Revisiting and Evaluating

Colour Constancy and Colour Stabilisation

Algorithms

Ghalia Hemrit

School of Computing Sciences

University of East Anglia

This dissertation is submitted for the degree of

Doctor of Philosophy

September, 2020

Colour is all. When colour is right, form is right. Colour is everything, colour is vibration like music; everything is vibration.

Adam and Eve expelled from Paradise, 1961.
Marc Chagall

Abstract

When we capture a scene with a digital camera, the sensor generates a digital response which is the Raw image. This response depends on the ambient light, the object reflectance and the sensitivity of the camera. The generated image is processed with the the camera pipeline, which is a series of operations aiming at processing the colours of the image to make it more pleasant for the user. Further colour processing can also be performed on the pipeline output image. This said, processing the colours is not only important for aesthetic reasons, but also for various computer vision tasks where a faithful reproduction of the scene colours is needed e.g. for object recognition and tracking. In this thesis, we focus on two important colour processing operations: colour constancy and colour stabilisation.

Colour constancy is the ability of a visual system to see an object with the same colour independently of the light colour; the camera processes the image so the scene looks like captured under a canonical light, usually a white light. This means that when we take two images of, let's say, a green apple in the sunlight and indoor under a tungsten light, we want the apple to appear green in both cases. To do that one important step of the pipeline is to estimate the light colour in the scene to then discount it from the image.

In this thesis we first focus on the illuminant estimation problem, in particular on the performance evaluation of illuminant estimation algorithms on the benchmark ColorChecker dataset. More precisely, we show the importance of the accuracy of the ground-truth illuminants when evaluating algorithms and comparing them.

The following part of the thesis is about chromagenic illuminant estimation which is based on using two images of the scene: one filtered and one unfiltered where the two images need to be registered. We revisit the preprocessing step (colour correction) of the chromagenic method and we introduce the use of the Monge-Kantorovitch transform (MKT) that removes the need for the expensive registration task. We also introduce two new datasets of chromagenic images for the evaluation of illuminant estimation methods.

The last part of the thesis is about colour stabilisation which is particularly important in video processing, where consistency of colours is required across image frames. When the camera moves or when the shooting parameters change, the same object in the scene can appear with different colours in two consecutive frames. To solve for colour stabilisation given a pair of images of the same scene we need to process the first image to match the second. We propose using MKT to find the mapping. Our novel method gives competitive results compared to other recent methods while being less computationally expensive.

Acknowledgements

I have been privileged to work with Professor Graham Finlayson, Dr Michal Mackiewicz and Dr Mark Fisher as my supervisory team. Thank you for the opportunity that you offered me to be part of the Colour Lab and for your guidance and support throughout this great journey. Thank you to my examiners, Dr Maria Vanrell and Professor Richard Harvey for your recommendations, your time and valuable feedback on my thesis.

Thank you to the people I worked with from academia and industry: to Professor Mark Drew for your help with different research questions, to Professor Ronnier Luo for the inspiring conversations, and to Dr Javier Vazquez-Corral for the research collaboration and your support with my research. Thank you to Professor Tsumura's Lab team for our research collaboration and for the warm welcome during my visit in Japan. I would also like to thank my coauthors for the valuable discussions that we had on various research questions.

Thanks to my colleagues and fellow PhD researchers from the Colour Lab, starting with Dr Han Gong and Goeff French, thank you for your valuable support and kindness, and the other students and friends: Fufu, Yuteng, Ellie, Seth and Ethan among others, with whom I have great memories and enjoyed very much sharing the office and the PhD campus life. Thanks to my friends in Norwich and outside, in the UK, France, Tunisia and other places who brought me a lot of joy during the PhD journey and have always encouraged me even when living far and despite the challenges we all faced in particular during the pandemic.

My parents have always been supportive. I thank them very much for everything and most of all, my gratitude is to Didoun, my loving grandfather, whose memory will live on.

Table of contents

List of figures

List of tables

Publications

The following are publications related to this work by the author:

- Finlayson, G. D., Hemrit, G., Gijsenij, A., Gehler, P. V., "A Curious problem with using the colour checker dataset for illuminant estimation," Color and Imaging Conference, 64-69, 2017.
- Hemrit, G., Finlayson, G. D., Gijsenij, A., Gehler, P. V., Bianco, S., Funt, B., Shi, L., "Rehabilitating the ColorChecker dataset for illuminant estimation," Color and Imaging Conference, 350–353, 2018.
- Hemrit, G., Matsushita, F., Uchida, M., Vazquez-Corral, J., Gong, H., Tsumura, N., Finlayson, G. D., "Using the Monge-Kantorovitch Transform in Chromagenic Color Constancy for Pathophysiology," Computational Color Imaging Workshop, 121–133, 2019.
- Hemrit, G., Finlayson, G. D., Gijsenij, A., Gehler, P. V., Bianco, S., Drew, M. S., Funt, B., Shi, L., "Providing a Single Ground-truth for Illuminant Estimation for the ColorChecker Dataset," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1286 - 1287, 2019.

Chapter 1

Introduction

The largest focus of this thesis is illuminant estimation and in particular understanding how algorithms have been and should be evaluated. To be more precise we focus on the most widely used dataset for evaluating and ranking illuminant estimation algorithms - the ColorChecker Dataset - and find that almost all prior art use this data wrongly. We find that there are (at least) 3 different ground-truth illuminants sets. We called these ground-truth sets SFU, Gt1 and Gt2. In Chapter 3 we show the distributions of the ground-truth sets in Figure 3.2 which proves that they can be significantly different from one another, and yet authors are not aware of this fact. So, one author will say algorithm A is better than B, but they are using, let's say, the SFU ground-truth. Another author will say the converse but they are using the Gt1 ground-truth.

The first contribution we make is understanding this problem, its magnitude (large) and how it came about. My second contribution is to provide a new single (definitive and correct) ground-truth for the ColorChecker dataset by reprocessing the Raw images of the dataset and using the ground-truth calculation methodology described by Shi and Funt.

At this point in my research I became interested in developing a new illuminant estimation algorithm. I was particularly interested in the chromagenic approach which uses 2 images of the same scene (with and without a coloured filter being placed in front of the camera) and the premise is that the relationship between the colours across the two images helps identify the colour of the light. The method has a calibration step where given pairs of images, we calculate the transform that maps every filtered image to its unfiltered counterpart. Every transform is associated to an illuminant colour vector. To estimate the illuminant with the chromagenic method for a scene, given 2 images of the scene , one filtered and one unfirltered, the best mapping that describes the relationship between the 2 images is found which gives the estimate illuminant which is the related colours vector. Interestingly the prior art chromagenic algorithms could work well [11] but they often performed poorly as well, especially on real scenes images.

In my own research into this problem, I quickly understood a key reason why chromagenic might have problems. Fundamentally it assumes that the filtered and unfiltered images are registered. This is not the case and even after running a registration algorithm the two images are not and cannot be in pixel-wise registration. So, I became interested in the problem of mapping colours between images when registration is not possible.

This led to my 3rd and 4th contributions. Namely, I proposed to use the Monge-Kantorovitch (linear restriction) transform (MKT) [12] to match colours and solve for the pre-processing step of chromagenic illuminant estimation.

I evaluated this idea on a new chromagenic dataset which I created. MKT works well (better than assuming registration) as we show e.g. in Figure 5.8. So MKT discounts the need for an expensive registration operation. This work was also placed in the context of face images (captured for the purpose of Kampo medical diagnosis).

Immediately, we saw there was an opportunity to apply the same MKT methodology to help in the video colour stabilisation problem. Here, frame to frame the same scene colours can flicker in the corresponding images. This is because frame to frame the camera renders the images differently with different capture parameters or viewing angle. Here, we successfully simplified a state-of-the-art colour stabilisation algorithm based on the costly RANSAC operation by incorporating MKT which removes the need for using RANSAC and gives competitive results (e.g. Figures 6.7).

However, this detour into colour stabilisation meant that we did not actually develop a new chromagenic algorithm. However, we are confident that both the new dataset and the MKT approach will be useful for future research.

Chapter 2

Background

2.1 Colour Image Formation Model

The colour response depends on the spectral power distribution (SPD) of the illuminant (which determines how much energy the source emits at each wavelength λ of the electromagnetic spectrum), the scene objects reflectances and the visual system sensitivity function. Besides it also depends on a geometry factor which is the position of the visual system from the object (view direction and surface orientation).

The reflectance defines the proportion of the incident light that the object reflects. The reflected light direction and composition depend on the object surface and body properties, on the spectral composition of the light and the geometry factor. For instance, a white matte surface reflects the incident light equally for all wavelengths.

There exist different reflectance models to describe the formation of the colour response of a visual system [13][14]. Under the Lambertian assumption –which means that the object is a Lambertian diffuser and reflects the incident light equally in all directions– the response is independent of the geometry factor. The diffuse reflectance is called albedo as opposed to the specular reflection in one direction.

A Lambertian surface with reflectance $S(\lambda)$ illumiated by an SPD $E(\lambda)$, reflects the colour signal $C(\lambda)$ of wavelength λ such as :

$$C(\lambda) = S(\lambda)E(\lambda) \tag{2.1}$$

2.2 Colour Constancy

Colour constancy describes the ability of a visual system to see an object with a roughly constant colour regardless of the scene light colour. The human visual system has a high degree of colour constancy [15] (also known as chromatic adaptation) where the colour perception of an object also depends on its surrounding (local and global contrast) in the perceived scene [16].

In digital cameras and in the camera industry in general, this capability or processing task of the camera is often called 'white balance', referring to the ability of the system to correct the objects colours in the image by removing the cast of a reference light colour. If the captured scene is, let's say, under a yellow light then all the image RGBs are biased in the yellow direction. A white surface will look slightly yellowish. If we divide all the RGBs by the RGB for a white surface (in this case a yellowish response vector) we can remove the yellow cast from the image. In summary if we balance the response for a white to look 'right' then the other colours are also - more or less - correctly balanced.

The camera does not measure the light colour of the scene. This is why the reference light is in practice not the exact light in the scene but the type of light chosen in the camera parameters (day light, sky light, etc) or, in the case of a typical automatic white balance (AWB) mode in DSLR (Digital Single Lens Reflex) cameras or smartphone cameras, it is a standard D65 light (emission of a black body at about 6500K which corresponds to an average daylight). D65 is often the reference light in digital photography.

2.2.1 Digital Cameras Response

While the human visual system can only see over the visible spectrum, ranging roughly from 400 to 700 nm, a digital camera can be sensitive to a larger range of the spectrum depending on its sensor bandwidth, for example, Infrared-sensitive (IR) digital cameras. The majority of digital cameras however have a hot mirror which allows only the visible light to pass.

The camera response to the colour signal reflected by a surface in the scene for the wavelength λ (Equation 2.1) depends on the sensitivity function $Q(\lambda)$ of the camera

$$
\begin{aligned}
\rho(\lambda) = & \quad C(\lambda)Q(\lambda) \\
= & \quad E(\lambda)S(\lambda)Q(\lambda)
\end{aligned}
\tag{2.2}
$$

where $E(\lambda)$ is the SPD of the ambient light and $S(\lambda)$ is the surface reflectance.

Then we can write the Lambertian colour response ρ of a camera for the same surface of the scene, over the visible spectrum Ω as follows :

$$\rho = \int_{\Omega} E(\lambda) S(\lambda) Q(\lambda) d\lambda \tag{2.3}$$

We consider that all the surfaces in the scene receive the same light with the same spectral composition, i.e. $E(\lambda)$ does not depend on the scene surface.

The visible spectrum Ω can be sampled from the continuous domain to a $N = 31$ vectors $(400 : 10 : 700nm)$, therefore the integral from the equation becomes a summation.

$$\rho = \sum_{i=1}^{N} E_i S_i Q_i \tag{2.4}$$

It is useful to rewrite Equation 2.3 based on an integral in the language of linear algebra (specifically making an integral a dot-product). We use the RGB representation as most modern camera have three distinct classes of sensors so the response to light is defined by a triplet of responses in the red, green and blue parts of the spectrum.

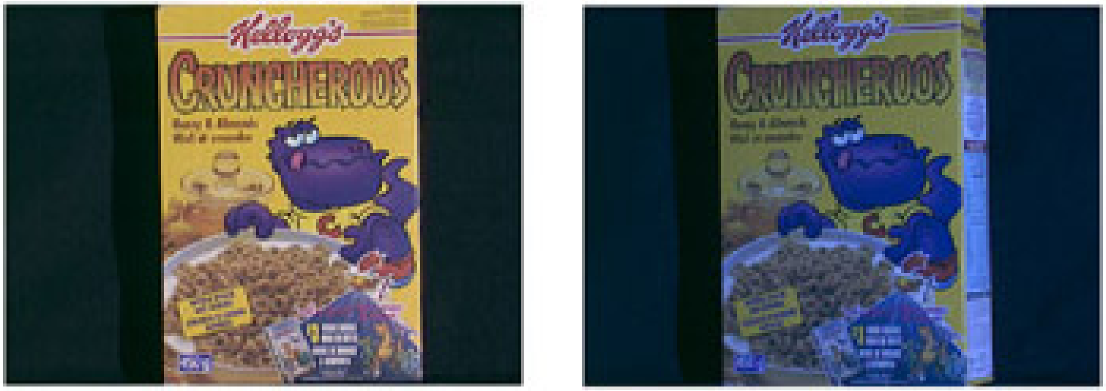


Fig. 2.1 The same scene under two different colour lights from the Barnard's dataset [1].

