral imaging to improve color constancy in single images. This chapter, however, is devoted to addressing color constancy in video content. Videos are essentially a sequence of individual images, which, from now on, will be referred to as ”frames”. The problem of correcting color distortion in frame sequences is known as Temporal Color Constancy and is a relatively new and challenging problem that has received little attention from the scientific community so far. The problem can be formulated as

$$\hat{c}_t = f(I_{t-(N-1)}, I_{t-(N-2)}...I_{t-1}, I_t) \tag{3.1}$$

where $f(\cdot)$ uses, besides the shot frame I_t to be corrected, the $(N-1)$ preceding frames $I_{t-1}...I_{t-(N-1)}$ [86].

While most existing algorithms insert citations of algorithms for color constancy were designed to address illumination changes in individual images, the straightforward approach to processing videos frame by frame using single-frame methods overlooks the temporal correlation of illumination changes between adjacent frames. This correlation can play a critical role in achieving accurate and stable temporal color constancy and is currently not being exploited by these algorithms. This approach, instead, can lead to the creation of artifacts.

Color constancy in videos poses several challenges that need to be taken into consideration. A temporal color constancy algorithm needs to maintain two important characteristics. The first one is to eliminate any color cast from the image and restore it to its original state as if it were captured under a known light source, which, for the sake of brevity, will be referred to as ”accuracy” for the rest of the work. In the video domain, this characteristic should be extended to every frame of the sequence. The consistency

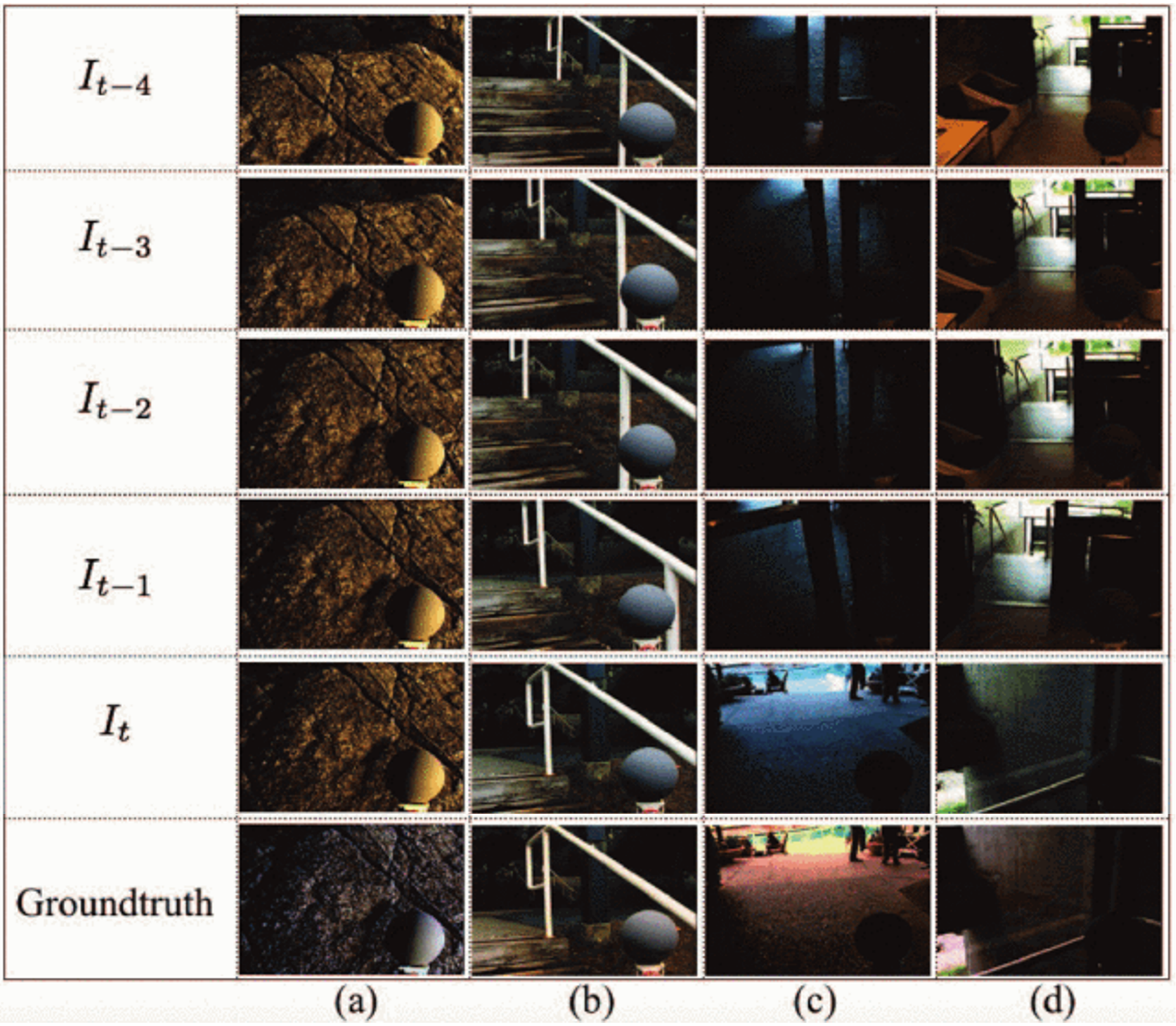


Figure 3.1: Color correction by temporal methods on frame I_t using five-frame sequences with (c)(d) and without (a)(b) significant pictorial content change, with (d) and without (a)(b)(c) significant illumination color change. Illumination color is visible on the ball in the bottom right corner. The images are from the SFU Gray Ball linear dataset, i.e. without gamma correction and thus appear to have unusual color composition. Image taken from [86]

of object colors from frame to frame when the lighting conditions are constant, known as "temporal consistency", is the second characteristic that a temporal color constancy algorithm needs to maintain. Achieving temporal consistency is a challenging task for temporal color constancy algorithms, and the lack of this characteristic creates a visual artifact known as "flickering". The processing of videos frame by frame is prone to the creation of such artifacts. However, it is essential to achieve this characteristic to ensure the overall video quality and perception of the viewer. The need to produce accurate and temporally stable results is not the only challenge to be addressed. While the state of the art suggests several metrics to evaluate the accuracy of a color constancy method, there is no metric able to evaluate the temporal stability of a frame sequence. Therefore, section

3.2 provides a common procedure to evaluate the accuracy and the temporal consistency of temporal color constancy algorithms. However, the lack of video datasets that include color targets poses an additional challenge to this task.

3.1 Related Works

Yang et al. [107] suggests a new matching invariant called illumination chromaticity constancy. They search for correspondence by analyzing the chromaticities of the color differences between the corresponding pixels in two camera images, define this correspondence as a chromaticity match, and then adopt majority voting for discretized values of the illuminant chromaticity. However, this method has a limitation in that it assumes the objects to be stationary and requires the images to be captured by a moving camera under the same lighting conditions.

Prinet et. al [85] introduces a new physically-based approach for illuminant chromaticity estimation from a temporal sequence. By keeping the same dichromatic reflection model, they show, experimentally, that the distribution of the incident light at edge points, where specularities may be often encountered, can be modeled by a Laplace distribution. This enables a robust and accurate global illuminant color estimation using a probabilistic optimization framework. Both Prinet's and Yang's methods are limited by the assumption that some surfaces in the scene have a specular reflectance component, and to process images in pairs.

Starting from the consideration that the frames in a video are generally highly correlated, Wang et. al. [102] proposes a video-based illuminant estimation algorithm that takes advantage of the common information between adjacent frames. The main idea is to cut the frame sequence into different "scenes". Assuming that all the frames in a scene are under the same illuminant, they propose a combination of frame-based illuminant estimations to calculate the global illuminant estimation of the scene. The algorithm consists of four main steps 1) Frame-based illuminant estimation, 2) scene cutting, 3) scene illuminant calculation 4) frame illumination adjusting.

Qian et. al. [86] proposes the RCC-Net, a novel recurrent deep net, which consists of a convolutional backbone for extracting spatial features, a convolutional LSTM for recurring features, a novel simulated sequence component, and a shallow network for merging. The baseline established by RCCNet was recently overcome by TCCNet [88], an improved version of the same method featuring a more powerful backbone CNN and a 2DLSTM for the sequential processing of the encoded frames.

3.2 On the Evaluation of Temporal and Spatial Stability of Color Constancy Algorithms

Color constancy methods are usually compared with angular error metrics such as the recovery error [51] and the reproduction error [32]. The comparison of color constancy methods based on angular errors is sometimes aided by statistical tools such as the Wilcoxon test [103], or by graphical tools such as the Angle-Retaining Chromaticity diagram [19]. Angular error evaluation has been extremely useful in assessing color constancy methods and guiding the research for many years. However, it neglects other important aspects of the characterization of color constancy algorithms, related to their stability in the video domain.

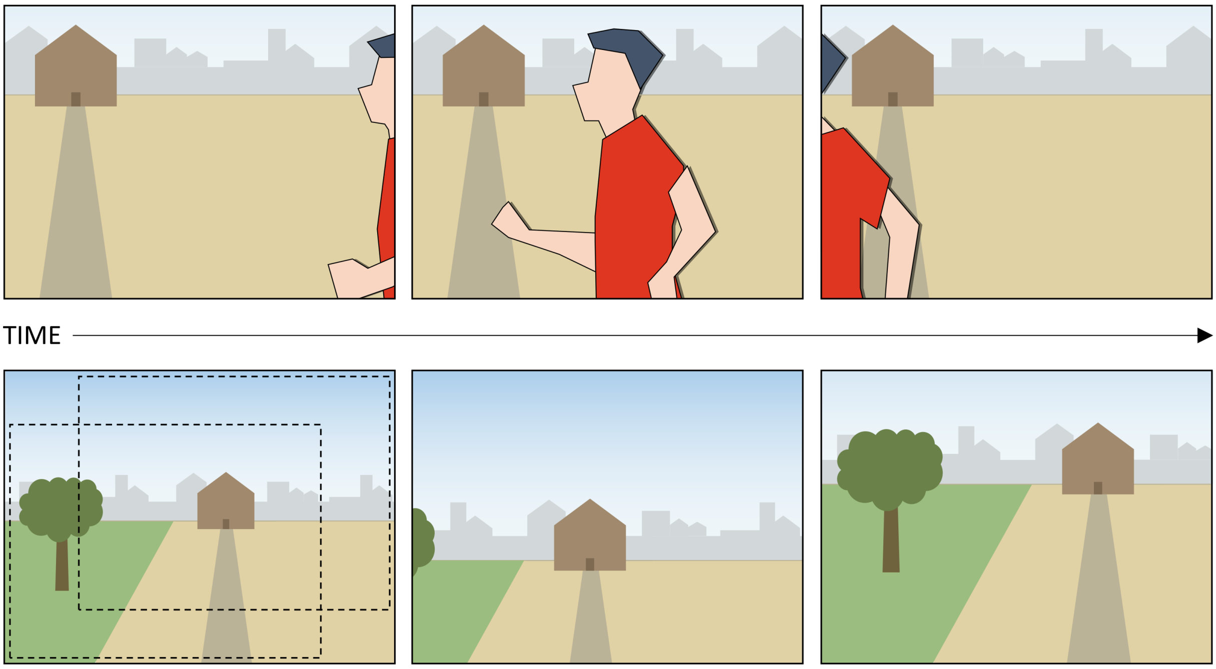


Figure 3.2: Changing scene content under constant illuminant conditions moving subjects in front of the camera (top) and panning/zooming camera (bottom). A color constancy algorithm that is temporally and spatially stable should provide a self-consistent response in these scenarios.

This property has become extremely valuable since consumer devices are increasingly used for video acquisition and reproduction [104] in this context, the discomfort of poor illuminant correction is potentially amplified if such correction also changes over time without justification, thus introducing unpleasant flickering artifacts. To this extent, existing works have tackled the problem of temporally-aware color constancy [86, 88], exploiting the information coming from multiple frames in order to produce a more robust illuminant estimation. Nonetheless, traditional single-frame methods can also be applied to video sequences, with or without the aid of temporal consistency post-processing [65]. As such, the main goal of this investigation is to study the direct applicability of single-frame color constancy algorithms in the video domain [17]. Two possible scenarios of interest have been identified moving subjects in front of the camera, and panning/zooming camera, as depicted in Figure 3.2. In these cases, if the scene illuminant remains constant, the expected behavior of a color constancy algorithm is that the output is also constant, ignoring the intrinsic chromaticity of newly framed elements.

The two scenarios of interest can be analyzed by resorting to appropriately annotated datasets for video color constancy, the best to date being the Gray Ball dataset [25], and the very recent Burst Color Constancy dataset (BCC) [88]. Both scenarios are depicted in such datasets, and often co-occur in the same video sequence. Temporal stability is a critical characteristic that determines the reliability of an algorithm. It refers to the algorithm's ability to produce consistent output responses when the lighting conditions remain unchanged, but the scene content changes over time. In other words, an algorithm with high temporal stability would be able to provide reliable results even when the environment is dynamic. In addition, the specific "panning/zooming camera" scenario can be synthetically recreated by introducing cropping operations at various scales in individual frames of a color constancy dataset. This is similar to a pre-processing technique proposed by Qian et al. [86] to simulate a camera movement for robust burst color constancy. This type of analysis of synthetic data allows for a more precise information about the specific case of a moving camera with stationary content. Given the nature of the experiment, the term *spatial stability* will be referred to as the capability of an algorithm to maintain a consistent illuminant estimation, assuming a unique illumination source, when looking at different portions of the scene.

For a real-life application, the assumption of unique illumination source is seldom completely verified, due for example to the presence of multiple lights at different correlated color temperatures, mutual surface inter-reflections, or coexistence of sun/shadow areas. Nonetheless, it is arguable that a time-consistent correction of the image color can be a desirable property, as long as moderate camera movements are in action. In other words, spatial stability should not be expected, nor enforced, for drastic changes in image framing. For this reason, the synthetic panning and zooming operations are applied at various scales, the trending stability is shown at all levels, and eventually focus the conclusions at the highest scale, which corresponds to the most moderate camera movement. Many algorithms for single-frame color constancy internally perform a spatially-varying illuminant estimation, which is eventually mapped back to a global illuminant by consensus [20], or clustered to perform multi-illuminant estimation [87]. Considerable efforts have been also put by the scientific community into providing spatially-varying annotation of illuminant information, resorting to moving color targets such as with the DRONE dataset [5], or to computer-generated imagery such as with the MIST dataset [46].

This work first analyzes the aforementioned temporal and spatial stability properties of existing suitable datasets for computational color constancy. Then it presents a methodology to analyze the stability of color constancy algorithms themselves. The goal is to define a common procedure for methods comparison, and to assess the degree to which such methods are sensitive to changes in the scene that do not depend from the illuminant source. The information emerging from our analysis identifies methods that are intrinsically stable, thus holding the greatest potential for expansion to video color constancy without heavily relying on temporal consistency post-processing techniques.

3.2.1 Datasets selection and pre-processing

In order to perform the stability analysis of color constancy algorithms free of any bias from the underlying data, it is necessary to exploit datasets that are, respectively, temporally and spatially stable. This section is dedicated to verify whether this condition is met on existing datasets and, when it is not, the required pre-processing steps is described. The same procedure could be potentially applied to any future datasets for video color constancy that is properly annotated.

3.2.1.1 Gray Ball Dataset

Gray Ball [25] is one of the few datasets potentially suitable for temporal color constancy. It contains 11,346 images divided into 15 sequences, with many shots acquired at close interval one from another. Many of the images depict people and include both indoor and outdoor scenarios, the latter taken in two different locations. The dataset was collected using a Sony VX-2000 digital video camera, and every shot includes the eponymous gray ball color target for ground truth annotation in the bottom-right corner. For illuminant estimation, the images have been masked to exclude the color target starting from pixel row 135 and column 226. The 360×240px images are provided in non-linear 8bit RGB format. Since several color constancy methods rely on the assumption of linear sensors, the following pipeline has been applied

1. Linearize the image (gamma correction with $\gamma = 2.2$)

2. Estimate the illuminant

3. De-linearize the estimated illuminant ($\gamma = {}^{1}\!/_{2.2}$)

It should be noted that the precise value for gamma correction is derived by common usages of the Gray Ball dataset [39], but it is not guaranteed to match the actual device characterization. This linearization strategy, despite being an approximation for color constancy outside the camera pipeline [2], still allows to process images that are closer to the RAW sensor data with respect to the original sRGB, while at the same time performing error analysis between the output of unaltered existing methods, and the official dataset ground truth.

The Gray Ball dataset does not respect the temporal stability characteristics needed for this work, therefore it requires a specific pre-processing to remove temporally-unstable sequences. For the purpose of this work, a temporally-stable sequence is defined as a sequence that 1) does not contain video cuts, 2) does not involve abrupt illuminant changes and 3) does not span a wide set of illuminants (even if gradually changing). These three conditions will be addressed in three different ways.

The video cuts have been resolved by human selection, meaning that the 15 original sequences of the Gray Ball dataset have been manually divided into 337 smaller sequences, containing only smooth transitions of the scene content. The final distribution of the resulting sequences length is shown in Figure 3.3.

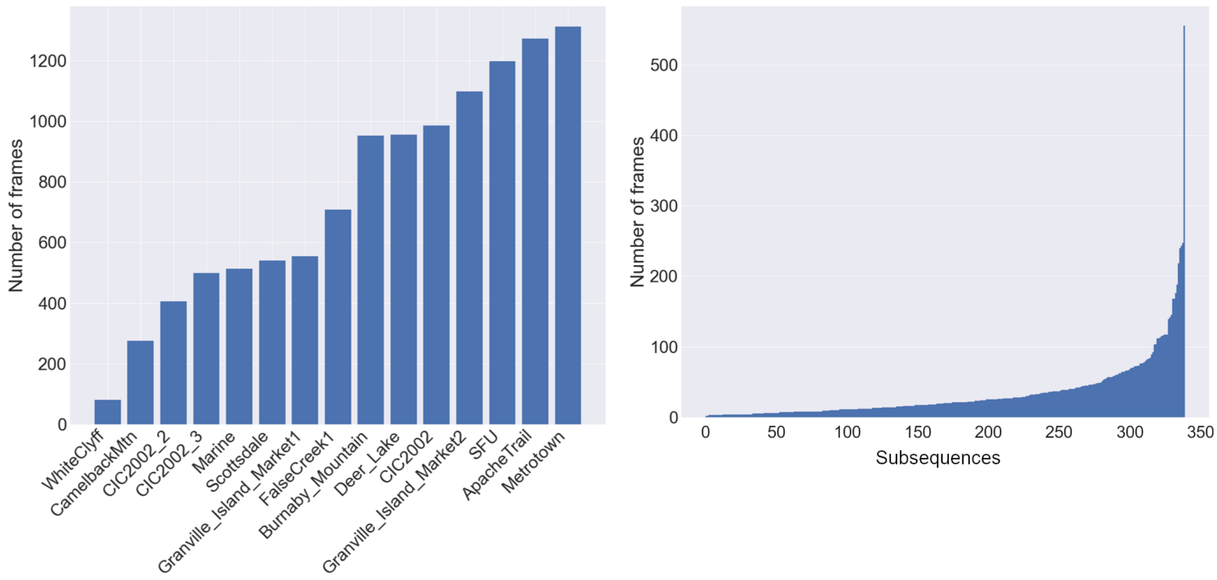


Figure 3.3: Visualization of the number of frames for each sequence of the Gray Ball dataset before the manual division for video cuts (left) and after (right).

The identification of abrupt illuminant changes has been achieved by quantifying the largest change in the expected illuminant $E = (r, g, b)$ between consecutive frames. More precisely, for each pair of consecutive frames in a sequence S, it has been calculated the recovery error between their ground truth illuminants, and then the maximum of such errors has been selected. From now on this metric will be called "maximum illuminant change" (MIC)

$$MIC(S) = \max(err_{rec}(E_{S_i},\ E_{S_{i+1}})),\ i = 1...N_S - 1 \tag{3.2}$$

Where N_S is the number of frames of sequence S.

The recovery error [51] as used in Equation 3.2, is computed between two generic illuminants as in equation 1.2.