The definition of a camera response proves that the role of the illuminant is as important as the role of the surface reflectance. In Figure 2.1 we show the picture of Cruncheroos box under two different colour lights where for this example the camera white-balance algorithm is switched off. Of course we do not see the same object as having different colours when the light changes. If we saw the two boxes under the yellow and blue lights our visual system would discount the yellowness and blueness. The difference we see in Figure 2.1 would appear to be much less. Our visual system discounts the colour of the light.

2.2.2 Colour Constancy in Digital Cameras

Performing colour constancy is not only important for aesthetic reasons in digital photography but also to solve for fundamental computer vision tasks in which objects need to be reliably identified by their colours, e.g. object recognition [17], object tracking [18] and scene understanding [19] where the task at hand might fail if the illumination is changing in the scene.

Given an image taken under a certain illuminant the aim of colour constancy is to convert the colours in the image by discounting the colour bias created by the scene illuminant to create colours that look 'natural' as if the image were captured under a white light. Very often, it is supposed there is a single uniform illuminant in the scene.

In the context of a digital visual system, computational colour constancy can be performed following one of these two approaches: either by first estimating the illuminant colour and then discounting it from the image colours or by a colour invariant approach as discussed in [20][21] which derive quantities which are invariant to the colour. In what follows we focus on the first approach: the illuminant estimation-based computational colour constancy.

In this case, colour constancy is performed in two steps. First, the illuminant colour is estimated in some way and then this colour vector is 'divided out' from the image. The 'dividing out' approach is analogous to the von Kries adaptation [22].

In Equation 2.5 we map a colour $\underline{\rho_o}$ viewed under an unknown light to the RGB $\underline{\rho_c}$ for the same surface viewed under a reference light. The mapping is a diagonal matrix $\mathbf{D_c}$. Mathematically, the diagonal matrix is implementing 'dividing out' by the estimate light and then mapping to the reference light. Crucially, the same diagonal matrix is applied to all image RGBs. This formalism is widely deployed, including [23][24][25]. One advantage of the 'diagonal approach' is that it, explicitly, teaches that illuminant estimation is a 3-dimensional problem. Rather than using a diagonal matrix, a 3×3 transform matrix could be used.

However, this means that 9 parameters have to be 'solved for', and this makes the estimation a more complex problem [26].

Rather than using the integral formulation for the colour response let us denote the RGB response of the ith surface under light $\underline{E_c}$ as:

$$\underline{\rho_c} = \mathbf{D_c}\underline{\rho_o} \tag{2.5}$$

where $\underline{\rho_o}$ denotes the response of the same surface viewed under light $\underline{E_o}$. The 3×3 diagonal matrix takes colours from light $\underline{E_o}$ to the reference light $\underline{E_c}$ (e.g. D65). Remembering our earlier discussion about what white-balance means in camera, the matrix $\mathbf{D_c}$ is calculated as follows:

$$\mathbf{D}_c = diag(\frac{\underline{\rho}_c}{\underline{\rho}_o}) \tag{2.6}$$

where we component-wise divide the vectors $\underline{\rho}_o$ and $\underline{\rho}_c$ which respectively denote the RGB response of a white surface viewed under lights $\underline{E_o}$ and $\underline{E_c}$. The function $diag()$ maps a vector to a diagonal matrix. The model so-formulated works and is discussed in[27]. We consider one single illuminant in the scene, for this reason, the transform is the same for all the scene surfaces, as opposed to the multi-illuminant case.

Digital cameras have both preset and dynamic means of illuminant estimation. Presetting the illuminant might be problematic. One example is when the camera is balanced for a given illuminant e.g. skylight and the object is lit by another illuminant, e.g. a tungsten light, this will skew the colour reproduction towards the red (in the direction of the illuminant). When the white balance mode is not preset in a camera, white-balancing the image relies on an illuminant estimation algorithm. We present in the next section some of the most commonly used and recent algorithms.

2.3 Illuminant Estimation Algorithms

There are two main large categories of illuminant estimation algorithms: algorithms that estimate the illuminant via a 'bag of pixels', called the statistical approaches like [28][29][30][31], and the learning-based methods [32][33], including deep learning [34][35][36][37][2][38]. Another important category includes the algorithms that are based on a calibration stage like the colour by correlation [39] and the gamut mapping [40] methods. There are other less commonly used methods that look for physical insights to drive the light estimation. For example, in the specular highlight method [41], highlights are sought in the scene. It is then assumed that the highlight colour is the same as the illuminant colour (true for dielectic materials). Other examples are [42][13][43][44]. The chromagenic colour constancy method [45][46][47][48] is another physics-based method which uses the illuminant dependant relationship between the filtered and unfiltered images of the same scene to estimate the light colour.

The majority of the methods solve for a single estimate and in the recent years, various works were published about producing multiple estimates for multi-illuminants images. In [49][50] the user preference and guidance are used to respectively make a final decision of a global estimate and a light mixture given two estimates.

2.3.1 Statistical Methods

One of the most used illuminant estimation algorithm which is Grey-World [29] is also one of the simplest. It assumes that the average of the scenes surfaces is a grey surface (defined reflectance). As a consequence the colour of the illuminant is derived from the average colour in the image.

Another commonly used illuminant estimation method is White-Patch [28]. It assumes that the captured scene always contains a white patch whose reflectance is maximum. The pixel value with the maximum intensity in the image is assumed to be a reproduction of a white surface under the unknown illuminant.

Although these assumptions can work surprisingly well they often fail. In fact, they are restrictive and very often violated for real-world scenes. Finlayson and Trezzi introduced the Shades of Grey method [30]. This method generalises the previous ones by supposing that the scene average is some shade of gray. The estimated illuminant is expressed as follows for the channel k:

$$E_i = \frac{\left(\sum_{i=1}^{N} \rho_i^p\right)^{\frac{1}{p}}}{N^{\frac{1}{p}}} \quad ; \quad i \in \{r,g,b\} \quad and \quad p \geq 1 \tag{2.7}$$

where E_k is is the r, g or b value of the illuminant colour $\underline{E_{est}}$, p is the Minkowski norm, N is the number of pixels in the image and ρ_i is the r, g or b value of the ith pixel (We group all the pixels in a vector). When $p = 1$, Equation 2.7 corresponds to the Grey-World assumption and when $p = \infty$ it corresponds to Max-RGB.

This idea was further extended to the derivative domain by Van de Weijer et al. In [54] they unified various algorithms and proposed the Grey-Edge hypothesis, in which the illuminant is expressed with an integral over the image in Equation 2.8.

$$\underline{E}_{est}(n,p,\sigma) = \frac{1}{\alpha}\left(\int_I |\nabla^n \rho_\sigma|^p \, dI\right)^{\frac{1}{p}} \tag{2.8}$$

where the integral is over the domain of the image I, ∇ is the partial derivative operator, n is the order of the derivative, p is the Minkowski norm, α is a constant to be chosen such that the illuminant colour $\underline{E_{est}}$ has unit length (using the L_2 norm), $\rho_\sigma = \rho \otimes \mathbf{G}_\sigma$ is the convolution of the image with a Gaussian filter $\mathbf{G}_\sigma$ with a scale parameter σ and $||$ denotes the absolute value.

2.3.2 Learning-based Methods

The most recent algorithms of illuminant estimation using a deep neural network usually outperform most statistics and physics-based methods on the benchmark datasets (see colorconstancy.com [55], a widely used comparison site for illuminant estimation research hosting data and results).

A learning-based method has two steps: a learning step and a test step. First, a model is learnt from a set of features extracted from the training set. During the training, a cost function is minimised to have the smallest loss (error) between the estimate and the ground-truth (called the 'target' variable) for every sample from the training set. The choice of the cost function is important and often it is the angular error in the illuminant estimation case. Once the model is learnt, it can be tested on a new set of images.

Many of the learnt models are built upon handcrafted features (shapes, colours, edges, etc) extracted from the training set of images, e.g. in [56–60, 32, 61–63]. This requires complex tasks of image processing and data representation. More recently convolutional neural networks (CNN) have been used to learn the features, e.g. in [34, 64, 37, 35, 36, 2, 38]. In fact, the different neurons are capable of identifying different features in the image from highlights to edges and semantic content and complex textures. Deeper layers are usually devoted to more complex objects as demonstrated in the survey on neurons selectivity by Rafegas et al. [65].

In colour constancy, the available datasets are often not large enough to train the network. For this reason a data augmentation method (including rotating, flipping, etc) is added to the training as a technique to generate more data and prevent the over-fitting of the model [66, 67]. Another practice is using a synthetically generated dataset like in [64] by changing the illuminants in the training images [68].

One important difference between the recent deep neural networks-based methods is the sub-sampling of the data. The Convolutional Colour Constancy (CCC) algorithm by Bianco et al. [34] takes as inputs small patches. The challenge with the small patches is that some of them do not carry any semantic information and therefore they represent a noisy data for the training. Shi et al. [37] improved the patches-based learning using by proposing a novel architecture of a Deep Specialised Network DS-Net which consists of two interacting CNNs.

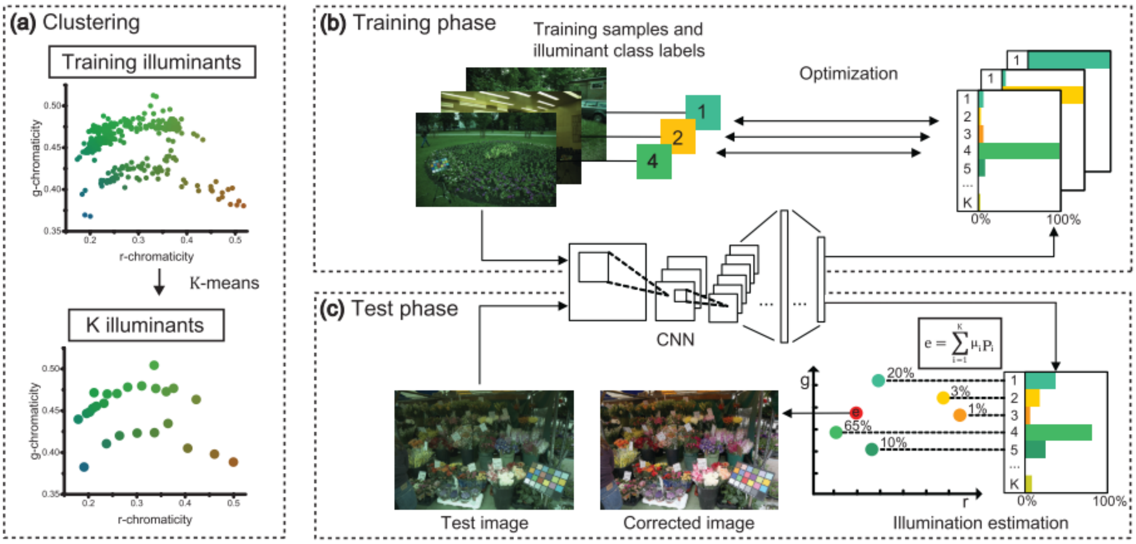


Fig. 2.2 Overview of the classification deep learning method of [2].

Another interesting patch-based approach is the CNN-based classification method by Oh and Kim [2], see Figure 2.2. The method gives a single estimate of the illuminant using pre-defined clusters of ground-truth illuminants. The method has a pre-processing step where the ground-truth illuminants used for the training are clustered using K-means to make the classification easier. The network is trained with labelled input images and learns to predict the probability (weight) of a new image belonging to each cluster. The estimate is a weighted average of the centroids of the pre-defined clusters:

$$\underline{E}_{est} = \sum_{i=1}^{N} \underline{\mu}_i p_i \tag{2.9}$$

where $\underline{\mu}_i$ is the ith cluster centroid and p_i is the probability of the image belonging to the ith cluster. The CNN network has five feature extraction layers of the convolution and the max-pooling, followed by three fully-connected layers for classification, see Figure 2.3. The cost is a multinomial negative log-likelihood function. The network takes 227 pixels inputs. The training images are cropped at random positions at the cropped patches are re-scaled to the CNN input size. The FC4 method, proposed by Hu et al. [38], uses instead full-resolution images.

In general it is difficult to train a model that generalises well. Models are dataset-dependant. This is why the learning-based algorithms often fail when the test images differ very much from the images used for training. For this reason, it is necessary to retrain each network

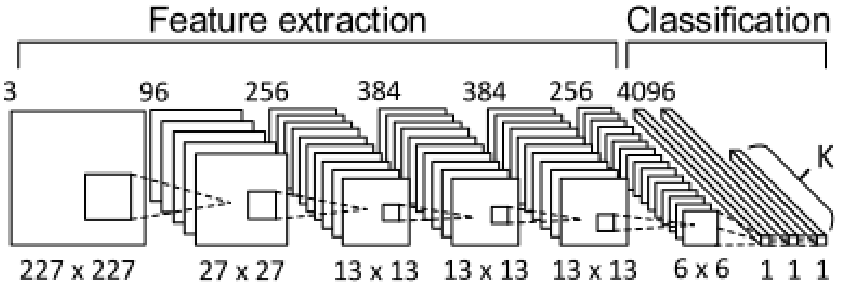


Fig. 2.3 Architecture of the classification method network of [2].

for every new application. One example is AlexNet [69] introduced initially for image classification which was re-trained later for illuminant estimation by Hu et al. [38]. In fact for illuminant estimation, we aim at estimating the illuminant regardless of the scene content while for image classification, we need to produce an estimate regardless of the scene illuminant.