Rather, a metric for scatteredness has been used to identify sequences spanning a large range of illuminants. Specifically, the ground truth illuminants have been converted into Angle-Retaining Chromaticity (ARC) [19], a bidimensional representation where euclidean distances correspond to angular distances in the original RGB space. Then, the standard distance [16] of the resulting points has been computed, which is a bidimensional generalization of the standard deviation, defined as

$$STD(S) = \sqrt{\sum_{i=1}^{N_S} \frac{(x_{S_i} - \overline{x_S})^2}{N_S} + \sum_{i=1}^{N_S} \frac{(y_{S_i} - \overline{y_S})^2}{N_S}} \tag{3.3}$$

where (x_{S_i}, y_{S_i}) are the ARC coordinates of the i-th illuminant of sequence S, and $(\overline{x_S}, \overline{y_S})$ indicates the average of each coordinate for the sequence.

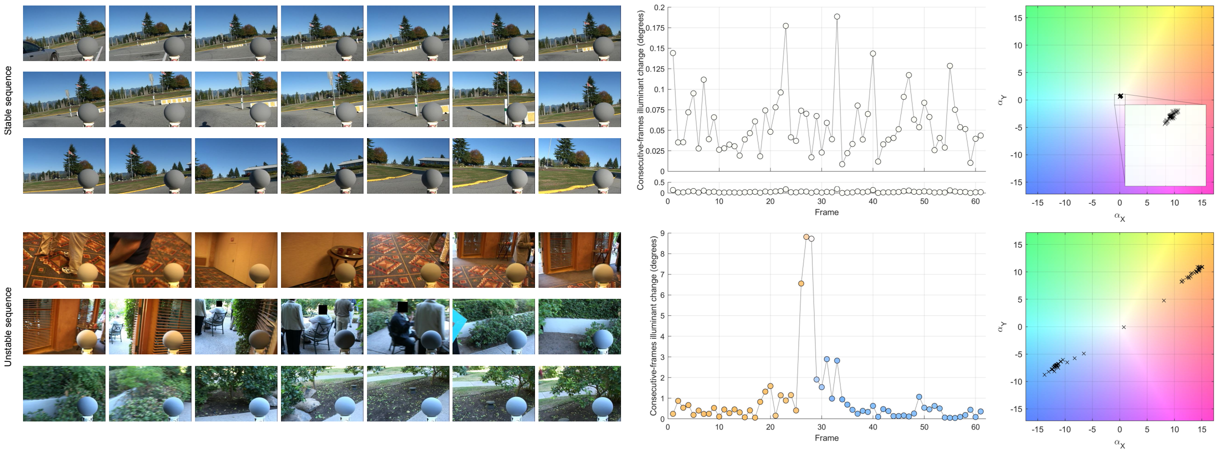


Figure 3.4: Temporal stability analysis of two sequences from the Gray Ball dataset a stable sequence ($MIC = 0.189, STD = 0.115$) (top) and an unstable sequence ($MIC = 8.812, STD = 14.912$) (bottom). For each sequence, it shows a sample of the frames (left), the illuminant change between consecutive frames (center), and the illuminants distribution in ARC diagram (right).

The information captured by these measures is visualized in Figure 3.4 for each sequence, the illuminant change between consecutive frames is shown (whose maximum corresponds to MIC), as well as the ground truth illuminants distribution in Angle Retaining Chromaticity (whose scatteredness corresponds to STD). The two metrics, MIC and STD, were then combined to provide a single value that describes the instability of each sequence first, it has been computed the standard score of both metrics, by normalizing them for the corresponding cross-sequence average and standard deviation, and subsequently, an equal-weight average is been computed. The resulting distribution was finally split in half by using the median value as a threshold to divide the dataset into 168 stable sequences and 169 unstable sequences. In the following, the term “filtered Gray

Ball dataset" will stand for the selection of temporally stable sequences. Backtracking this division to the initial measures, it roughly corresponds to applying a threshold over MIC at 1.5° and STD at 0.8°, which appear adequate after a visual inspection of the dataset. However, due to the arbitrary nature of any specific threshold, our entire dataset division into subsequences is available, along with the corresponding values of stability-related measures, so as to allow further developments by other researchers [18].

With respect to spatial stability, the Gray Ball dataset has been used for many years for global illuminant estimation analysis, under the implicit assumption of spatial stability. This assumption is, however, generally unsubstantiated. It is possible, for example, to find outdoor scenarios where part of the scene is illuminated by direct sunlight, and part of it is in shadow, illuminated only by the blue of the sky. This configuration breaks the assumption of constant illumination across the image, and, furthermore, it cannot be automatically filtered. Despite the fact that the color target (the gray ball) can capture incoming light from different directions, it only describes the illumination condition in the foreground of the picture, and it does not provide any way to associate the different illuminant chromaticities to specific image regions. Notwithstanding these considerations, for the sake of completeness and for consistency with the existing corpus of color research, spatial stability analysis will be performed on this dataset as well.

3.2.1.2 Burst Color Constancy Dataset

The Burst Color Constancy dataset (BCC), sometimes referred to as the Temporal Benchmark dataset, was recently presented by Qian et al. [88], and it was specifically collected to meet the requirements of the temporal color constancy problem. It consists of 600 sequences of varying lengths (between 3 and 17 frames), divided into 400 sequences for the training set and 200 for the test set, the latter used in our analysis. Consistently with the Gray Ball dataset, BCC covers indoor and outdoor scenes with varying weather and daylight conditions. The images were shot with the use of a Huawei Mate 20 Pro mobile phone, and stored in a proprietary 16-bit RAW format. Reprocessed 8-bit PNG images at 3648×2736px resolution were also made available by the dataset authors, and these were specifically used in our work.

With respect to the temporal color constancy problem, the sequences collected for the BCC dataset are assumed implicitly stable by design. This allowed the authors to avoid capturing the images with a color calibration target installed in the scene, and thus to avoid unintentionally conveying information to learning-based methods. Instead, the SpyderCube calibration target was put in the scene immediately after the sequence acquisition, to create an out-of-sequence reference shot that represents the entire video sequence. For these reasons, there has been no need, nor possibility, of a pre-processing step.

Concerning spatial stability, the BCC dataset was collected and presented without any explicit statement in terms of single or multiple illuminant sources. A visual inspection of the dataset images confirmed the absence of images visibly illuminated by multiple sources of light, with the exception of few daylight/shadow instances as observed in the Gray Ball dataset as well.

3.2.2 Analyzed color constancy methods

The stability analysis has been conducted on a selected variety of color constancy algorithms, including traditional solutions based on handcrafted features, as well as more recent approaches based on deep learning. All methods are sensor-independent and, when necessary, trained on different datasets than the ones used for our analysis, so as to ensure the absence of any bias and to provide fair results. This particular set of methods has been selected as a case study, however, the same procedure can be applied to any existing method for color constancy.

Edge-based color constancy (EB) [100] is a popular framework introduced in 2007 by van de Weijer et al. as a generalization of multiple algorithms based on low-level image statistics. The free parameters of these methods (Minkowski norm p and standard deviation σ) have been selected as reported in [12]

- Grey World (GW) $p = 1$, $\sigma = 0$.
- White Point (WP) $p = \infty$, $\sigma = 0$.
- Shades of Gray (SoG) $p = 4$, $\sigma = 0$.
- General Grey World (GGW) $p = 9$, $\sigma = 9$.
- 1st order Grey Edge (GE1) $p = 1$, $\sigma = 6$.
- 2nd order Grey Edge (GE2) $p = 1$, $\sigma = 1$.

The standard deviation parameter σ describes the Gaussian filter applied by the underlying algorithms, and as such its impact on the final performance is tightly related to the size of the input image. The images from the BCC dataset have been downscaled to have the maximum side be 360 pixels long, thus reaching the same dimensions as the images from the Gray Ball dataset. Preliminary experiments also showed that downscaling the BCC dataset resulted, on average, in better performance w.r.t. upscaling the Gray Ball dataset. This pre-processing has been applied only for Edge-Based color constancy since other algorithms have different requirements or involve an internal rescaling of the input image.

More recent color constancy algorithms have also been considered. Cheng et al. [24] introduced a color constancy algorithm based on Principal Component Analysis (PCA), observing that the mere analysis of color distribution provides as much information for illuminant estimation as a more complex spatial analysis. Their solution selects a predefined percentage of dark and bright pixels using a projection distance in the color distribution. In our experiments, the percentage parameter has been set to 3.5% following the best-performing configuration reported by the authors.

The Grayness Index (GI) [87] is a learning-free metric developed by Qian et al. to identify neutral surfaces (gray pixels) in an input image following the Dichromatic Reflection Model [94]. This allows the estimation of single illuminant as well as multiple illuminant information. The default pre-tuned parameters from the official implementation have been used in our work.

Quasi-Unsupervised color constancy (QU) [10] was developed by Bianco et al. to detect achromatic pixels in color images, after conversion to grayscale. Their solution is based on a convolutional neural network that can be trained without color constancy annotation, relying instead on the weak assumption that training images have been approximately balanced. The model used in this analysis was trained on images from the ILSVRC2012 dataset of the ImageNet initiative [90].

Fully Convolutional Color Constancy with Confidence-weighted Pooling (FC4) [54] by Hu et al. implements a neural network architecture that assigns confidence weights to various patches of an input image, based on the level of information and reliability that such patches are estimated to carry for the task of color constancy. The official implementation is supplied with pre-trained models on each fold of the ColorChecker dataset [37]. In this analysis, the SqueezeNet-based model [56] pretrained on “fold 2 and 0” has been used.

In Sensor-Independent Illumination Estimation (SIIE) [1] authors Afifi et al. developed a learnable sensor-independent pseudo-RAW space to be used to “canonicalize” the RGB values of any given camera, under the explicit assumption of input linear RAW-RGB images. Due to the nature of the Gray Ball dataset, where images are not in RAW format but already processed by an undisclosed camera pipeline, this method is expected to underperform, despite our synthetic linearization. For this analysis, the “MATLAB 2018b” model pre-trained on the NUS [24] and Cube+ [9] datasets has been used.

3.2.3 Analysis of temporal stability

This section is dedicated to the assessment of the temporal stability of color constancy algorithms, under the assumption of temporally-stable sequences such as those from the filtered Gray Ball dataset, and from the Burst Color Constancy (BCC) dataset.

Two measures have been applied to describe the temporal stability of a color constancy algorithm maximum illuminant change MIC, and standard distance STD. These are the same criteria defined in equations 3.2 and 3.3 of Section 3.2.1 to automatically identify

Table 3.1: Temporal stability and error evaluation of color constancy algorithms on the filtered Gray Ball dataset. All values are expressed in degrees, and the lower the better.