2.4 Algorithms Performance Evaluation

When a new illuminant estimation algorithm is introduced, it is important to know how well it is performing compared to other recent and state of the art methods. To do so the authors evaluate the new algorithm on one or multiple benchmark datasets of images and it can be compared and ranked to the other methods.

2.4.1 Benchmark Datasets

To evaluate the performance of an illuminant estimation algorithm, the algorithm is tested on a set of images. Every image has a ground-truth or a correct answer of the illuminant colour. The ground-truth is often an RGB 3–vector but it can also be a ground-truth map where the illuminant colour is measured for every pixel of the image, see e.g. [70]. Sometimes a spectroradiometer is used to measure the scene light characteristics. In this case, the camera spectral sensitivities need to be known so this approach is rarely used.

There exist multiple benchmark datasets and since 2012, an average of one dataset per year has been introduced [71]. What is important about a dataset for algorithms evaluation is the size of the dataset (learning methods need large datasets for training to prevent over-fitting), the number of cameras used (having multiple cameras allows testing the stability of the algorithm to the sensor changes) and the presence of a colour target.

Often the images have a colour target, also called a calibration object. The ground-truth illuminant RGB is defined, let's say, from the white surface of the colour target. Examples of these colour targets are the colour chart or the grey card used to calculate the illuminant colour. The colour targets are useful for other applications like colour correction (see Chapter 5), colour stabilisation (Chapter 6) or the mapping to the XYZ colour space. Very often in illuminant estimation flat targets are used. The most popular is the Macbeth ColorChecker chart [72] used in the images from [32][73][74][75]. The IT 8 by Wolf Faust [76] is another colour chart. The illuminant of a flat colour target is sometimes not consistent with the illuminant of the observed scene. For example, for an outside scene, in the presence of shadow and sunlight, the flat target can measure either the shadow or the sunlight but not both depending on where it is placed in the scene. Cuirea et al. [77] and Bani et al. [78] tried to address this problem by using respectively for the dataset collection a grey sphere and a SpyderCube [79].

The ColorChecker dataset was introduced in 2008 by Gehler et al.[32]. It has 568 RGB images taken with two cameras. Every scene has a ColorChecker chart in it which gives the ground-truth illuminant colour. The dataset is described in more details in the next Chapter 3 as well as the calculation methodology of its ground-truth. We re-evaluate in the next chapter the performance of illuminant estimation algorithms on this dataset. Samples from the dataset are presented in Figure 3.1.

The NUS dataset, introduced in 2014 by Cheng et al. [75] is similar in nature to the ColorChecker dataset but it has more images. The same scenes were captured with 9 different commercial cameras (about 200 scenes by each camera). A Macbeth ColorChecker is placed in every scene. The scenes represent indoor and outdoor sceneries and have various lighting conditions.

The SFU Greyball dataset[77] has 11000 images of indoor and outdoor scenes taken with a Sony VX-2000 DSLR video camera. The scenes were captured in two locations: Vancouver in Canada and Scottsdale in Arizona. The video camera generated 2 hours of video which corresponds to 15 unique frames per second. At most 3 frames per second were kept to have the images different enough. The ground-truth illuminant was measured using a 4.8 cm smooth grey sphere connected to the video camera. The eponymous grey ball appears in every scene and is placed in the predominant light part of the scene. Samples from the dataset are presented in Figure 2.4.

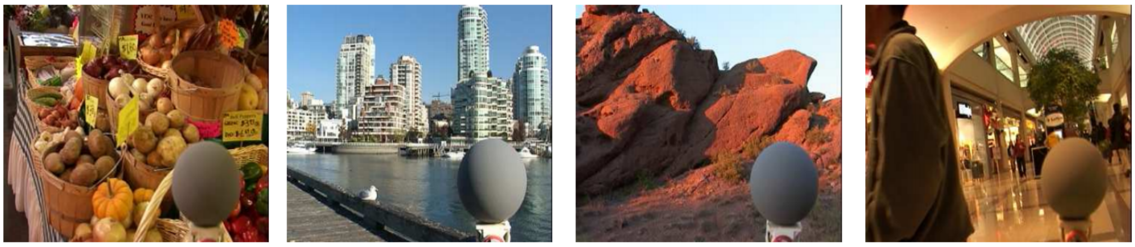


Fig. 2.4 Samples of scenes from the Greyball dataset.

Another dataset using the grey sphere as a colour target is the Calibrated Barcelona dataset by Vazquez-Corral et al. [80]. It has 83 images which were taken with a calibrated Sigma Foveon D10 camera: the colour sensors spectral sensitivities were measured which allows a pixel-wise conversion of the images to the CIE XYZ.

The Cube dataset [78] has 1365 exclusively outdoor images taken with a Canon EOS 550D camera in parts of Croatia, Slovenia and Austria. In the bottom right of every scene a SpyderCube [79] calibration object is placed. Thanks to its different surfaces (black, white and grey) the SpyderCube allows capturing scenes in a wide range of exposures. Samples from the dataset are presented in Figure 2.5. An extension of the dataset named CubePlus of 342 images was presented by the authors in 2019.

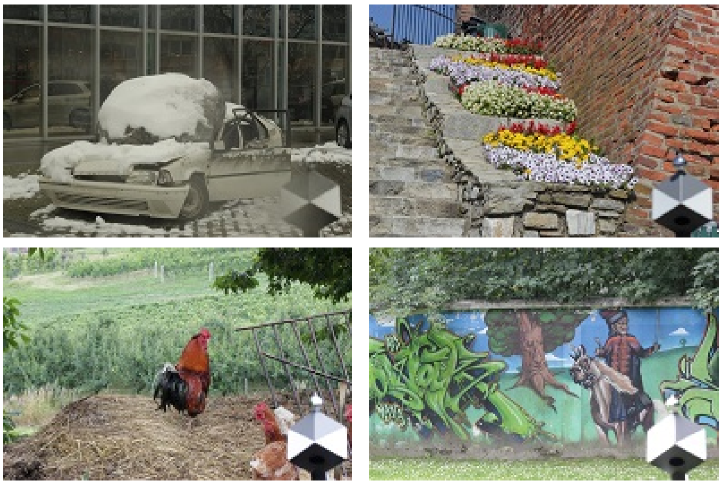


Fig. 2.5 Samples of scenes from the Cube dataset.

Another recent benchmark dataset is the INTEL-TAU [74] which is an extension of the INTEL-TUT dataset [73]. It has 7022 RGB Raw images and is the largest high-resolution dataset available for illuminant estimation.

There exist other types of datasets like hyperspectral datasets. One example is the Barnard's dataset [1] composed of 743 scenes captured under 11 distinct illuminants, where every scene is provided with the spectral data measured with a spectrophotometer. The dataset also includes the camera sensor functions, 4 sets of illuminant spectra, and a set of 1995 surface reflectance spectra compiled from several sources. Another hyperspectral dataset is the ICVL dataset [81] composed of 100 images of 1392×1300 in spatial resolution and 519 spectral bands (400–1000nm at roughly 1.25nm increments).

2.4.2 Error Metrics for Evaluating Algorithms

Most illuminant estimation algorithms return the RGB of the estimated scene illuminant. The ground-truth RGB is extracted from the image, from a reference target (e.g. grey patch). Then, the error is a function of the estimated and ground-truth RGBs. Summary statistics of the errors over the dataset of images (mean, median, quantiles, etc) are given to conclude on the performance of the algorithm.

The choice of the metric to calculate this error is important as different metrics may result in different errors then in different rankings of the methods. Surveys about these metrics are presented in [82][83].

So how do we compare two vectors? First it is accepted that we cannot recover the absolute intensity of the light so we wish to compare the vectors in an intensity independent way. There are two common approaches. In the first each RGB is converted to a chromaticity $r = R/(R+G+B)$ and $g = G/(R+G+B)$ and $b = B/(R+G+B)$ and the error is the Euclidean distance (Equation 2.12) between the estimated and ground-truth chromaticities. More-commonly research use the angular error (Equation 2.10), first introduced [84].

$$err_{recovery} = \cos^{-1}\left(\frac{\underline{E}_{gt} \cdot \underline{E}_{est}}{||\underline{E}_{gt}||\,||\underline{E}_{est}||}\right) \tag{2.10}$$

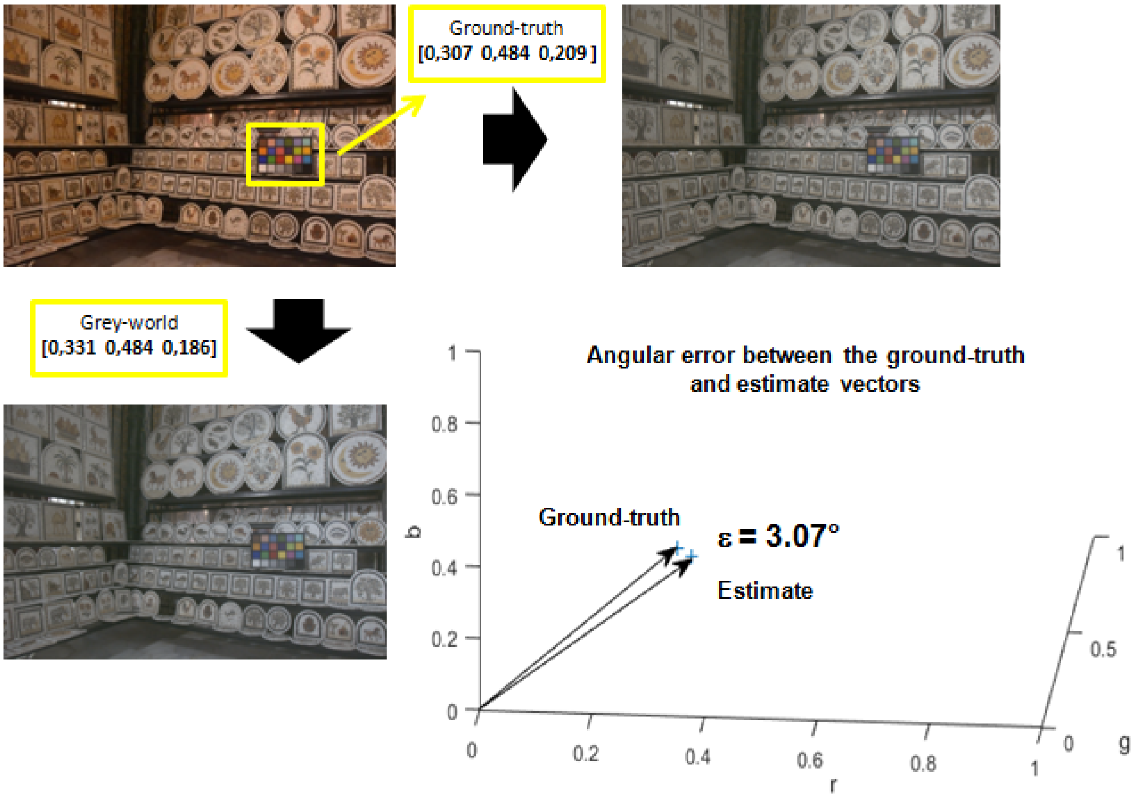


Fig. 2.6 Original image (upper left) and white-balanced image with ground-truth measured from the ColorChecker (right) and with an estimate from the Grey-world method (bottom left). The angle between the 2 illuminant colour vectors is plotted.

where $\underline{E_{gt}}$ is the ground-truth RGB 3-vector, $\underline{E_{est}}$ is the estimated RGB 3-vector (output of the algorithm), $'.'$ is the vector dot-product and $'\|\ \|'$ is the Euclidean norm, also called L_2 norm.

The angular error is defined as the angle between the ground-truth and the estimated vectors (see Figure 2.6). Finlayson et al. reviewed the angular metric in [10] which they refer to as the recovery error and introduced a new angular error, the reproduction error. It measures the angle between the RGB image of an achromatic surface under a white light and the RGB image of an achromatic surface when the actual and estimated illuminants are ‘divided out’, as expressed in the following Equation 2.11.

$$err_{reproduction} = \cos^{-1}\left(\frac{\left(\underline{E}_{gt}/\underline{E}_{est}\right)\cdot\underline{U}}{\left\|\underline{E}_{gt}/\underline{E}_{est}\right\|\sqrt{3}}\right) \tag{2.11}$$

where ’/’ is the element-wise division and $\underline{U}$ is the 3-unit vector and $\|\underline{U}\| = \sqrt{3}$. It was shown in [10] that the reproduction angular error is more stable than its counterpart, the original angular metric. In fact, with the new metric, the angular error for the same scene and algorithm stays constant when the illuminant varies. This property was tested on various benchmark datasets [10], and this is the reason why the reproduction angular error is a more accurate metric for the evaluation of illuminant estimation.

The Euclidean distance gives the L_2 norm between the two illuminant vectors: the estimate and the ground-truth in the *rgb* chromaticity space. It is expressed as follows:

$$d_{euc}(\underline{E}_{gt},\underline{E}_{est}) = \sqrt{(r_{gt}-r_{est})^2+(g_{gt}-g_{est})^2+(b_{gt}-b_{est})^2} \tag{2.12}$$

where $\underline{E_{gt}}$ is the ground-truth colour vector and $\underline{E_{est}}$ is the estimate.