	Stability measures		Error measures	
Method	$MIC \downarrow$	$STD \downarrow$	$err_{rec} \downarrow$	$err_{rep} \downarrow$
GW [100]	3.38	2.76	6.79	7.06
WP [100]	2.81	1.55	5.76	6.01
SoG [100]	3.14	2.57	5.88	6.06
GGW [100]	3.82	3.01	6.31	6.52
GE1 [100]	2.78	2.35	5.66	5.89
GE2 [100]	**1.80**	**1.39**	5.41	5.74
PCA [24]	3.66	2.74	5.75	6.04
GI [87]	7.81	4.95	7.65	8.03
QU [10]	2.89	2.15	5.40	5.63
FC4 [54]	3.35	1.97	**4.63**	**4.97**
SIIE [1]	2.32	2.30	6.31	6.65

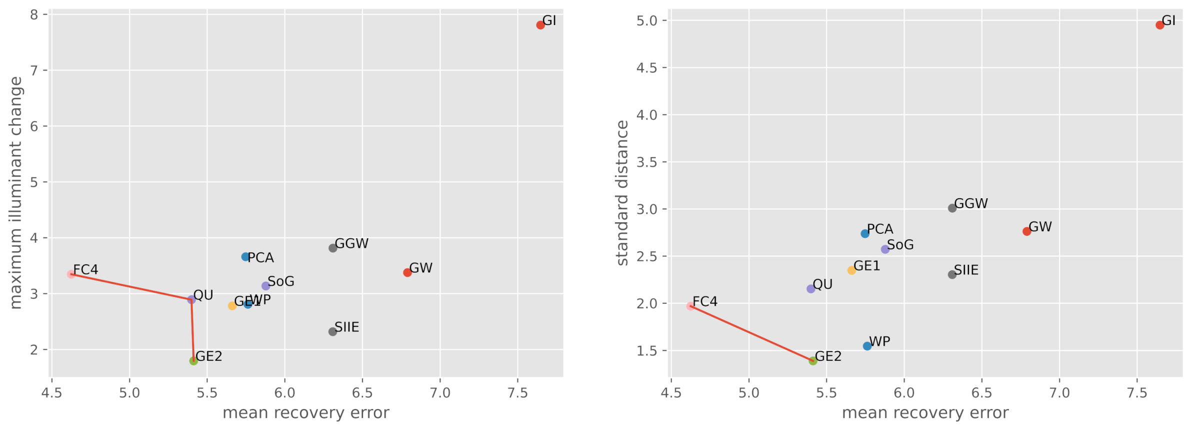


Figure 3.5: Visualization of the temporal stability and recovery error of different methods on the Gray Ball dataset. Temporal stability is expressed as either maximum illuminant change (left) or as standard distance (right). The Pareto front is visualized. For all measures, the lower the better.

stable sequences in the Gray Ball dataset, however in this case the evaluation has been performed on estimated illuminants as opposed to ground truth illuminants. Each method was assigned two temporal stability scores, by averaging each of the two aforementioned measures across the sequences, for any given dataset.

Temporal stability alone is hardly effective in evaluating the quality of a color constancy algorithm. For example, a “do nothing” algorithm would score the best value for temporal stability, while not being able to produce an effective illuminant estimation. For this reason, all algorithms have also been evaluated in terms of traditional single-frame error measures, such as the recovery and reproduction error. Recovery error and reproduction error are computed according to Equation 1.2 and Equation 1.3, respectively.

For the sake of consistency with the stability measures, in this analysis, the error values have been averaged for each frame in a sequence and subsequently averaged for the cross-sequence results. Our stability/error evaluation is conceptually equivalent to assessing a solution in terms of precision and accuracy. The results related to the filtered Gray Ball dataset are presented in Table 3.1, in terms of maximum illuminant change (MIC), standard distance (STD), recovery error (err_{rec}) and reproduction error (err_{rep}). The two temporal stability metrics are also visualized in Figure 3.5 in conjunction with the recovery error, in order to better visualize the performance of the analyzed methods.

The first thing that the experiment on the Gray Ball dataset suggests is that methods perform very similarly for both the standard distance and maximum illuminant change metrics. Generally speaking, then, stability and accuracy are also partially correlated, as highlighted in Figure 3.5. This behavior is a consequence of the focus on temporally-stable datasets on such data, a method can be globally accurate only if it is also temporally stable. This is specifically manifested in the absence of points in the top-left corner of the plots. Despite this correlation, several rank inversions are present between stability and error measures. For example, while FC4 is the most accurate method, it is surpassed in MIC-based stability by several methods. Of these, GE2 and WP appear to be the most stable in terms of the STD metric as well. Conversely, the worst method on this dataset, GI, performs consistently the worst on all the chosen metrics.

The same analysis for temporal stability has been performed on the BCC dataset, as illustrated in Table 3.2 and Figure 3.6. Similar observations from the Gray Ball can be extended to this dataset as well, in terms of correlation between the different metrics. The SIIE method outperformed FC4 on the Gray Ball dataset only in terms of MIC, while it performs consistently better for both temporal stability metrics on the BCC dataset. In this case, the single worst-performing method according to all metrics is the very simple white point (WP) algorithm.

The different conclusions that can be derived from analyzing the two datasets can be traced back to 1) the type of images the Gray Ball dataset is not distributed in linear RAW format, thus limiting the accuracy of color constancy algorithms, and 2) the type of annotations the BCC only has sequence-level ground truth information, which is in line with the assumption of temporal stability, but reduces the precision of the analysis.

3.2.4 Analysis of spatial stability

This section is dedicated to the assessment of the spatial stability of color constancy algorithms, under the implicit assumption of spatially stable datasets. Each image has been divided into five windows on a set of fixed locations that cover the entire image one for each angle and one in the center. A larger window size implies a higher overlap among windows, which corresponds to moderate camera movements in our synthetic setup.

Table 3.2: Temporal stability and error evaluation of color constancy algorithms on the BCC dataset. All values are expressed in degrees, and the lower the better.

	Stability measures		Error measures	
Method	$MIC \downarrow$	$STD \downarrow$	$err_{rec} \downarrow$	$err_{rep} \downarrow$
GW [100]	2.70	3.20	5.32	7.08
WP [100]	6.31	5.07	6.98	8.26
SoG [100]	4.10	3.49	5.31	6.95
GGW [100]	4.37	4.01	5.92	7.61
GE1 [100]	3.58	3.40	5.52	7.30
GE2 [100]	3.40	3.01	4.79	6.10
PCA [24]	4.01	3.33	5.16	7.12
GI [87]	4.11	3.50	4.63	6.30
QU [10]	3.26	2.95	4.28	5.87
FC4 [10]	3.08	2.64	**4.21**	**5.75**
SIIE [1]	**2.54**	**2.52**	4.49	6.06

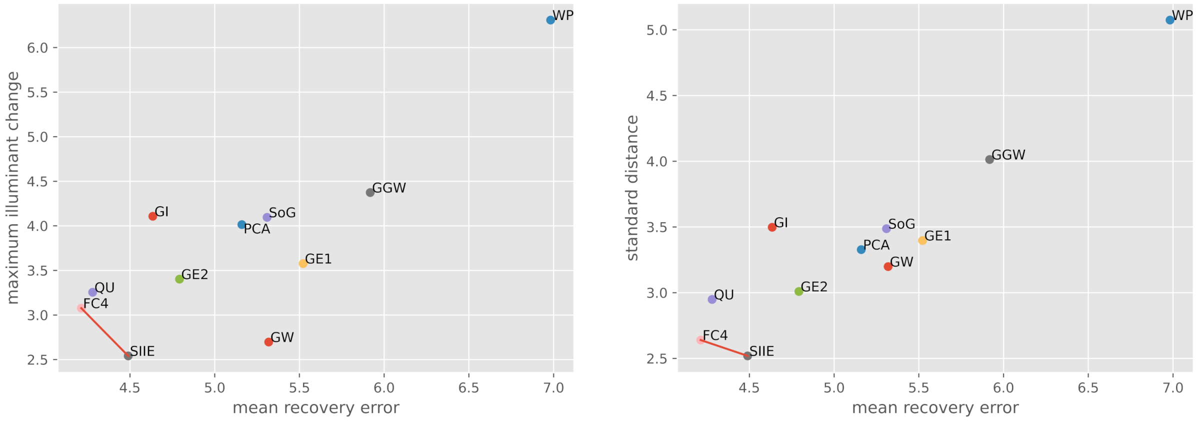


Figure 3.6: Visualization of the temporal stability and recovery error of different methods on the BCC dataset. Temporal stability is expressed as either maximum illuminant change (left) or as standard distance (right). The Pareto front is visualized. For all measures, the lower the better.

In such scenarios, it is arguable that having a consistent output in the color correction is a desirable property also for real-life applications, as the change in overall incident illumination on such scale can be expected to be limited. Given the arbitrary nature of fixing a window size, information is presented at various scales from 50% to 90% of the original image sides, with a step of 10%, and the final scale (90%) will be used as a reference to derive any conclusions about spatial stability.

All the analyzed color constancy methods have been applied to each window of each image. As for the temporal stability problem, it is important to capture both the accuracy and precision of analyzed algorithms. Specifically, they are presented in the form of angular errors err_{rec} and err_{rep} for accuracy, and standard distance STD for precision.

Table 3.3: Spatial stability evaluation of color constancy algorithms on the Gray Ball dataset at 90% window side. All values are expressed in degrees, and the lower the better.

	Stability measures	Error measures	
Method	$STD \downarrow$	$err_{rec} \downarrow$	$err_{rep} \downarrow$
GW [100]	0.49	7.09	7.62
WP [100]	0.34	6.83	7.08
SoG [100]	0.44	6.17	6.49
GGW [100]	0.60	6.82	7.19
GE1 [100]	0.45	6.05	6.44
GE2 [100]	**0.28**	5.74	6.16
PCA [24]	0.49	6.40	6.85
GI [87]	0.96	7.02	7.61
QU [10]	0.55	6.45	6.73
FC4 [54]	1.12	**5.67**	**6.02**
SIIE [1]	0.49	7.53	7.92

The maximum illuminant change MIC was well suited to highlight flickering phenomena in temporal sequences, but it does not provide any meaningful information if applied to the five windows of spatial stability analysis.

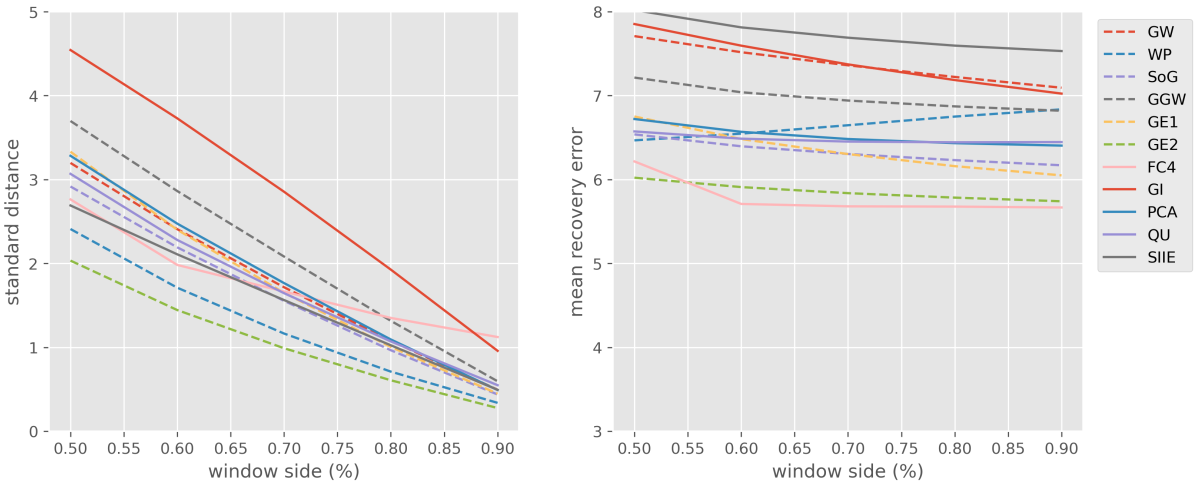


Figure 3.7: Visualization of the standard distance (left) and recovery error (right) of different methods on the Gray Ball dataset. For all measures, the lower the better.

Table 3.3 and Figure 3.7 report the aforementioned metrics on the Gray Ball dataset, with a detail per window size in the figure. Since larger windows are necessarily more overlapped, they are expected to lead to better stability, so any comparison should be made across methods and not across window sizes. This expectation is verified, as visible in the left side of Figure 3.7, where all curves exhibit a monotonic decreasing behavior. The plotted information at 90% window side is also presented numerically in Table 3.3. The most spatially stable algorithm on the Gray Ball dataset is GE2, in line with the

Table 3.4: Spatial stability evaluation of color constancy algorithms on the BCC dataset at 90% window side. All values are expressed in degrees, and the lower the better.

	Stability measures	Error measures	
Method	$STD \downarrow$	$err_{rec} \downarrow$	$err_{rep} \downarrow$
GW [100]	**0.24**	5.65	7.46
WP [100]	0.55	6.70	7.99
SoG [100]	0.31	5.52	7.18
GGW [100]	0.40	6.07	7.77
GE1 [100]	0.38	5.88	7.72
GE2 [100]	0.29	5.03	6.38
PCA [24]	0.30	5.40	7.39
GI [87]	0.40	4.87	6.55
QU [10]	0.40	4.45	6.04
FC4 [54]	0.43	**3.84**	**5.33**
SIIE [1]	0.29	4.55	6.10

analysis on temporal stability from Section 3.2.3. Interestingly, FC4 is the least stable algorithm specifically for large window sizes (90%), although it maintains a relatively consistent stability performance for smaller window sizes, compared to other methods. In terms of error analysis, a general trend of improvement with larger window sizes is also expected from the recovery error curves in Figure 3.7. To this extent, the only exception appears to be WP, which exhibits a reverse trend one possible explanation is that, by forcing the algorithm to ignore parts of the image, it is more likely to produce a better estimation from one or more windows. The behavior of FC4 is also unusual, displaying a significant improvement from 50% window size to 60%, but then essentially maintaining the same error performance for the remaining window sizes. Nonetheless, it is, in general, the best-performing method, consistently with the Gray Ball dataset analysis.

Table 3.4 and Figure 3.8 present the same spatial stability analysis on the BCC dataset, showing an overall comparable behavior. The expectation to observe improving performance with increasing window sizes is respected also for this dataset, with the only observed exception being the WP on the recovery error metric. FC4 is the most accurate algorithm, with a significant gap from the second-best methods QU and SIIE. In terms of spatial stability, however, the conceptually simple GW appears to exhibit the best performance.

It is also interesting to observe that, on the BCC dataset, the error statistics on learning-based algorithms can be neatly separated from those of traditional handcrafted solutions, while this does not apply to the Gray Ball dataset.

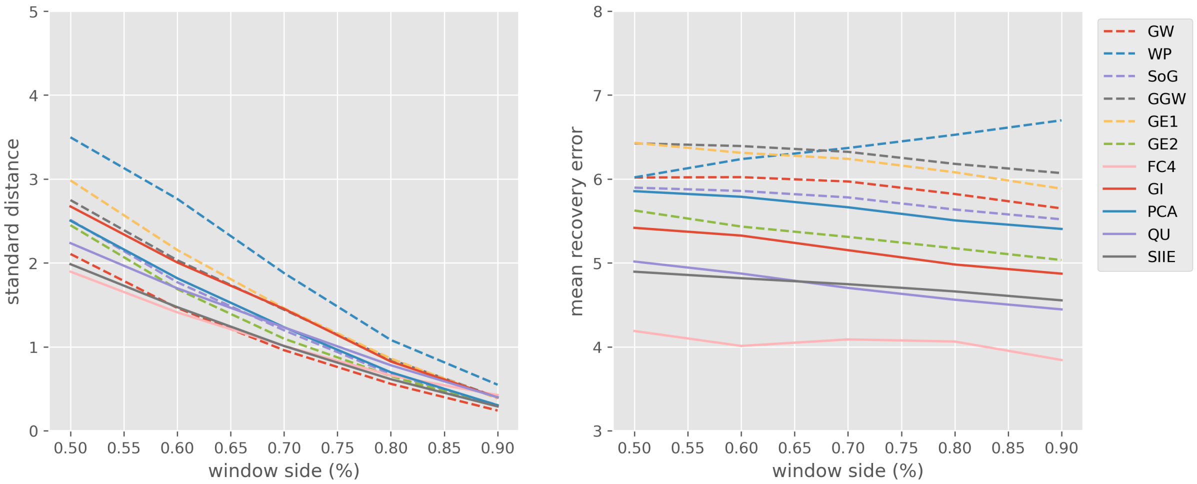


Figure 3.8: Visualization of the standard distance (left) and recovery error (right) of different methods on the BCC dataset. For all measures, the lower the better.

3.2.5 Discussion of aggregated results

The analysis presented in this paper highlights the importance of evaluating algorithms according to multiple measures, and it can be useful in selecting the most appropriate color constancy method depending on specific application constraints.

To this extent, Table 3.5 presents an aggregated view of the information produced in the previous sections. The temporal stability rank is based on an average of temporal stability measures MIC and STD on both the Gray Ball and BCC datasets. Similarly, the spatial rank is based on the average of spatial STD on both datasets, and the error rank is based on err_{rec} and err_{rep} on the two datasets. Finally, a global rank is presented in the last column of Table 3.5. This ordering is determined through Borda's method for rank aggregation [14, 70], where individual ranks are averaged, and elements are re-ranked based on such average. This approach enables the combination of statistics coming from different domains (temporal, spatial, error), as it provides invariance to the different magnitude and distribution of the underlying values. Equal-weighted average has been used for our analysis, although in the future different applications might motivate the selection of different weights for the temporal, spatial, and error components. In this particular setup, the best ranking method at a global level appears to be GE2, coherently with the individual temporal and spatial rank assessments. The second method is SIIE, which strikes a good balance across all evaluation criteria. The highly-accurate and temporally-stable method FC4 is penalized in global rank by its spatial instability, thus achieving intermediate overall performance. Finally, the lowest-ranked method in our experimental setup appears to be GI, which is negatively impacted by its poor error performance on the Gray Ball dataset, and by its generally low stability.

Table 3.5: Aggregated ranks of the analyzed color constancy methods, according to multiple measures. Underlying values are expressed in degrees

Method	Temporal rank	Spatial rank	Error rank	Global rank
GW [100]	5 (3.01)	2 (0.37)	9 (6.76)	6
WP [100]	10 (3.94)	7 (0.44)	11 (6.95)	9
SoG [100]	7 (3.33)	2 (0.37)	4 (6.19)	3
GGW [100]	9 (3.80)	9 (0.50)	10 (6.78)	9
GE1 [100]	6 (3.03)	6 (0.42)	7 (6.31)	7
GE2 [100]	1 (2.40)	1 (0.28)	3 (5.67)	1
PCA [24]	8 (3.44)	5 (0.40)	6 (6.26)	7
GI [87]	11 (5.09)	10 (0.68)	8 (6.58)	11
QU [10]	4 (2.81)	8 (0.47)	2 (5.61)	4
FC4 [54]	3 (2.76)	11 (0.77)	1 (5.05)	5
SIIE [1]	2 (2.42)	4 (0.39)	5 (6.20)	2

3.2.6 Conclusions

This work presents a new methodology to evaluate color constancy algorithms by taking into account their temporal and spatial stability. Two color constancy datasets have been selected from the state of the art the Gray Ball and the BCC, which have been analyzed and pre-processed for the purpose of this evaluation, making the resulting characterization available for public download. A case study has been conducted on a wide set of color constancy algorithms, although the presented evaluation methodology can be applied to any given method. Concerning temporal stability, which measures the output consistency throughout frames in a video sequence, it has been observed a general correlation with traditional error metrics. However, some notable exceptions have been identified. For example, the popular FC4 algorithm is consistently the best-performing one in terms of angular error, but it is outperformed in terms of stability by the SIIE algorithm on both analyzed datasets, and by several other methods on the Gray Ball dataset. The spatial stability analysis, which evaluates their output consistency across multiple windows of the input image, also led to similar conclusions FC4 has been identified as the least stable algorithm for large window sizes on the Gray Ball dataset, and among the least stable ones on the BCC dataset, despite confirming its supremacy in terms of traditional error measures.