The Euclidean distance can be generalised to the Minkowski distance which is a family of distances. The Minkowski distance is equal to the Euclidean distance for $p = 2$. It is written as follows:

$$d_p(\underline{E}_{gt}, \underline{E}_{est}) = ((r_{gt} - r_{est})^p + (g_{gt} - g_{est})^p + (b_{gt} - b_{est})^p)^{\frac{1}{p}} \qquad with \quad p \geq 1 \tag{2.13}$$

Other metrics are referred to as perceptual distances [82]. These error measures are calculated in colour spaces considered as perceptually uniform like CIELAB and CIELUV [85] or using more intuitive coordinates : chroma C and hue h.

The Euclidean distance can be calculated in the CIELAB colour space. It is considered as more uniform. The sensitivity function of the human visual system is nonuniform, for this reason the weighted (Perceptual) Euclidean distance PED is a good alternative metric (Equation 2.14).

$$PED(\underline{E}_{gt}, \underline{E}_{est}) = \sqrt{w_R(r_{gt} - r_{est})^2 + w_G(g_{gt} - g_{est})^2 + w_B(b_{gt} - b_{est})^2} \tag{2.14}$$

where r, g and b are the chromaticities values of the 3-colour vectors $\underline{E}_{gt}$ and $\underline{E}_{est}$ and the weights represent spectral sensitivity measures with $w_R + w_G + w_B = 1$.

Other metrics are the colour-difference in the CIE76 space (see Equation 2.15) and the CIE2000 space ΔE_{00} [86]. We use them for the evaluation of colour stabilisation methods in Chapter 6.

$$\Delta E^*_{ab} = \sqrt{(L^*_{gt} - L^*_{est})^2 + (a^*_{gt} - a^*_{est})^2 + (b^*_{gt} - b^*_{est})^2} \tag{2.15}$$

where L^*, a^* and b^* are the components of $\underline{E}_{gt}$ and $\underline{E}_{est}$ colour vectors in the CIE LAB colour space, and represent respectively the lightness, the green-red component and the blue-yellow component.

CIE ΔE_{00} metric is based on CIELAB and was developed following the procedures agreed by CIE TC1-47 [87]. It includes not only lightness, chroma, and hue weighting functions, but also an interactive term between chroma and hue differences for improving the performance for blue colours and a scaling factor for the CIELAB a^* scale for improving the performance for gray colours [88].

2.5 Colour Stabilisation

Colour stabilisation is about colour consistency between frames of the same scene and aims at having the same objects appearing with the same colour when the shooting conditions change: the camera parameters change or when multiple cameras are used. In fact, often in videos we notice a ‘flickering’ [89] as we move the camera or as objects in the scene move. This is because all the algorithms in the camera’s processing pipeline are based on the content of the scene. And, if the scene content changes then the processing applied to the captured image also changes and the video flickers. Colour stabilisation is particularly important in TV broadcasting and for amateur videos processing.

The colour stabilisation problem is posed as follows: given a source image and a target image (also referred to as a reference image in [90]), where some regions or objects appear in both images, how can we process the source image to match the target image and looks as if it was captured under the same conditions?

2.5.1 SIFT-based Colour stabilisation

In colour stabilisation, from frame to frame of the same scene, there might be a lot of content that is the same but also significant scene content will have changed. So, the first step of colour stabilisation is to identify points and/or regions that match. This is not an easy problem. In the first colour stabilisation algorithm, we find corresponding points using SIFT.

The SIFT method was introduced initially for object recognition by Lowe [91]. SIFT stands for Scale-Invariant Feature Transform. It says that the method transforms image data into scale-invariant coordinates relative to local features. The SIFT features provide a robust matching and are invariant to image scaling, rotation, change in the illumination and to 3D camera viewpoint. They are localised in both the spatial and frequency domains making them more robust to noise.

The number of features depends on the scene content and on the tuning parameters. For many computer vision applications the higher the number of features the better.

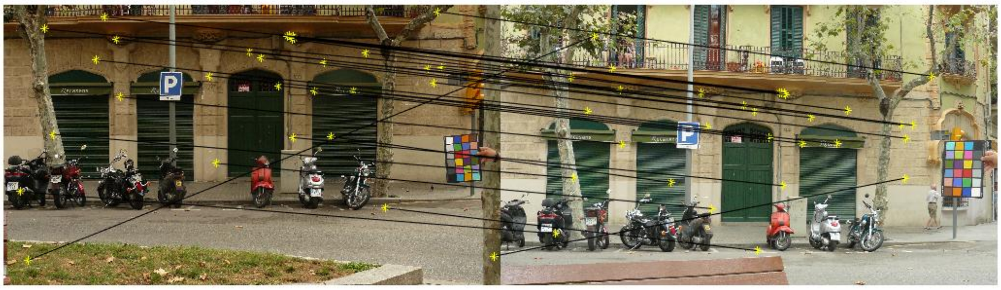


Fig. 2.7 SIFT correspondence points between two images using the VLFeat algorithm from [3][4]. Only 30 matches are plotted.

Vazquez-Corral and Bertalmío [90] use the SIFT implementation from [3] for colour stabilisation. The algorithm of [3] works as follows: it first finds points of interest in the images $\mathbf{I}_{src}$ and $\mathbf{I}_{tar}$ and builds feature descriptors to characterise these points. Now these two point sets must be matched to each other. For every single feature descriptor from one image the algorithm gives the closest descriptor from the other image. It outputs the positions of the matching descriptors respectively in the 2 images such as the points of interest match as follows :

$$\mathbf{F}_{src} \approx \mathbf{F}_{tar} \tag{2.16}$$

where $\mathbf{F}_{src} = \mathbf{I}_{src}(P_{src})$ and $\mathbf{F}_{tar} = \mathbf{I}_{tar}(P_{tar})$ are the corresponding points in the images source and target (P_{src} and P_{tar} are index sets with the positions of these points in every image).

For a pair of images, a source and a target, SIFT finds the sets of correspondence points as shown in Figure 2.7.

The algorithm gives 2723 matches for this pair of images (from about 15000 features descriptors found in every image). Given the correspondence points the SIFT-based colour stabilisation aims at finding the mapping transform, denoted $\mathbf{T}_{LST}$ by minimizing the colour error with a least-squares regression [90] as follows:

$$\mathbf{T}_{LST} = \mathbf{F}_{src}{}^{+}\mathbf{F}_{tar} \tag{2.17}$$

where $^{+}$ denotes the Moore-Penrose inverse.

The mapping transform is then used to process the source image to match the target. Details of how the SIFT algorithm works is outside the scope of this thesis, but details can be found in [91][3][4].

2.5.2 RANSAC+SIFT-based Colour Stabilisation

When we run SIFT on the images $\mathbf{I}_{scr}$ and $\mathbf{I}_{tar}$, we have the set of points of interest $\mathbf{F}_{src_o}$ and $\mathbf{F}_{tar_o}$ for the images. Further, assuming that the content changes between 2 image frames of the same scene we expect similar content to be spatially proximate between the 2 images. Thus, the matching problem can be described as finding the ‘nearest neighbour’ to a point in $\mathbf{I}_{scr}$ (in I_{tar}) (where ‘nearest’ is a small neighbour of image locations). Every point in $\mathbf{F}_{src_o}$ has one candidate match in $\mathbf{F}_{tar_o}$. In the work of [90] it is assumed that correctly matching pairs should be linearly related. To find this linear transform we randomly choose

3 corresponding points and calculate the matching transform (we have 3 pairs of matched RGBs –the SIFT points– so they must be exactly linearly related by a 3×3 matrix). We then see how well this matrix predicts the rest of the data. We repeat this process until we find the 3×3 matrix that works best overall. What we have just described is an application of the RANdom SAmple Consensus (RANSAC) algorithm.

RANSAC [92] is a well-known and widely used technique in computer vision in particular for image registration [93] and object matching and recognition [94]. It is also used to solve for colour-related problems like for colour correction [95] i.e the mapping to XYZ colour space and illuminant estimation [96].

Vazquez-Corral and Bertalmío [90] proposed using RANSAC to solve for colour stabilisation given a source image and a target image. The RANSAC refines the SIFT matches. In fact SIFT can generate incorrect matches as shown in Figure 2.7 (e.g. the algorithm wrongly matches the grass in the left image with the tree leaves in the right image) which results in a poor colour stabilisation. RANSAC refines the matches by removing the correspondence pixels which have large colour errors. An example of the refined correspondence pixels between two images from [90] is given in Figure 2.8. RANSAC decreases the number of matches from 2723 to 1743 for this example. Clearly the RANSAC matches are more correct than the SIFT matches in 2.7.

In RANSAC-based colour stabilisation [90], the correspondence points are used to calculate a mapping transform by least-squares regression as follows:

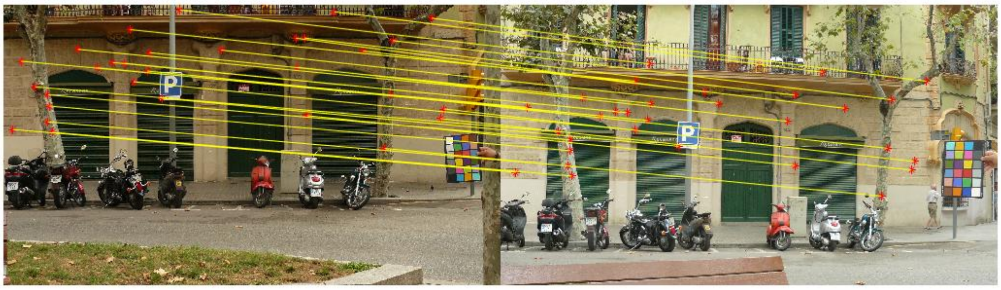


Fig. 2.8 RANSAC correspondence points between two images. Only 30 matches are plotted.

$$\mathbf{T}_{LST} = \mathbf{F}_{src}^{'}{}^{+}\mathbf{F}_{tar}^{'} \tag{2.18}$$

where $\mathbf{F}^{'}$ is the refined new set of points with RANSAC.

The mapping transform is then used to process the source image to match the target. Details of how RANSAC points are defined and found is outside the scope of this thesis, but details can be found in [92].

Chapter 3

A Curious Problem with Using the ColorChecker Dataset for Illuminant Estimation

In the previous chapter, we presented some of the commonly used methods and better performing algorithms of illuminant estimation for colour constancy. We talked about the importance of evaluating these algorithms in order to be able to compare them to prove that a newly introduced method advances the state of the art. The evaluation is made using benchmark datasets of images. In other terms, to compare algorithms A and B, we choose one set of images and we perform the methods A and B on these images to estimate the illuminants colours, then we compare these estimates to the pre-determined ground-truth

–or set of correct answers of the illuminants colours – by calculating various error measures to make conclusions on the performance of A and B and to be able to compare them. Consistently, the images used for the evaluation and comparison should be the same, as well as the ground-truth.

In this chapter, we present the curious problem with the ColorChecker dataset, perhaps the most widely used publically available dataset used in illuminant estimation. More specifically, we found that this dataset not only has at least three sets of different ground-truths that have been used for performance evaluation all together but also that these ground-truths have calculation errors. This curious problem led to misleading results of the evaluation and rankings of illuminant estimation algorithms. Moreover, when re-evaluating algorithms with the three ground-truths separately we discovered that the rankings are totally different when using one or the other ground-truth and can even be in a reverse order. This is a quite worrying fact because depending on which ground-truth is used, authors will claim that their particular algorithm is better than another. Some of the work reported in this chapter also appears in [5][6][7].

3.1 Dataset Description

The ColorChecker dataset was introduced by Gehler et al. in 2008 [32]. It has 568 RGB images of indoor and outdoor scenes collected by the first author Peter Gehler when he was doing an internship at the Vision Group of Microsoft Research offices in Cambridge, in and around the research offices. The images include photographies of people, landscapes and other tourist-type scenes in various light conditions. Two DSLR (Digital Single Lens Reflex) cameras were used, a Canon 1Ds (86 images) and a Canon 5D (482 images). All cameras settings were set to auto-mode and the images were saved in a Canon Raw format (of size 4082×2718 pixels for the Canon 1Ds and 4386×2920 pixels for the Canon 5D images), plus the camera pipeline JPEG format.

The Raw format allows saving the image measured by the CMOS sensor without any extra processing. Information about automatic white balancing (AWB), automatic exposure (AE) and other camera parameters of the image capture are stored as metadata of the Raw file.

Compared to the other available benchmark datasets, the ColorChecker dataset has an average number of high quality images as well as a medium variety of photographic scenes scenes compared to other datasets which are mostly lab-based or technical scenes [97]. The scenes in the images look more to what camera users would often capture.

Every image has a classic X-Rite Macbeth ColorChecker [72] placed in it. The colour chart is used for measuring the colour of the illuminant in the scene. The ground-truth for every scene is defined as the RGB response from the achromatic (grey) patches of the ColorChecker. The ground-truth of the ColorChecker dataset is a 568×3 vector of white-points, one 3-vector white-point for each image from the set. Researchers in colour constancy have been using a linear version of the dataset processed by Shi and Funt and available for download from the sfu web-site [98] since 2011. These are the images that were used with the algorithms that we present in what follows. Samples from the dataset (outputs of the camera pipeline) are shown in Figure 3.1.