The analysis conducted in this paper lays the basis for identifying those single-shot color constancy algorithms that hold the greatest potential for expansion to video color constancy. Future investigations could also account for computational complexity a method that is characterized by good or average performance in terms of traditional angular error, but which displays scarce temporal and spatial stability, would potentially require a post-processing step to enforce temporal consistency. The resulting overhead at inference time could be prohibitive for a video-oriented application if the initial method is not inherently efficient.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in Ref. [18].

3.3 LSTM for the illuminant estimation in mobile settings

Burst color constancy exploits a small group of frames to estimate the sequence illuminant, based on the assumption that those frames have been acquired under the same lighting condition. However, in video acquisition, the illuminant can change from frame to frame, for example, the video may be acquired by a person walking from a room with natural light to a room with artificial light, or it may be acquired in a forest where the lighting changes due to the movement of the sun and the shadows cast by the trees or, yet again, in a city street where the lighting changes due to the presence of different types of street lamps. Temporal Color Constancy, instead aims at estimating the illuminant of the N_{th} frame of the sequence (referred to as shot-frame). To do so it suggests exploiting the N preceding frame to the shot frame, in order to base the estimation on more information. We extend this concept, by saying that if the N_{th} frame can benefit from the information of the first frame, then also the first frame can benefit from the information of the N_{th} frame. Therefore, we propose a method that given a sequence of frames returns an illuminant estimation for each frame.

3.3.1 Method

The goal of this work is to establish a method for achieving temporal color constancy that guarantees consistent illuminant estimations over time. The method takes a sequence of frames as input and produces an illuminant estimation for each individual frame. The illuminant estimations not only need to correct the color of each frame, but they also need to remain stable over time when compared with adjacent frames and frames containing the same objects.

The method consists of a recurrent neural network, composed of 1) a convolutional neural block, designed to extract semantic features, and 2) an LSTM block, that provides spatial recurrent information. For the convolutional neural block, we selected the

MobileNetv2 architecture since it is specifically tailored for resource-constrained environments [92] and fast execution time. This aspect becomes particularly problematic in a video-oriented domain, where time is considered critical. Fast execution time is particularly important in real-time scenarios, such as assisted driving, but also for off-line color constancy set-up, where a fast computation is still critical for long video sequences. In such scenarios, therefore, it is fundamental to select an efficient architectural design [112].

Two are the main characteristics of MobileNetv2 the depthwise separable convolution and the linear bottlenecks.

Depthwise separable convolution [53] is an operation that replaces a full convolutional operation with a factorized version that splits convolution into two separate layers. The first layer is called a depthwise convolution, it performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a 1 × 1 convolution, called a pointwise convolution, which is responsible for building new features through computing linear combinations of the input channels. Depthwise separable convolution works empirically as well as a traditional convolutional operation and reduces computation compared to traditional layers by almost a factor of k^2, where k is the kernel size.

The linear bottleneck layers, instead, are responsible for reducing the dimensionality of the activation space by preserving the manifold of interest. The bottleneck blocks appear similar to residual blocks where each block contains an input followed by several bottlenecks then followed by expansion [47]. The structure of a bottleneck layer is shown in Table 3.6 for a block of size $h \times w$, expansion factor t and kernel size k with $d\prime$ input channels and $d\prime\prime$ output channels.

Input	Operator	Output
$h \times w \times k$	1×1 conv2d, ReLu6	$h \times w \times (tk)$
$h \times w \times tk$	3×3 dwise s=s, ReLu6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times (tk)$	linear 1×1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k\prime$

Table 3.6: Bottleneck residual block transforming from k to $k\prime$ channels, with stride s, and expansion factor t.

Table 3.7, instead, shows the entire architecture of MobileNetV2. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers described in the Table 3.6. ReLU6 has been selected as the non-linearity activation function because of its robustness when used with low-precision computation [27]. Kernel size is also set up to be 3×3 as it is standard for modern networks, and utilizes dropout and batch normalization during training.

For this work, we used the entire structure of MobileNetV2, with the exception of the last two layers. Instead, we decided to feed the resulting feature map to an LSTM layer,

input	operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	-

Table 3.7: MobileNetV2 Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3 × 3 kernels. The expansion factor t is always applied to the input size as described in Table 3.6.

with the purpose of capturing spatial recurrent features.

Long-short-term memory (LSTM) networks have been first introduced by Hochreiter and Schmidhuber [49]. They are a special kind of Recurrent Neural Network capable of learning long-term dependencies in sequences. Traditional Recurrent Neural Networks are sensitive to back-propagation error, which grows or shrinks at each time step until it blows up or vanishes. It has been empirically proven that a traditional Recurrent Neural Network cannot bridge more than 5-10 time steps [96]. LSTM is a gradient-based method capable of addressing the gradient vanishing problem. A common LSTM unit is composed of a cell, an input gate, an output gate [48], and a forget gate [38]. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Forget gates decide what information to discard from a previous state by assigning a previous state, compared to a current input, a value between 0 and 1. A (rounded) value of 1 means to keep the information, and a value of 0 means to discard it. Input gates decide which pieces of new information to store in the current state, using the same system as forget gates. Output gates control which pieces of information in the current state to output by assigning a value from 0 to 1 to the information, considering the previous and current states. Selectively outputting relevant information from the current state allows the LSTM network to maintain useful, long-term dependencies to make predictions, both in current and future time steps. In greater detail, during training, our LSTM layer takes a sequence of dimensionality 20×1280, it computes a series of hidden states $(h_1, h_2, ..., h_t)$ and produces an output of dimensionality 20×51. The computation is carried out by iterating the following operations

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where h_t is the hidden state at time t, c_t is the cell state at time t, x_t is the input at time t, h_{t-1} is the hidden state of the layer at time $t-1$ or the initial hidden state at time 0, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. σ is the sigmoid function, and $\odot$ is the Hadamard product [91].

The resulting feature map h_t is then fed to a fully connected layer, with a σ sigmoid activation function, of dimensionality 51×3. Thus, the final network architecture resulting is shown in Table 3.8.

input	operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$n_f \times 1280$	LSTM 1280×51	-	51	1	-
$n_f \times 51$	FC 51×3	-	3	1	-

Table 3.8: Final architecture structure, where n_f is the number of frames provided to the network. The description of columns t, c, n, and s, is the same as the one provided for 3.7.

3.3.2 Experiments

A temporal color constancy method should provide illuminant estimations that satisfy two main characteristics. The first one is the "accuracy", with this term we indicate the ability of a color constancy method to return an illuminant estimation able to discard any color cast from the given image. The second term, instead, is the temporal consistency, which is the ability of a given algorithm to maintain a consistent output response over similar illuminated frames, despite the change of content. Therefore, this work proposes to evaluate the performance of the hereby presented algorithm by taking into consideration both the accuracy and the temporal consistency characteristics. The accuracy will be evaluated in terms of recovery error (as defined in Equation 1.2), while the temporal consistency will be evaluated in terms of ”max illuminant change” (MIC) and standard deviation (as defined respectively in Equation 3.2 and 3.3).

	sequence mean recovery error	sequence median recovery error	sequence max recovery error	mean standard deviation	mean MIC	single frame mean recovery error
all sequences	2.54	2.11	13.16	0.62	0.70	2.40
stable sequences	2.00	0.81	9.52	0.65	0.71	1.90
unstable sequences	2.86	2.45	13.16	0.61	0.70	2.68

Table 3.9: Evaluation of the performance of the presented method. The evaluation has been conducted both in terms of recovery error (°), MIC, and standard deviation, for all the metrics the lower the value the better. The performance is also analyzed by the stability of the frame sequence.

Table 3.9 returns a view image of the performance of the proposed method for temporal color constancy. Since the stability of a sequence of frames is a key factor in an analysis like this one, the performance is reported for all the sequences, and for the stable and unstable sequences separately. Also, the first four columns report the results of the recovery error per sequence, which means that first the recovery errors of a sequence get averaged, and then the other metrics are performed. While, the last column of the table, returns the single frame mean recovery error performance, which means that the frames are considered as single images, instead as part of a sequence. As predicted the table shows that the method performs better for the stable sequences than for the unstable ones in terms of recovery error, in fact, we can see an improvement of 30% for the sequence mean recovery error for the stable sequences compared to the unstable ones. This is even more evident for the sequence median recovery error, where the improvement for the stable sequences compared to the unstable ones is 66%. Instead, this is not true for the stability metrics, for which the method has similar performance despite the stability of the sequence provided as input.

The state-of-the-art does not provide a method that extends the concept of temporal color constancy as the one proposed in this work, therefore, the most suitable one has been

method	mean recovery error	median recovery error
RCC	9.33	7.40
MN+LSTM	2.12	1.06

Table 3.10: Performance comparison between the MobileNet+LSTM method with the state-of-the-art RCC method. The comparison is performed in terms of mean and median recovery error, expressed in degrees (°) the lower the better.

selected for comparison (Recurrent Color Constancy (RCC) [86]). For a fair comparison, RCC has been run on the Greyball dataset with the partition utilized for the previous experiment. Table 3.10 shows the comparison between the two methods, carried out in terms of mean and median recovery error. The proposed temporal color constancy method outperforms the RCC method in both metrics, in particular, it reaches a 77% improvement for the mean recovery error and a 85% improvement for the median recovery error.

3.3.3 Conclusions

Temporal color constancy is a new and challenging problem, that has received little attention in the state of the art. Some of the published methods are based on the assumption that the preceding frame of the shot frame contains useful information that can lead to a more accurate illuminant estimation. This work extends the assumption and proves that if the shot frame can benefit from the information stored in the preceding frames, then also those frames can benefit from the shot frame. The proposed method is composed of a backbone that extracts semantic information, and an LSTM layer, that maintains spatial recurrent information over the frames. The experiments show that an improvement up to 77% for the mean recovery error is possible with respect to the state-of-the-art. The work also proves that such results are achievable even in resource-constrained environments since MobileNetV2 has been selected as the backbone network.

Chapter 4

Conclusions

In this thesis, the primary objective has been to shed light on the complex and challenging problem of computational color constancy. The nature of this problem is ill-posed, which makes it even more challenging. Moreover, it encompasses various multidisciplinary aspects that need to be taken into account while tackling the issue. The thesis takes a comprehensive approach and delves into two possible domain extensions of the problem - multispectral and temporal. The aim is to provide a detailed understanding of the problem and propose effective solutions to overcome it.