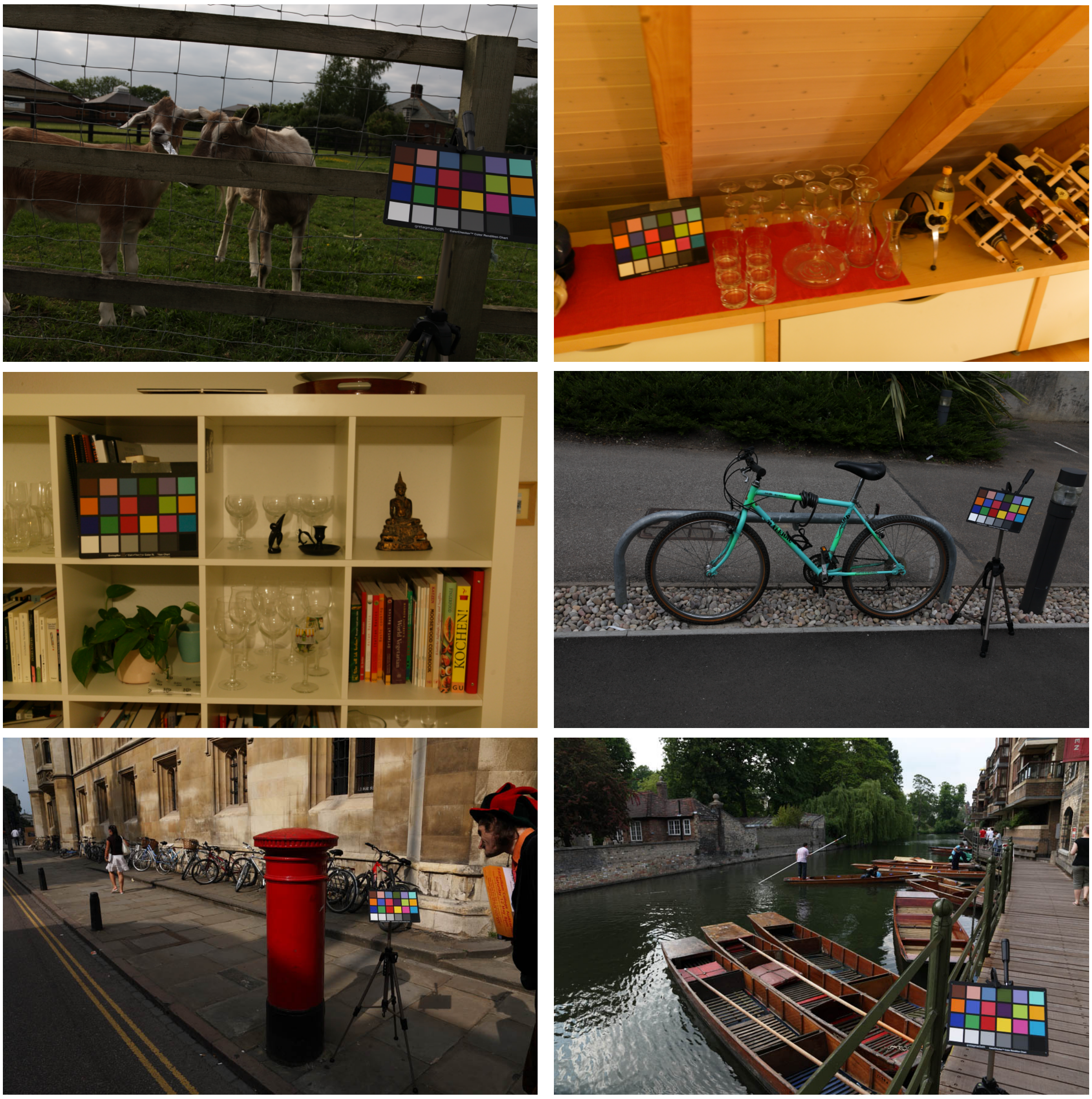


Fig. 3.1 Samples of indoor and outdoor scenes from the ColorChecker dataset; here the images are in a JPEG format. They are the camera pipeline outputs.

The ColorChecker dataset is one of the most referred to sets of images for evaluating the performance of illuminant estimation algorithms. The ground-truth is not only necessary to be compared to the algorithm estimates for the performance evaluation but also when considering learning methods, like deep-learning methods [34][36], the ground-truth is used for training the network and learning the model.

3.2 A Curious Problem with the ColorChecker Dataset

3.2.1 Problem Description

We found that the ColorChecker dataset has more than one set of ground-truth, that every set was calculated differently and more importantly that these sets were used indistinctly to compare algorithms and rank them when evaluated on the ColorChecker dataset. This situation was a nonsense. The first ground-truth set is from the sfu web-site [98]. We label it the SFU ground-truth (referred to as SFU/Gt3 in [7]). The two other sets are from colorconstancy.com [55], a widely used comparison site for illuminant estimation research hosting data and results. We label these two ground-truth sets Gt1 and Gt2.

The three sets are different. Gt1 and Gt2 have similar colour distributions but this small difference has still significant effects on the results that we present in Section 3.3. We plot in Figure 3.2 the rg chromaticities gamut of two of the ground-truth sets SFU and Gt1. It shows that the distributions of SFU and Gt1 are quite separate. The distribution of SFU is also more compact, which suggests that if SFU is the true correct answer for the white-points, then the ColorChecker dataset has a smaller variety of illuminants and therefore illuminant estimation would be more easily performed on this dataset. We explain in the next section where these sets of ground-truth come from and the differences in their calculations.

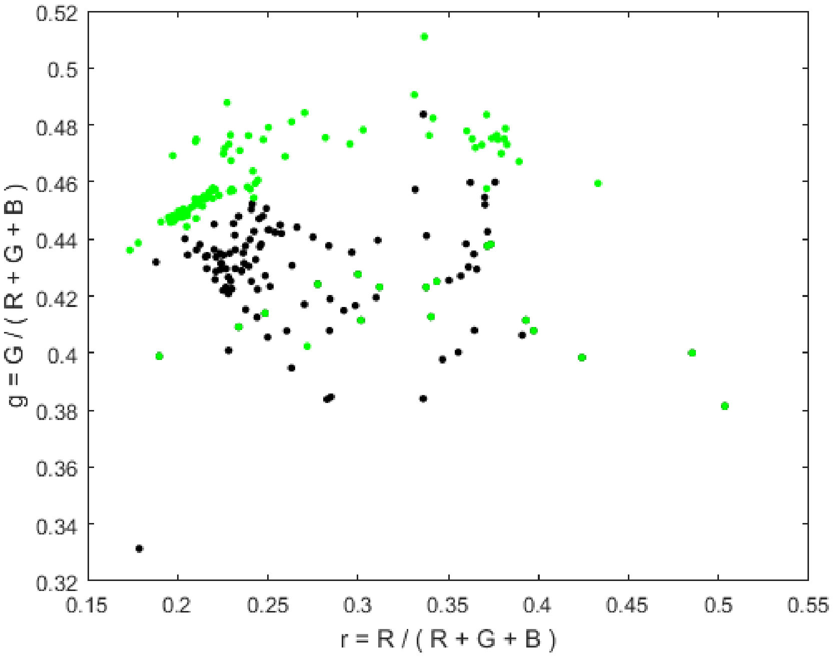


Fig. 3.2 2D chromaticity gamut (r,g) of the ColorChecker dataset ground-truths.(green) SFU ground-truth, (black) Gt1 ground-truth. Gt2 (r,g) distribution is very close to Gt1, though slightly different (so, for clarity, is not plotted)[5].

3.2.2 Ground-truth Sets History and Calculation Methodology

The three existing sets of ground-truth of the ColorChecker dataset, called, SFU, Gt1 and Gt2 were created at different points of time. SFU, from the sfu web-site, was introduced by Shi and Funt. On the same web-page, the calculation methodology is given. It is the calculation methodology that we adopted for the re-calculation of the ground-truth that we present in the next chapter. The illuminant colour corresponds to median RGB response from the brightest achromatic patch (median R, median G and median B).

The Gt1 ground-truth is from the same data (patches RGBs) as SFU. We found the Gt1 ground-truth on colorconstancy.com. Authors used mostly Gt1 for evaluating algorithms while SFU was used more recently with deep-learning methods like in [36]. The first author of [36], Jonathan Barron, declared to us that the SFU ground-truth was used in a personal communication [99].

The methodology described by Shi and Funt stipulates that a 'black level' needs to be subtracted from the Raw data (Raw images and patches RGBs) before any further processing. The 'black level' is the average camera response when there is no light i.e. when the picture is taken with the lens cap on [100]. It was estimated for this dataset by finding the minimum pixel values across the whole dataset and it is equal to 0 for the Canon 1Ds images and 129 for the Canon 5D images [98].

Now we can explain more clearly the difference between SFU and Gt1: Gt1 does not have the 'black level' subtracted (from the patches RGBs data), while SFU, introduced afterwards, helped correcting this error in the calculation even though the problem with Gt1 was not known and not yet identified at that time. However, following the introduction of SFU, both Gt1 and SFU continued to be used for algorithms evaluation and this led to even a bigger problem of mixed rankings of illuminant estimation methods on the ColorChecker dataset.

Thus, the difference between Gt1 and SFU is the 'black level' subtraction: when Gt1 was calculated, the calculation methodology instructions on the sfu web-site indicated the 'black level' was already subtracted from the data. Later around March 2016, the text of the web-page was edited to indicate the reader still needs to subtract the 'black level' offset from the original images. So, in effect, Gt1 is the ground-truth correctly calculated according to the instructions on the sfu website before 2016, and SFU, is the data correctly calculated –according to the new methodology– after 2016.

The ground-truth Gt2 has a different story, more uncoupled from the other two, although also from the same linear images. The Gt2 set is due to Bianco et al. [101] who found errors in SFU : 'Using the one generated by Shi and Funt, we noticed that for some images the Macbeth ColorChecker coordinates (both the bounding box and the corners of each patch) were wrong and thus the illuminant ground-truth was wrong.' Indeed, we also noticed these other errors in the calculation of SFU and consequently of Gt1 (derived from the same data) when comparing the re-calculated ground-truth (see next Chapter 4) to the existing ones. We also found a problem with the calculation itself of SFU (for a small number of images). Specifically, we found that a saturation problem occurred in 3 images where the R, G and B of the light come from two (or three) different achromatic patches, e.g. from the white patch and the lightest grey-patch. To solve this ambiguity, we propose to read the R,G and B values of the illuminant colour from the same brightest non-saturated (no digital count in any channel >3300) grey patch.

However, Gt2 did not perfectly solve the problem of the ground-truth. In fact, the 'black level' offset was not subtracted from the original Raw images when Gt2 was calculated. For this reason and except for some images, Gt2 is similar to Gt1. Bianco et al. added to their comment: 'To have an idea of the differences, the maximum angular difference between Gt1 and Gt2 is around 20°, and the median one is around 0.03°'. The angular error (the angle between the ground-truth and estimated light vectors, see later in this chapter for a discussion) here expresses the same as chromaticity differences between a pair of white-points (see Section 2.4.2 for more details about this metric). Significantly, that the ground-truth Gt2 have been also used more recently in learning methods (learning and evaluation test on the ColorChecker dataset), for instance by Bianco et al. [34].

3.3 Evaluating Illuminant Estimation Algorithms

The existence of more than one set of ground-truth of the ColorChecker dataset with wrong calculation methodologies resulted in misleading results in the field of colour constancy. These results– performance evaluation and rankings– are reported in the comprehensive illuminant estimation survey by Gijsenij et al. [97] and on colorconstancy.com [55]. We updated the web-site with the corrected data in March 2018, under section Datasets\ColorChecker RECommended.

3.3.1 Algorithms Evaluation

In this section we present results of performance evaluation and rankings with the three sets of ground-truths of the ColorChecker dataset that we found and called, SFU, Gt1 and Gt2. We chose 23 illuminant estimation algorithms and re-evaluated them using the ColorChecker output results (estimates) available to us from colorconstancy.com [55]. We calculated the angular errors (see Section 2.4.2) using the estimate and the three ground-truths for every algorithm and we found that the results of the evaluation are very different when using one or the other ground-truth. This is shown for six algorithms in Figure 3.3 in terms of mean recovery error (Equation 2.10) and in Figure 3.4 in terms of mean reproduction error (Equation 2.11) over the dataset when comparing the results for the three ground-truths.

The problem is more serious when there is large change in order or when the rankings of some algorithms are in reverse order depending on the ground-truth we use. This is the case for the algorithms 1st and 2nd orders Grey-Edge [54], Bayesian [32][52] and CART-based Combination [56], which are in reverse order for SFU compared to Gt1 or Gt2. There is no change in ranks when using Gt1 or Gt2, as explained in the previous section the two sets are very similar except for a few outliers, the coordinates of the colour chart in some of the images were corrected in the calculation of Gt2.

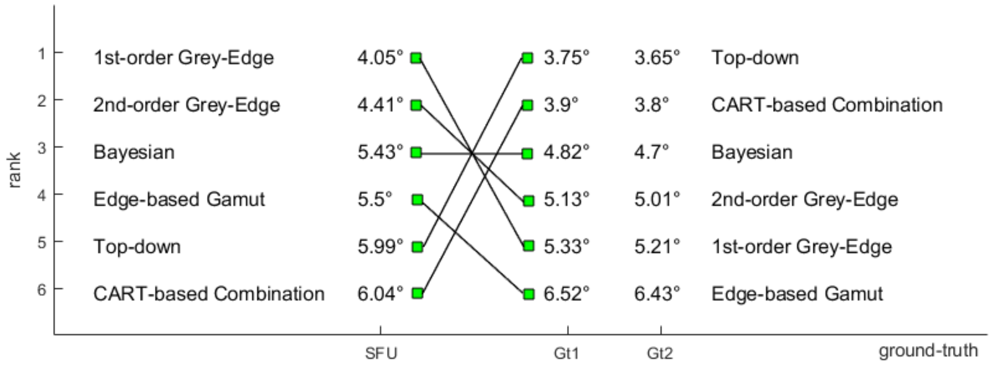


Fig. 3.3 Ranks of six algorithms in terms of mean recovery error for the 3 ground-truths, SFU, Gt1 and Gt2 and values of the errors [5].

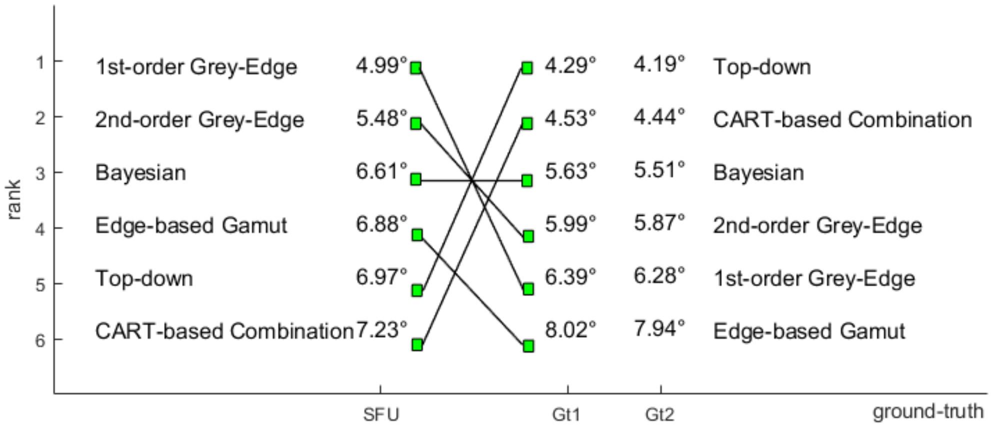


Fig. 3.4 Ranks of six algorithms in terms of mean reproduction error for the 3 ground-truths, SFU, Gt1 and Gt2 and values of the errors [5].

Tables 3.1 to 3.4 show the six best ranked algorithms (from the 23 chosen ones) in terms of median, trimean ($Trimean = \frac{25\%quantile+2\times median+75\%quantile}{4}$) and 95% quantile reproduction error, for separately the three ground-truths SFU, Gt1 and Gt2.

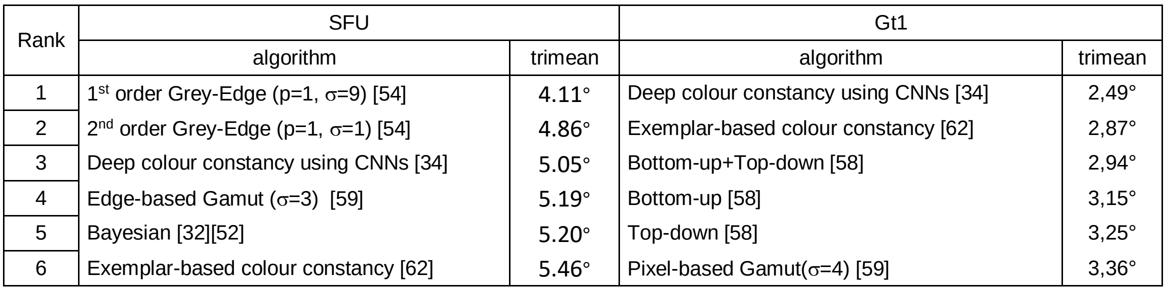
Often the same algorithms do not appear on the best six for SFU versus Gt1. In terms of errors as well, SFU does not give the same range of errors (median or trimean or 95% quantile) as Gt1. Errors when using SFU seems larger for the same algorithm. This is also the case for learning-based algorithms. In fact, optimisation methods like [34] used for the training stage the Gt1 ground-truth, so we can expect the errors to be higher when tested using the different ground-truth set SFU and this is actually the case. We did not re-train these algorithms models on the different ground-truths –this is because the codes were not available to us– but we used the available models outputs from colorconstancy.com. Another example is the Fast Fourier Colour Constancy algorithm of Barron and Tsai [36]. The model was trained using SFU and the test error with SFU is smaller than the error using the Gt1 ground-truth.

The ranks of algorithms using Gt1 and Gt2 are the same (for the mean and median errors) as well as the errors which are very close –for this reason we do not show them– but the rank differs slightly for the 95% quantile error (see Table 3.4) for Gt1 versus Gt2.

Table 3.1 The six best algorithms in terms of median reproduction error for SFU vs Gt1 (best algorithms for Gt2 are the same), the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [5].

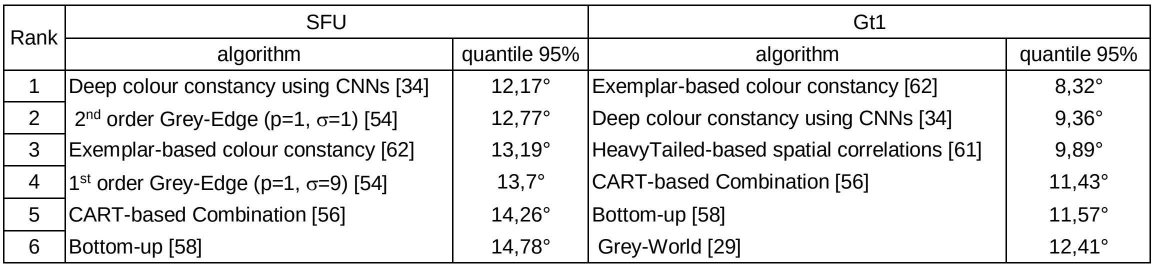
Rank	SFU		Gt1	
	algorithm	median	algorithm	median
1	1st order Grey-Edge (p=1, σ=9) [54]	3,77°	Deep colour constancy using CNNs [34]	2,24°
2	Edge-based Gamut (σ=3) [59]	4,54°	Exemplar-based colour constancy [62]	2,64°
3	2nd order Grey-Edge (p=1, σ=1) [54]	4,59°	Pixel-based Gamut (σ=4) [59]	2,73°
4	Bayesian [32][52]	4,62°	Intersection-based Gamut (σ=4) [59]	2,74°
5	Deep colour constancy using CNNs [34]	4,75°	Bottom-up+Top-down [58]	2,75°
6	Pixel-based Gamut (σ=4) [59]	5,21°	Bottom-up [58]	2,98°

Table 3.2 The six best algorithms in terms of trimean reproduction error SFU vs Gt1 (best algorithms for Gt2 are the same), the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [5].



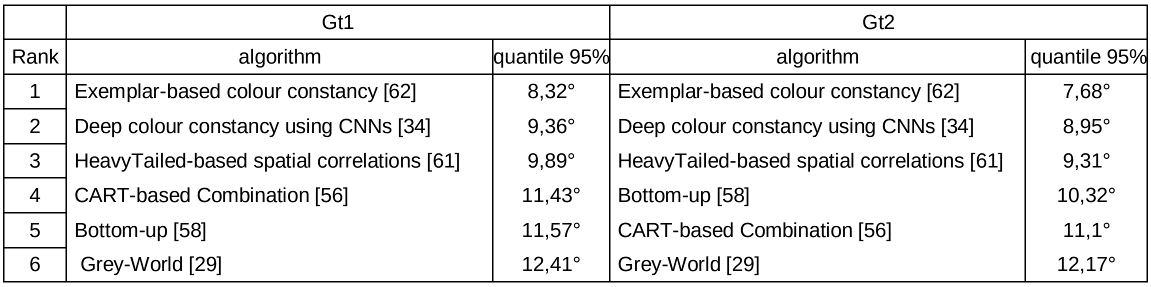
Rank	SFU		Gt1	
	algorithm	trimean	algorithm	trimean
1	1st order Grey-Edge (p=1, σ=9) [54]	4.11°	Deep colour constancy using CNNs [34]	2,49°
2	2nd order Grey-Edge (p=1, σ=1) [54]	4.86°	Exemplar-based colour constancy [62]	2,87°
3	Deep colour constancy using CNNs [34]	5.05°	Bottom-up+Top-down [58]	2,94°
4	Edge-based Gamut (σ=3) [59]	5.19°	Bottom-up [58]	3,15°
5	Bayesian [32][52]	5.20°	Top-down [58]	3,25°
6	Exemplar-based colour constancy [62]	5.46°	Pixel-based Gamut(σ=4) [59]	3,36°

Table 3.3 The six best algorithms in terms of quantile 95% reproduction error SFU vs Gt1, the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [5].



Rank	SFU		Gt1	
	algorithm	quantile 95%	algorithm	quantile 95%
1	Deep colour constancy using CNNs [34]	12,17°	Exemplar-based colour constancy [62]	8,32°
2	2nd order Grey-Edge (p=1, σ=1) [54]	12,77°	Deep colour constancy using CNNs [34]	9,36°
3	Exemplar-based colour constancy [62]	13,19°	HeavyTailed-based spatial correlations [61]	9,89°
4	1st order Grey-Edge (p=1, σ=9) [54]	13,7°	CART-based Combination [56]	11,43°
5	CART-based Combination [56]	14,26°	Bottom-up [58]	11,57°
6	Bottom-up [58]	14,78°	Grey-World [29]	12,41°

Table 3.4 The six best algorithms in terms of quantile 95% reproduction error for Gt1 vs Gt2, the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [5].



	Gt1		Gt2	
Rank	algorithm	quantile 95%	algorithm	quantile 95%
1	Exemplar-based colour constancy [62]	8,32°	Exemplar-based colour constancy [62]	7,68°
2	Deep colour constancy using CNNs [34]	9,36°	Deep colour constancy using CNNs [34]	8,95°
3	HeavyTailed-based spatial correlations [61]	9,89°	HeavyTailed-based spatial correlations [61]	9,31°
4	CART-based Combination [56]	11,43°	Bottom-up [58]	10,32°
5	Bottom-up [58]	11,57°	CART-based Combination [56]	11,1°
6	Grey-World [29]	12,41°	Grey-World [29]	12,17°

3.4 Conclusion

In this chapter of the thesis, we presented the problem with the ground-truth of the ColorChecker dataset, an extensively used dataset for the performance evaluation of illuminant estimation algorithms. The problem is due to the existence of three different sets of ground-truths that have been used all together in the evaluation. This curious situation is a nonsense. In addition to this problem these ground-truths are not completely accurate and we explain this assertion in details in this chapter.

Chapter 4

Rehabilitating the ColorChecker Dataset for Illuminant Estimation

We presented in the previous chapter the rather worrying discovery about the ground-truth –the correct answers of the illuminants colours– of the ColorChecker dataset for illuminant estimation. We showed that there exist three sets of ground-truth not only different but also with incorrect calculation methodologies. All of these three sets are used in algorithms evaluation. But, they are used separately. Authors conclude that their algorithm outperforms another using one set of ground-truths but the comparison algorithm was optimised for a completely different set of ground-truths.

The evaluation results of the state-of-the-art methods in colour constancy on benchmark datasets are provided on colorconstancy.com and reported in [97]. Worryingly, the fact that there are three ground-truth sets used is not highlighted. Rather, all the algorithms simply stipulate they are using the ColorChecker dataset without mentioning which ground-truth was used (because prior to our work the issue of multiple ground-truths was not understood). So right now the field is comparing apples with oranges. The question we asked is : how do we rehabilitate the ColorChecker dataset for illuminant estimation following this discovery?

We worked with Peter Gehler, author of the ColorChecker dataset, Arjan Gijsenij, author of colorconstancy.com, Simone Bianco, author of the Gt2 ground-truth set [101] and other renowned researchers in the field of colour constancy and we made the decision to re-calculate the ground-truth, that we called REC for recommended ground-truth. We then re-evaluated state-of-the-art and most recent algorithms on the REC ground-truth.

In this chapter, we first explain the calculation methodology of the new re-calculated ground-truth REC of the ColorChecker dataset. Then we compare this new set with the existing ones in terms of similarities and differences. Finally in the last section, we present results on the evaluation and rankings of illuminant estimation algorithms and compare them with the available results in the field, hoping that with this work we are bringing a final answer to the ColorChecker dataset problem. Some of the work reported in this chapter also appears in [6][7].

4.1 Providing a Single Ground-truth for the ColorChecker Dataset

The ColorChecker dataset for the performance evaluation of illuminant estimation methods has three sets of ground-truth that we labelled respectively SFU, Gt1 and Gt2. None of the three sets is correctly calculated. We made the decision to re-calculate the ground-truth and we called the new set REC, for recommended.

We re-processed the dataset Raw images and re-calculated the ground-truth using the processing and calculation methodology described by Shi and Funt on the sfu web-site [98].

4.1.1 Dataset Images Processing

We used a linear version of the images that we generated ourselves from the Raw set of images collected by Gehler et al.[32]. The Raw images are 4082×2718 pixels (Canon 1Ds) and 4386×2920 pixels (Canon 5D) 12-bit floating-point format with an RGGB Bayer pattern.

We performed a linear processing with DcRaw (a programme for decoding Raw image format files)[102]. Consistently for testing colour constancy methods, we do not apply at this stage the auto-saved white balance.

We do not use the DcRaw demosaicing option. Rather, we simply downsample the R, G and B images. The R and B images are downsampled directly (there is only one non-zero value in each 2×2 pixel region and we choose this in the half resolution image). For the green channel there are two non-zero values and we average these. The manual demosaicing takes into account that the two cameras, the Canon 1Ds and the Canon 5D have the same pixels pattern but with different offsets.