First, the thesis offers a comprehensive insight into the advanced concepts of computational color constancy, multispectral imaging, and temporal color constancy. It elaborates on the challenges faced while dealing with these concepts, their related works, and limitations. Second, methods facing the challenges and addressing the presented limitations are proposed.

Multispectral Color Constancy One of the main contributions of this thesis is to show that multispectral imaging yields the potential to improve traditional RGB color constancy algorithms. Traditional imaging systems capture the light in three wavelength bands, allowing for a spatially accurate representation that enables humans to detect and recognize objects. However, the spectral information content is limited. The aim of chapter 2 has been to show that the additional information acquired by multispectral imaging can be used to improve the accuracy of illuminant estimation methods. Two are the approaches presented to investigate this claim. The work presented in 2.2 suggests extending a selected group of statistical white balancing methods to the multispectral domain and using the multispectral illuminant estimation to improve the traditional RGB color constancy. As shown, converting the multispectral illuminant estimation to the RGB domain is not sufficient to improve the performance of such methods. Therefore, four re-elaboration techniques are presented to adjust the multispectral illuminant estimations to better match the target illuminant after conversion. This approach and its main hypothesis have been proven to be successful. The work in section 2.3, instead, aims to investigate whether using both multispectral and color imaging can help estimate color illuminants. While multispectral sensors are becoming more affordable and lightweight, full-spatial resolution sensors are still too expensive. Thus, this work aims to combine color information, and spectral information at a low resolution. The main focus of this investigation is to evaluate the accuracy of the illuminant estimation method at different spatial resolutions, while also determining which domain (color or spectral) is better suited for the learning phase. The results show that combining color and spectral information can improve the traditional RGB color constancy problem, but higher-resolution spectral information is necessary to achieve the best results.

Temporal Color Constancy When attempting to extend color constancy to the temporal domain, a new set of challenges arises. Firstly, a temporal color constancy method must be capable of correcting all frames in a sequence, ensuring that the colors of the scene appear as if they were captured in a controlled environment. Secondly, objects and colors in these scenes must remain consistent over time and should not create any unpleasant artifacts. Section 3.2 studies the direct applicability of single-frame methods to frame sequences. To do so, it provides a common procedure to analyze the temporal and spatial stability of single-frame color constancy methods. The evaluation of temporal stability relies largely on the consistency of illuminant estimations provided by the single-frame methods over time. Thus, evaluating temporal consistency requires a suitable metric for automatic assessment. This work aims to provide the scientific community with suitable metrics, in particular, maximum illuminant change (MIC) and standard deviation (SD). The analysis highlighted a certain correlation between traditional error metrics, designed to assess the accuracy of the illuminant estimation, and the temporal consistency metrics. However, the analysis also detected some exceptions. Section 3.3, instead, provides a method for illuminant estimation in video sequences. The approach involves convolutional neural networks and Long Short-Term Memory networks (LSTM). This method also broadens the definition of temporal color constancy. Typically, temporal color constancy is used to estimate the illuminant of a selected frame by utilizing information from previous frames. However, the proposed approach expands the idea that all adjacent frames in a sequence have the potential to improve illuminant estimation, regardless of their order. This study corroborates this hypothesis and outperforms current state-of-the-art performance. Furthermore, this outcome is achieved by using a convolutional neural network specifically designed for resource-limited settings.

References

[1] Mahmoud Afifi and Michael S Brown. Sensor-independent illumination estimation for dnn models. *arXiv preprint arXiv:1912.06888*, 2019.

[2] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1535–1544, 2019.

[3] Mirko Agarla, Simone Bianco, Marco Buzzelli, Luigi Celona, and Raimondo Schettini. Fast-n-squeeze: towards real-time spectral reconstruction from RGB images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1132–1139, 2022.

[4] Vivek Agarwal, Besma R Abidi, Andreas Koschan, and Mongi A Abidi. An overview of color constancy algorithms. *Journal of Pattern Recognition Research*, 1(1):42–54, 2006.

[5] Hoda Aghaei and Brian Funt. A flying gray ball multi-illuminant image dataset for color research. In *Color and Imaging Conference*, volume 2020, pages 142–149. Society for Imaging Science and Technology, 2020.

[6] Boaz Arad, Ohad Ben-Shahar, and Radu Timofte. Ntire 2018 challenge on spectral reconstruction from RGB images. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 929–938. IEEE, 2018.

[7] Boaz Arad, Radu Timofte, Ohad Ben-Shahar, Yi-Tun Lin, and Graham D Finlayson. Ntire 2020 challenge on spectral reconstruction from an RGB image. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 446–447. IEEE/CVF, 2020.

[8] Boaz Arad, Radu Timofte, Rony Yahel, Nimrod Morag, Amir Bernat, Yuanhao Cai, Jing Lin, Zudi Lin, Haoqian Wang, Yulun Zhang, et al. Ntire 2022 spectral recovery challenge and data set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 863–881, 2022.

[9] Nikola Banić, Karlo Koščević, and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017.

[10] Simone Bianco and Claudio Cusano. Quasi-unsupervised color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12212–12221, 2019.

[11] Simone Bianco, Arcangelo R Bruna, Filippo Naccari, and Raimondo Schettini. Color correction pipeline optimization for digital cameras. *Journal of Electronic Imaging*, 22(2):023014, 2013.

[12] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using CNNs. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 81–89, 2015.

[13] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Single and multiple illuminant estimation using convolutional neural networks. *IEEE Transactions on Image Processing*, 26(9):4347–4362, 2017.

[14] JC de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.

[15] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980.

[16] James E Burt, Gerald M Barber, and David L Rigby. *Elementary statistics for geographers*. Guilford Press, 2009.

[17] Marco Buzzelli and Ilaria Erba. On the evaluation of temporal and spatial stability of color constancy algorithms. *JOSA A*, 38(9):1349–1356, 2021.

[18] Marco Buzzelli and Ilaria Erba. Temporal and Spatial Stability of Color Constancy Algorithms — Imaging and Vision Laboratory, 2021. URL `http://www.ivl.disco.unimib.it/activities/color-stability/`. `http://www.ivl.disco.unimib.it/activities/color-stability/` (Accessed on 28 July 2021).

[19] Marco Buzzelli, Simone Bianco, and Raimondo Schettini. Arc: Angle-retaining chromaticity diagram for color constancy error analysis. *J. Opt. Soc. Am. A*, 37(11): 1721–1730, Nov 2020. doi: 10.1364/JOSAA.398692.

[20] Marco Buzzelli, Riccardo Riva, Simone Bianco, and Raimondo Schettini. Consensus-driven illuminant estimation with gans. In *Thirteenth International Conference on Machine Vision*, volume 11605, page 1160520. International Society for Optics and Photonics, 2021.

[21] Marco Buzzelli, Simone Zini, Simone Bianco, Gianluigi Ciocca, Raimondo Schettini, and Mikhail K Tchobanou. Analysis of biases in automatic white balance datasets and methods. *Color Research & Application*, 2022.

[22] Marco Buzzelli, Simone Zini, Simone Bianco, Gianluigi Ciocca, Raimondo Schettini, and Mikhail K Tchobanou. Analysis of biases in automatic white balance datasets and methods. *Color Research & Application*, 48(1):40–62, 2023.

[23] Ayan Chakrabarti and Todd Zickler. Statistics of real-world hyperspectral images. In *Conference on Computer Vision and Pattern Recognition*, pages 193–200. IEEE, 2011.

[24] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5):1049–1058, 2014.

[25] Florian Ciurea and Brian Funt. A large image database for color constancy research. In *Color and Imaging Conference*, number 1, pages 160–164. Society for Imaging Science and Technology, 2003.

[26] Erfan Daneshpajooh. Drone sensors in the application of precision agriculture. 05 2021.

[27] PJ Daniell. Lectures on Cauchy's problem in linear partial differential equations. by J. Hadamard. pp. viii+ 316. 15s. net. 1923.(per Oxford University Press.). *The Mathematical Gazette*, 12(171):173–174, 1924.

[28] Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Mobile computational photography: A tour. *arXiv preprint arXiv:2102.09000*, 2021.

[29] Ilaria Erba, Marco Buzzelli, and Raimondo Schettini. RGB color constancy using multispectral pixel information. *sumbitted to JOSA A*, 2023.

[30] Ilaria Erba, Marco Buzzelli, Jean-Bapthis Thomas, Jon Hardeberg, and Raimondo Schettini. Improving RGB illuminant estimation exploiting spectral average radiance. *sumbitted to JOSA A*, 2023.

[31] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, pages 37–41. Society for Imaging Science and Technology, 2004.

[32] Graham D Finlayson and Roshanak Zakizadeh. Reproduction angular error: An improved performance metric for illuminant estimation. *Perception*, 310(1):1–26, 2014.

[33] David A Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5(1):5–35, 1990.

[34] David H Foster. Color constancy. *Vision research*, 51(7):674–700, 2011.

[35] David H Foster, Kinjiro Amano, Sérgio MC Nascimento, and Michael J Foster. Frequency of metamerism in natural scenes. *JOSA A*, 23(10):2359–2372, 2006.

[36] Yuval Garini, Ian T Young, and George McNamara. Spectral imaging: principles and applications. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, 69(8):735–747, 2006.

[37] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[38] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471, 2000.

[39] Arjan Gijsenij and Theo Gevers. Results per Dataset (Recovery error) — Color Constancy, 2011. URL `http://colorconstancy.com/evaluation/results-per-dataset/index.html#sfugreyball_linear`. `http://colorconstancy.com/evaluation/results-per-dataset/index.html#sfugreyball_linear` (Accessed on 27 July 2021).

[40] Arjan Gijsenij, Theo Gevers, and Marcel P Lucassen. Perceptual analysis of distance measures for color constancy algorithms. *JOSA A*, 26(10):2243–2256, 2009.

[41] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Computational color constancy: Survey and experiments. *IEEE transactions on image processing*, 20(9): 2475–2489, 2011.

[42] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Improving color constancy by photometric edge weighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):918–929, 2011.

[43] Han Gong. Convolutional mean: A simple convolutional neural network for illuminant estimation. *arXiv preprint arXiv:2001.04911*, 2020.

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[45] John Guild. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 230(681-693):149–187, 1931.

[46] Xiangpeng Hao and Brian Funt. A multi-illuminant synthetic image test set. *Color Research & Application*, 45(6):1055–1066, 2020.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[48] Sepp Hochreiter and Jürgen Schmidhuber. LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, 9, 1996.