In Figure 4.1 we show an example of the processed images. We have applied a 2.2 gamma to make the images bright enough to see (but this gamma is not applied when we process the images, it is for illustration only). The images look flat because there is no 'tone mapping' applied to the images [103]. That the images have a mild greenish cast is, presumably, due to the green sensor in the camera being relatively more sensitive to light than the blue and red channels. As prescribed in the Shi-Funt methodology [98], we also subtracted the black-level (see last chapter for details). An important detail is the images are 12-bit (in the range [0,4095]).

The images that we create are available from on colorconstancy.com, (datasets web-page). We also provide two sets of mask images, the first for masking saturated (one or more colour channels with digital count$>$3300, knowing that the images are 2^{12}-bit floating-point) and clipped pixels (R or G or B channel is equal to 0) from every image to use the dataset for illuminant estimation; and the second for masking the colour chart area for every image of the dataset. Besides we provide the colour charts coordinates and a copy of the REC ground-truth, on the same web-page. We also made the code available in the 'Color Chart from Images' repository on github.com.

4.1.2 REC Ground-truth Calculation Methodology

Every image from the dataset has an eponymous Macbeth ColorChecker chart [72] placed in the scene. The ColorChecker has 24 patches including six achromatic (grey) patches. The ground-truth RGB light colour for every scene is defined as the median RGB value from the brightest achromatic patch (ranked by average of the selected squares with no digital count>3300 [each image is in 12 bits]). The median RGB is simply the medians calculated in the red, green and blue channels. All three medians are calculated from the same achromatic patch. In fact, we set up this condition in response to the SFU ground-truth which was occasionally drawn from more than one grey patch for a few images (for R versus G versus B channels), eg. red from the white patch and green and blue from the second grey patch. The 3300 threshold eliminates any maximum clipping and the effects of any possible non-linearity in sensor response that might occur as intensities approaching the maximum of 4095 [98]. Clipping at the sensor level occurs when intensity of the pixel falls outside of the minimum or the maximum which can be represented due to an exposure problem, and results in a pixel channel value cut-off and auto-set respectively to 0 or maximum.

The median is used instead of the mean to exclude any pixels in the square selection (see Figure 4.1) which are not part of the patch but are from the bounding box (this can occur considering the manual selections on the charts) and elements of the scene like vegetation which hides tiny parts of the patches in a few images.

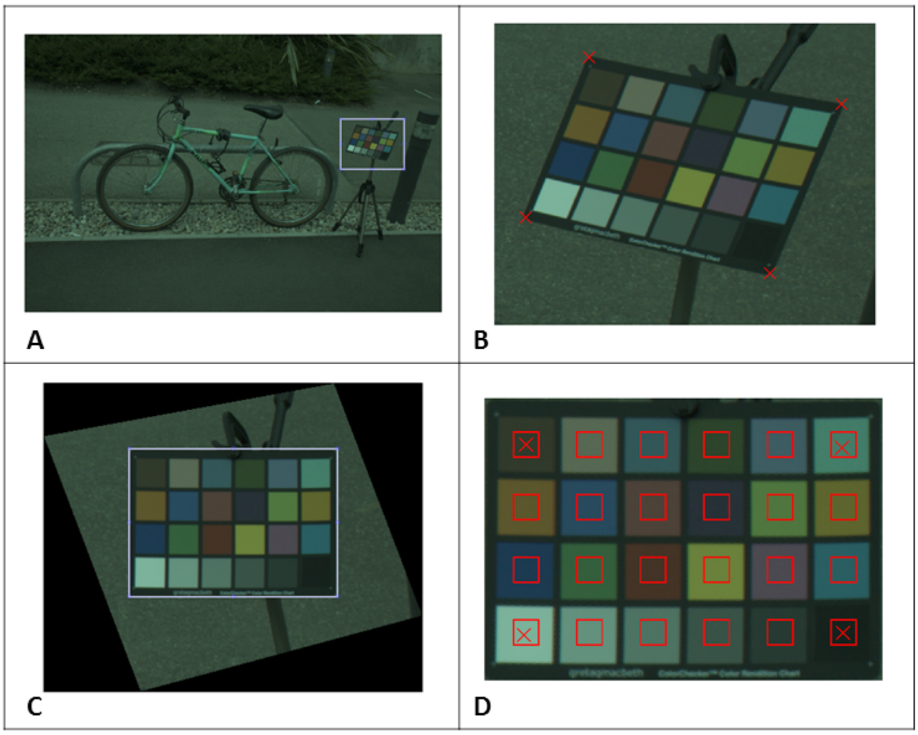


Fig. 4.1 The 4 main steps of the colour chart processing. A) we select roughly the chart area in the image. B) we select the 4 corners of the chart. C) the chart image is geometrically transformed to be in front of the camera, then we select more precisely its contour. D) we select the centres of the 4 corners patches as well as one square of interest in one patch, this square selection is then automatically replicated over all patches. The medians per channel of the achromatic red-square regions are calculated [6].

We used our own code and the toolbox in [104] to calculate the ground-truth. Figure 4.1 shows the main four steps of the processing of the colour chart images for the calculation of the ground-truth with our code. Our method allows processing all the 568 images of the dataset one by one semi-automatically.

4.1.3 Comparison of All Ground-truth Sets

In this section we compare the RECommended ground-truth with the other available sets that we labelled: Gt1, Gt2 and SFU. Figure 4.2 shows, for a subset of the 568 images, a comparison of the chromaticity gamuts of the three sets, Gt2 is not plotted as it is almost the same as Gt1 (see previous chapter). The convex hulls [105] in the same Figure show that the 3 ground-truths distributions have both overlapping and disjoint areas. The SFU ground-truth set is very close to our new ground-truth except for a few outliers: the green squares mainly to the left of the plot which do not overlap with the black crosses. For the specific corresponding images, these points in SFU are different from the ones in REC.

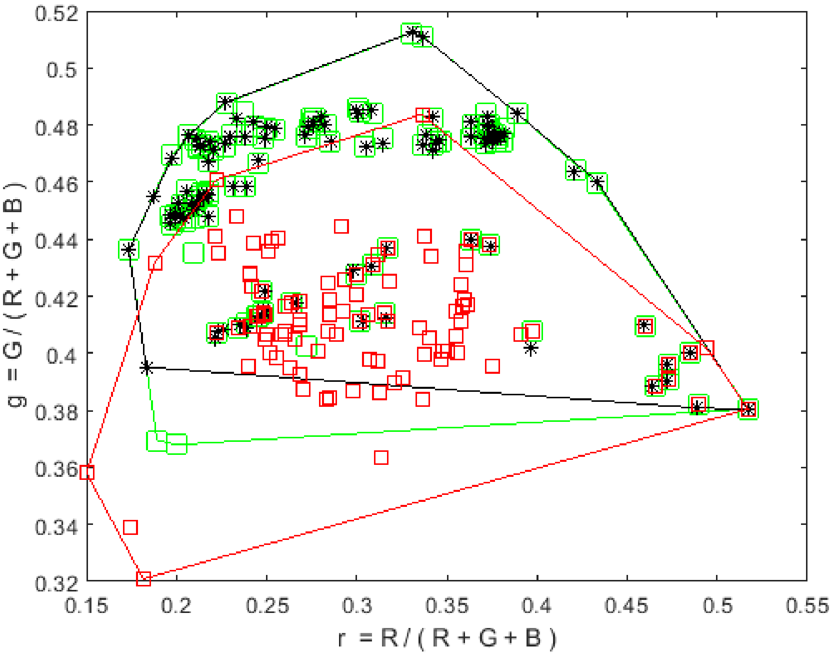


Fig. 4.2 New RECommended ground-truth chromaticities are plotted as black asterisks. The SFU ground-truth is shown as green squares and Gt1 as red squares. The distributions convex hulls are plotted. 100 of the 568 chromaticities are shown. [7]

We can easily relate this to the description of the ground-truths problems presented in the previous chapter. These outliers are due to the problems in calculating correctly the colour charts coordinates (Bianco's observation) [101] and to our own discovery that the white-point was on occasion drawn from different grey patches (for R versus G versus B). As an example, it could be that the R and G values are calculated from the white patch in the colour chart because all the values in these patches are below the 3300 clip limit. In contrast, the green channel might clip for this patch and this means –by the SFU methodology– that the white-point in the green channel was calculated from the light grey-patch on the colour chart. In our methodology, we calculate the values for all three channels from the same light-grey patch. Despite this, the preponderance of the two sets of data are in, more or less, precise alignment. In contrast, the points in Gt1 are far from REC.

4.2 Re-evaluation of Illuminant Estimation Algorithms

In this section, we re-evaluate on the ColorChecker dataset the 23 chosen illuminant estimation algorithms from colorconstancy.com and Barron's learning Fast Fourier algorithm [36] using the REC ground-truth. As for the other three ground-truths Gt1, Gt2 and SFU, we use for this evaluation the illuminants colours estimates of the algorithms available to us from colorconstancy.com and provided to us by Jonathan Barron [99] for the FFCC algorithm. We did not run the different methods ourselves on the ColorChecker dataset because the codes were not always available to us.

We compare the performance evaluation angular errors of the illuminant estimation methods for REC versus SFU versus Gt1. We do not include in the following Tables the results with Gt2, because they are very comparable to Gt1.

Table 4.1 shows the median recovery angular error over the dataset. The recovery angular error, the most commonly used in the literature [83][82], is the angle between the actual and estimated illuminants. The reproduction angular error [10] is a more stable measure across illumination changes (see Section 2.4.2 for more details).

In Table 4.1, the 23 algorithms are ranked according to the median recovery angular error. On one hand, the ranking is the same for REC and SFU, although the median errors are slightly different. This is expected as (recall Figure 4.2) the chromaticity points in REC and SFU are very close.

On the other hand, the ranking of algorithms with Gt1 is considerably different. The five best algorithms with Gt1 are not in the top five according to REC. Vice versa, the five best algorithms according to REC are among the worst-performing algorithms with Gt1.

Fast Fourier Colour Constancy [36] – which is in significant part built on top of a machine learning algorithm – is best according to the REC ground-truth but is the 15th best with Gt1. This may not be surprising as this algorithm was trained on SFU (similar to REC).

Table 4.1 Ranking of 23 algorithms in terms of median recovery error for REC versus SFU versus Gt1; the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [6].

Algorithm	REC		SFU		Gt1	
	median	rank	median	rank	median	rank
Fast Fourier Color Constancy (model Q) [36]	1.13°	1	1.13°	1	3.82°	15
1st order Grey-Edge (p=1, σ=6) [54]	3.09°	2	3.08°	2	4.52°	18
Edge-based Gamut (σ=4) [59]	3.27°	3	3.26°	3	5.04°	19
2nd order Grey-Edge (p=1, σ=1) [54]	3.57°	4	3.59°	4	4.44°	17
Bayesian [32][52]	3.85°	5	3.87°	5	3.46°	14
Deep Color Constancy using CNNs [34]	3.94°	6	3.93°	6	1.99°	1
Exemplar-based Colour Constancy [62]	4.36°	7	4.37°	7	2.27°	2
Pixel-based Gamut (σ=4) [59]	4.41°	8	4.43°	8	2.33°	3
Intersection-based Gamut (σ=4) [59]	4.41°	9	4.43°	9	2.34°	4
Bottom-up + Top-down [58]	4.54°	10	4.55°	10	2.47°	5
Top-Down [58]	4.60°	11	4.62°	11	2.63°	7
Using Natural Image Statistics [33]	4.70°	12	4.72°	12	3.13°	11
Heavy Tailed-based Spatial Correlations [61]	4.76°	13	4.80°	13	2.96°	9
Bottom-Up [58]	4.90°	14	4.90°	14	2.56°	6
CART-based Selection [56]	5.09°	15	5.09°	15	3.35°	12
Alex-Net + SVR using CNNs [34]	5.30°	16	5.30°	16	3.11°	10
CART-based Combination [56]	5.52°	17	5.52°	17	2.91°	8
General Grey-World (p=9, σ=9) [54]	5.95°	18	5.95°	18	3.46°	13
White-Patch [28]	6.74°	19	6.74°	19	5.68°	20
Shades-of-Grey (p=4) [30]	6.83°	20	6.85°	20	4.01°	16
Regression (SVR) [60]	9.60°	21	9.62°	21	6.73°	22
Grey-World [29]	9.97°	22	9.99°	22	6.28°	21
Inverse-Intensity Chromaticity Space [41]	11.61°	23	11.58°	23	11.52°	23

Table 4.2 Ranking of 23 algorithms in terms of median reproduction error [10] for REC versus SFU versus Gt1; the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [6].