[49] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[50] Steven Hordley, Graham Finalyson, and Peter Morovic. A multi-spectral image database and its application to image rendering across illumination. In *International Conference on Image and Graphics*, pages 394–397. IEEE, 2004.

[51] Steven D Hordley and Graham D Finlayson. Reevaluation of color constancy algorithm performance. *JOSA A*, 23(5):1008–1020, 2006.

[52] Steven D Hordley and Graham D Finlayson. Reevaluation of color constancy algorithm performance. *JOSA A*, 23(5):1008–1020, 2006.

[53] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[54] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4085–4094, 2017.

[55] Po-Chieh Hung and Zhen Zhang. Electronic devices with color compensation, April 19 2022. US Patent 11,308,846.

[56] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[57] Haris Ahmad Khan. *Multispectral constancy for illuminant invariant representation of multispectral images.* PhD thesis, Université Bourgogne Franche-Comté; Norwegian University of Science and . . . , 2018.

[58] Haris Ahmad Khan, Jean Baptiste Thomas, and Jon Yngve Hardeberg. Multispectral constancy based on spectral adaptation transform. In *Image Analysis: 20th Scandinavian Conference, SCIA 2017, Tromsø, Norway, June 12–14, 2017, Proceedings, Part II 20*, pages 459–470. Springer, 2017.

[59] Haris Ahmad Khan, Jean-Baptiste Thomas, Jon Yngve Hardeberg, and Olivier Laligant. Illuminant estimation in multispectral imaging. *JOSA A*, 34(7):1085–1098, 2017.

[60] Haris Ahmad Khan, Jean-Baptiste Thomas, and Jon Yngve Hardeberg. Towards highlight based illuminant estimation in multispectral images. In *Image and Signal Processing: 8th International Conference, ICISP 2018, Cherbourg, France, July 2-4, 2018, Proceedings 8*, pages 517–525. Springer, 2018.

[61] Haris Ahmad Khan, Jean-Baptiste Thomas, Jon Yngve Hardeberg, and Olivier Laligant. Spectral adaptation transform for multispectral constancy. *Journal of Imaging Science and Technology*, 62(2):20504–1, 2018.

[62] Vlado Kitanovski, Jean-Baptiste Thomas, and Jon Yngve Hardeberg. Reflectance estimation from snapshot multispectral images captured under unknown illumination. In *Color and Imaging Conference 29*, pages 264–269. Society for Imaging Science and Technology, 2021.

[63] Samu Koskinen, Erman Acar, and Joni-Kristian Kämäräinen. Single pixel spectral color constancy. In *The 32nd British Machine Vision Conference*, 2021.

[64] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.

[65] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018.

[66] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971.

[67] Reiner Lenz, Peter Meer, and Markku Hauta-Kasari. Spectral-based illumination estimation and color correction. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 24(2):98–111, 1999.

[68] Shuwei Li, Jikai Wang, Michael S Brown, and Robby T Tan. Transcc: Transformer-based multiple illuminant color constancy using multitask learning. *arXiv preprint arXiv:2211.08772*, 2022.

[69] Yuqi Li, Qiang Fu, and Wolfgang Heidrich. Multispectral illumination estimation using deep unrolling network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2672–2681, 2021.

[70] Shili Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, 2010.

[71] Yi-Tun Lin and Graham D Finlayson. Investigating the upper-bound performance of sparse-coding-based spectral reconstruction from RGB images. In *Color and Imaging Conference*, volume 2021, pages 19–24, 2021.

[72] Yi-Tun Lin and Graham D Finlayson. An investigation on worst-case spectral reconstruction from RGB images via radiance mondrian world assumption. *Color Research & Application*, 48(2):230–242, 2023.

[73] Yi-Tun Lin and Graham D Finlayson. A rehabilitation of pixel-based spectral reconstruction from RGB images. *Sensors*, 23(8):4155, 2023.

[74] Changhong Liu, Wei Liu, Xuzhong Lu, Fei Ma, Wei Chen, Jianbo Yang, and Lei Zheng. Application of multispectral imaging to determine quality attributes and ripeness stage in strawberry fruit. *PloS one*, 9(2):e87818, 2014.

[75] Alexander D Logvinenko, Brian Funt, Hamidreza Mirzaei, and Rumi Tokunaga. Rethinking colour constancy. *PLoS One*, 10(9):e0135029, 2015.

[76] Yoshitsugu Manabe, Kosuke Sato, and Seiji Inokuchi. An object recognition through continuous spectral images. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 858–860. IEEE, 1994.

[77] Yuri Murakami, Masahiro Yamaguchi, and Nagaaki Ohyama. Hybrid-resolution multispectral imaging using color filter array. *Optics express*, 20(7):7173–7183, 2012.

[78] Yuri Murakami, Keiichiro Nakazaki, and Masahiro Yamaguchi. Hybrid-resolution spectral video system using low-resolution spectral sensor. *Optics Express*, 22(17):20311–20325, 2014.

[79] Keiichiro Nakazaki, Yuri Murakami, and Masahiro Yamaguchi. Hybrid-resolution spectral imaging system using adaptive regression-based reconstruction. In *Image and Signal Processing: 6th International Conference, ICISP 2014, Cherbourg, France, June 30–July 2, 2014. Proceedings 6*, pages 142–150. Springer, 2014.

[80] Sérgio MC Nascimento, Flávio P Ferreira, and David H Foster. Statistics of spatial cone-excitation ratios in natural scenes. *JOSA A*, 19(8):1484–1490, 2002.

[81] Sérgio MC Nascimento, Kinjiro Amano, and David H Foster. Spatial distributions of local illumination color in natural scenes. *Elsevier Vision research*, 120:39–44, 2016.

[82] Rang MH Nguyen, Dilip K Prasad, and Michael S Brown. Training-based spectral reconstruction from a single RGB image. In *European Conference on Computer Vision*, pages 186–201. Springer, 2014.

[83] Rang MH Nguyen, Dilip K Prasad, and Michael S Brown. Training-based spectral reconstruction from a single RGB image. In *European Conference on Computer Vision*, pages 186–201. Springer, 2014.

[84] Manu Parmar, Francisco Imai, Sung Ho Park, and Joyce Farrell. A database of high dynamic range visible and near-infrared multispectral images. In *Digital photography iv*, volume 6817, page 68170N. International Society for Optics and Photonics, 2008.

[85] Veronique Prinet, Dani Lischinski, and Michael Werman. Illuminant chromaticity from image sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3320–3327, 2013.

[86] Yanlin Qian, Ke Chen, Jarno Nikkanen, Joni-Kristian Kamarainen, and Jiri Matas. Recurrent color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5458–5466, 2017.

[87] Yanlin Qian, Joni-Kristian Kamarainen, Jarno Nikkanen, and Jiri Matas. On finding gray pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8062–8070, 2019.

[88] Yanlin Qian, Jani Käpylä, Joni-Kristian Kämäräinen, Samu Koskinen, and Jiri Matas. A benchmark for burst color constancy. In *European Conference on Computer Vision*, pages 359–375. Springer, 2020.

[89] Antonio Robles-Kelly and Ran Wei. A convolutional neural network for pixelwise illuminant recovery in colour and spectral images. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 109–114. IEEE, 2018.

[90] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[91] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[92] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[93] Armin Schneider and Hubertus Feussner. *Biomedical engineering in gastrointestinal surgery*. Academic Press, 2017.

[94] Steven A Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985.

[95] Torbjørn Skauli and Joyce Farrell. A collection of hyperspectral images for imaging systems research. In *Digital Photography IX*, volume 8660, page 86600C. International Society for Optics and Photonics, 2013.

[96] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

[97] Tong Su, Yu Zhou, Yao Yu, Xun Cao, and Sidan Du. Illumination separation of non-lambertian scenes from a single hyperspectral image. *Optics express*, 26(20): 26167–26178, 2018.

[98] Tong Su, Yu Zhou, Yao Yu, Xun Cao, and Sidan Du. Illumination separation of non-lambertian scenes from a single hyperspectral image. *Optics express*, 26(20): 26167–26178, 2018.

[99] S Unninayar and L Olsen. Monitoring, observations, and remote sensing–global dimensions. 2008.

[100] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on image processing*, 16(9):2207–2214, 2007.

[101] J Von Kries. Theoretische studien über die umstimmung des sehorgans. *Festschrift der Albrecht-Ludwigs-Universität*, 32:145–158, 1902.

[102] Ning Wang, Brian Funt, Congyan Lang, and De Xu. Video-based illumination estimation. In *Computational Color Imaging: Third International Workshop, CCIW 2011, Milan, Italy, April 20-21, 2011. Proceedings 3*, pages 188–198. Springer, 2011.

[103] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

[104] Susan Wojcicki. YouTube at 15: My personal journey and the road ahead, 2020. URL `https://blog.youtube/news-and-events/youtube-at-15-my-personal-journey`. `https://blog.youtube/news-and-events/youtube-at-15-my-personal-journey` (Accessed on 25 May 2021).

[105] Jin Xu, Zishan Li, Bowen Du, Miaomiao Zhang, and Jing Liu. Reluplex made more practical: Leaky relu. In *2020 IEEE Symposium on Computers and communications (ISCC)*, pages 1–7. IEEE, 2020.

[106] Jyoti Yadav and Monika Sharma. A review of k-mean algorithm. *Int. J. Eng. Trends Technol*, 4(7):2972–2976, 2013.

[107] Qingxiong Yang, Shengnan Wang, Narendra Ahuja, and Ruigang Yang. A uniform framework for estimating illumination chromaticity, correspondence, and specular reflection. *IEEE Transactions on Image Processing*, 20(1):53–63, 2010.

[108] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.

[109] Thomas Young. Ii. the bakerian lecture. on the theory of light and colours. *Philosophical transactions of the Royal Society of London*, (92):12–48, 1802.

[110] Yinqiang Zheng, Imari Sato, and Yoichi Sato. Illumination and reflectance spectra separation of a hyperspectral image meets low-rank matrix factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1779–1787, 2015.

[111] Hong Bo Zhou and Jun Tao Gao. Automatic method for determining cluster number based on silhouette coefficient. *Advanced materials research*, 951:227–230, 2014.

[112] Simone Zini, Marco Buzzelli, Simone Bianco, and Raimondo Schettini. COCOA: Combining Color Constancy Algorithms for Images and Videos. *IEEE Transactions on Computational Imaging*, 8:795–807, 2022.