Algorithm	REC		SFU		Gt1	
	median	rank	median	rank	median	rank
Fast Fourier Color Constancy (model Q) [36]	1.43°	1	1.45°	1	4.75°	16
1st order Grey-Edge (p=1, σ=9) [54]	3.74°	2	3.77°	2	4.90°	18
2nd order Grey-Edge (p=1, σ=1) [54]	4.54°	3	4.59°	4	4.76°	17
Edge-based Gamut (σ=3) [59]	4.56°	4	4.54°	3	5.88°	19
Bayesian [32][52]	4.60°	5	4.62°	5	3.92°	13
Deep Color Constancy using CNNs [34]	4.72°	6	4.75°	6	2.24°	1
Exemplar-based Colour Constancy [62]	5.17°	7	5.22°	9	2.64°	2
Pixel-based Gamut (σ=4) [59]	5.20°	8	5.21°	7	2.73°	3
Intersection-based Gamut (σ=4) [59]	5.20°	9	5.21°	8	2.74°	4
Bottom-up + Top-down [58]	5.36°	10	5.35°	10	2.75°	5
Top-Down [58]	5.41°	11	5.43°	11	3.11°	7
Using Natural Image Statistics [33]	5.46°	12	5.47°	12	3.55°	10
Heavy Tailed-based Spatial Correlations [61]	5.83°	13	5.88°	13	3.48°	8
Bottom-Up [58]	5.85°	14	5.89°	14	2.98°	6
CART-based Selection [56]	6.08°	15	6.03°	15	3.90°	12
Alex-Net + SVR using CNNs [34]	6.34°	16	6.38°	16	3.65°	11
CART-based Combination [56]	6.56°	17	6.55°	17	3.48°	9
General Grey-World (p=9, σ=9) [54]	6.66°	18	6.65°	18	3.98°	14
Shades-of-Grey (p=4) [30]	7.55°	19	7.56°	19	4.43°	15
White-Patch [28]	8.04°	20	8.03°	20	6.48°	20
Regression (SVR) [60]	10.31°	21	10.39°	21	7.42°	22
Grey-World [29]	10.63°	22	10.68°	22	6.81°	21
Inverse-Intensity Chromaticity Space [41]	12.81°	23	12.77°	23	12.77°	23

Deep Colour Constancy using CNNs [34] is 6th based on REC and top ranked based on Gt1. Again, this is not surprising since this algorithm was trained on Gt2 (which, we recall, is very similar to Gt1 but very different from SFU and REC).

In Table 4.2, we consider the ranking of the 23 illuminant estimation algorithms according to the median reproduction angular error[10]. This time, results with REC and SFU locally differ. The algorithms 2nd order Grey-edge and Edge-based Gamut are in reverse order. Exemplar-based Colour Constancy, Pixel-based and Intersection-based Gamut methods are also in different orders when using REC versus SFU. Once again, the results with Gt1 are significantly different. The reader can also notice that some algorithms have different rankings for the recovery error versus the reproduction error when the same ground-truth is used. This is related to the characteristics of the two angular measures [10] and shows again that the choice of the metric has also an effect on the evaluation (cf Chapter 2).

Table 4.3 shows that rankings of six algorithms are reversed when using REC versus the legacy ground-truth Gt1 in terms of median recovery angular error. Gt1 is incorrect but it was widely used, in particular in the evaluation of algorithms on colorconstancy.com and in [97] which was cited in the big majority of past research in the field. For example, the algorithms Edge-based Gamut Mapping [51][59], 2nd order Grey-Edge [54] and Bayesian [52][32] are in reverse order.

Table 4.3 The performance of 6 algorithms are in reverse order in terms of their median recovery error when the new REC versus the legacy Gt1 ground-truth is used. Note the Minkowski norm p and the smoothing value σ (standard deviation of the Gaussian filter) are the optimal parameters [7].

Algorithm	REC	Gt1
Edge-based Gamut (σ=4)	3.27°	5.04°
2nd order Grey-Edge (p=1, σ=1)	3.57°	4.44°
Bayesian	3.85°	3.46°
Using Natural Image statistics	4.70°	3.13°
Heavy Tailed-based Spatial Correlations	4.76°	2.96°
Bottom-Up	4.90°	2.56°

4.3 Conclusion

In this chapter we introduced a new ground-truth for the ColorChecker dataset. We called the new ground-truth set REC for recommended ground-truth. We presented the calculation steps of REC with our code based on the methodology described by Shi and Funt on the sfu web-site. We also re-processed the Raw images following Shi and Funt methodology and generated a new set of linear images. We made the new data available on the colorconstancy.com web-site for the use in colour constancy, when comparing a new illuminant estimation method to the state of the art methods on the ColorChecker dataset or when using the existing rankings and performance evaluations relative to this dataset.

In this chapter, we also compared the REC set to the existing ones that we labelled Gt1, Gt2 and SFU. These three sets have multiple calculation problems that we presented in detail in the previous chapter. The comparison showed with no surprise that the SFU ground-truth is the closest to the newly-recalculated REC set, and that Gt1 and Gt2 are very different from REC (mainly due to the ‘black level’ offset which was not subtracted from the images). We re-evaluated 23 illuminant estimation methods on the ColorChecker dataset using the REC ground-truth and we presented the results of performance evaluation and rankings expressed in terms of angular errors. We compare these results with the ones using the other sets of ground-truth and show that there are considerable differences for the same algorithms in performance evaluations and rankings, which can be as dramatic as a reversed ranking.

Of course, based on these results and considering the re-processing of the images and the calculation methodology of the REC ground-truth, we invite the community to refer in future research to what we hope is a more definitive data and performance evaluation on the ColorChecker dataset.

Chapter 5

Monge-Kantorovitch Transform for Chromagenic Colour Correction

In this chapter we present our work on the chromagenic illuminant estimation method. This method has two stages: a calibration stage and a test stage and it is characterised by using two images of the same scene, one captured with a coloured filter and one without a filter, to solve for the illuminant colour. The relationship between the unfiltered image RGBs and the filtered counterparts is described with a linear transform and depends on both the scene illuminant and the chromagenic filter. The original chromagenic method is based on the least-squares transform (LST) which requires registering the images for the calibration and testing.

We revisit the chromagenic algorithm and propose to use instead of LST the Monge-Kantorovitch transform (MKT), to solve for the illuminant colour when the images are not registered. In fact, registration is complex and computationally expensive and with the MKT approach, the images do not need to be registered to perform chromagenic colour constancy.

In what follows we present first the original chromagenic method as introduced by Finlayson et al. [45][47][48]. Then we explain what is MKT and how it can replace LST. Then we introduce two new datasets of chromagenic RGB images and we evaluate the new MKT approach on these datasets. We compare the MKT and LST results and we show that using MKT outperforms LST on the first step of the chromagenic method.

To be clear, we do not re-evaluate the whole chromagenic estimation method but rather validate that the MKT transform facilitates the calculation of the linear transform relating an image to its filtered counterpart, a key element of the chromagenic approach.

5.1 Chromagenic Illuminant Estimation

Chromagenic illuminant estimation was first introduced by Finlayson et al. [45][47][48]. This method uses two RGB images of the same scene one filtered and one unfiltered. There exist other colour constancy methods based on the use of two images of the same scene, e.g. like a standard and an IR images in [106], but chromagenic uniquely uses an unfiltered and filtered pair.

In chromagenic colour constancy, the filtered image is obtained by placing a coloured filter in front of the camera objective. A filtered image of the scene is similar to the scene seen under a different illuminant. The filter has to have a discriminative power, which means that it leads to transforms as different as possible to discriminate between different illuminants. In other terms, the filter should maximise the inter-transforms variance, while yielding transforms that ensure a good mapping between the filtered images and the unfiltered counterparts.

Finlayson et al. in [107][11] showed that a suitable coloured filter that satisfies these properties should not cut off any of the colour channels (which is the case for example for the Kodak Wratten green saturated filter[108]) and should have a transmittance with enough variance across the spectrum. The filter cannot be a neutral density (uniform transmittance) which leads to a constant relation between the unfiltered and filtered images independent of the illuminant. Suitable filters candidates are presented in Figure 5.1.

The chromagenic illuminant estimation has two stages: a calibration stage and a test stage. The calibration is performed by calculating a set of transforms using a calibration images set. When testing on a new pair of images $\mathbf{I}_{NF}$ and $\mathbf{I}_{F}$ (respectively, the superscripts denote no-filter and filtered images) to determine the scene illuminant, we then look for the best transform $\mathbf{T}_{est}$ that best describes the relationship between the two images ($||\mathbf{I}_{F} - \mathbf{I}_{NF}\mathbf{T}_{est}||$ is minimum).

Having two images ($M \times N$ pixels) of the same scene allows computing a transform matrix $\mathbf{T}$ which describes the relation between the unfiltered image and its filtered counterpart. In other terms, this matrix maps the unfiltered RGBs to the filtered RGBs. In [47], it was shown that a 3×3 least-squares transform (LST) is a good solution (more complex mappings did not improve the efficiency of the algorithm in solving for the illuminant). The linear transform matrix, denoted $\mathbf{T}_{LST}$, is calculated using the Moore-Penrose inverse as follows:

$$\mathbf{T}_{LST} = \mathbf{I}_{NF}{}^{+}\mathbf{I}_{F} \tag{5.1}$$

where $\mathbf{I}_{NF}$ and $\mathbf{I}_{F}$ denote the 2-d $MN \times 3$ RGBs responses for respectively the unfiltered and filtered images (where every column has $M \times N$ R, G or B values of all the image pixels) and the superscript $^{+}$ denotes the Moore-Penrose inverse. In what follows we always use 2-d $MN \times 3$ matrices in the calculations.

The method requires a calibration step. The calibration is specific for a pair of a camera and a filter. Let us explain the calibration in more detail. First a set of m transforms matrices $\{\mathbf{T}_i\}$ is calculated from a set of m chromagenic pairs of images, for m lights. Every transform $\mathbf{T}_i$ is associated to an illuminant colour vector $\underline{e}_i = [r_i \; g_i \; b_i]$ which is the ground-truth illuminant of the related scene. Then this set of transforms and illuminant colours is used to estimate the illuminant on a test image (having a chromagenic pair of images). A least-squares regression is performed and the best matching matrix $\mathbf{T}_{best}$ is selected, by solving the minimisation problem according to Equation 5.2. The choice of the matrix gives the estimate $\underline{e}_{est}$, which is the illuminant associated with it at the prior step.

$$\mathbf{T}_{est} = argmin_{T_i} ||\mathbf{I}_{NF}\mathbf{T}_i - \mathbf{I}_F|| \tag{5.2}$$

Then like for all other illuminant estimation methods, the output of the chromagenic algorithm is 'divided-out' (from the non-filtered image) to remove the effect of the illuminant colour from the scene. That is if $\underline{\rho}$ is an rgb and $\underline{e}_{est}$ is the rgb of the estimated light then we calculate $\underline{\rho}/\underline{e}_{est}$, see [24] for more details on why this dividing out is the correct way to remove the colour bias due to illumination.

The least-squares based chromagenic method can deliver good results, e.g. the chromagenic gamut mapping [47] and the bright chromagenic algorithm [11]. One constraint of the chromagenic method, that we can perceive from Equation 5.2 is the need for well aligned images. Deriving the transform from Equation 5.1 only makes sense when there is a pixel-correspondence between the two images, and this is required for the pairs of images of both the calibration and the test stages. However registration is a computationally expensive processing. In fact, it is still an important field of research [109] and cannot always be reliably achieved.

Based on this fact, we propose in the next section the use of a different transform, the Monge-Kantorovitch transform (MKT), to solve for the transforms used in chromagenic illuminant estimation without the need for registering the images [9].

5.2 Monge-Kantorovitch Transform for Chromagenic Illuminant Estimation

5.2.1 The Monge-Kantorovitch Transform

We propose to solve for the linear transforms used in chromagenic colour constancy using the Monge-Kantorovitch transform (MKT). MKT was used in colour research in the last years for problems like for color-based similarity [110], colour transfer [111], image segmentation [112] or to allow unpenalised distributions transformations [113].

MKT is derived from the Monge-Kantorovitch theory of mass transportation [12]. Imagine we have a few piles of earth and equal to the volume of all the earth we have a set of holes. We would like to transfer the earth to fill the holes in a way that minimises the transportation energy. This problem is known as the mass preserving transport problem. Clearly we need to move each pile of earth as little as possible. This minimum distance for moving the earth is the Earth Mover's Distance [114] (or the Wasserstein-Kantorovitch Distance [115][116]).

According to [117], there is a unique solution for the linear mapping **T** between two distributions that minimises the displacement cost. The mapping matrix is a product of the covariance matrices of these distributions, and by being the product of symmetric matrices, MKT is also a symmetric matrix.

From an image point of view, we would like to find a continuous linear mapping to match distributions of the input (non-filtered image) with the filtered image and have as little change in their distributions as possible. The Monge-Kantorovitch transform – a 3 × 3 matrix – is an optimal solution to this linear restriction of the transportation problem.

5.2.2 Monge-Kantorovitch for Colour Correction

We chose the linear Monge-Kantorovitch transform (MKT) as a candidate to solve for colour correction when images are not registered. The colour correction is part of the calibration step of the chromagenic illuminant estimation algorithm. If we capture two images of a scene, an unfiltered image $\mathbf{I}_{NF}$ and a filtered image $\mathbf{I}_F$ using a coloured filter then we can calculate the 3 × 3 linear matrix as explained above in Section 5.1. The linear MKT maps the unfiltered image to its filtered counterpart. The MKT linear transform, denoted $\mathbf{T}_{MKT}$, is solved by minimizing:

$$\mathbf{T}_{MKT} = \min_{T} ||\mathbf{I}_{NF}\mathbf{T} - \mathbf{I}_{NF}||\, s.t. \mathbf{T}^{t}\mathbf{I}_{NF}^{t}\mathbf{I}_{NF}\mathbf{T} = \mathbf{I}_{F}^{t}\mathbf{I}_{F} \tag{5.3}$$

where ^{t} denotes the matrix transpose operator and $||||$ is the Euclidean distance (L_2 norm)

The least-squares transform (LST), denoted $\mathbf{T}_{LST}$, is obtained by minimising the Euclidean distance between the colour-corrected $\mathbf{I}_{NF}\mathbf{T}$ and the filtered image $\mathbf{I}_F$, in other terms LST minimises the amount of changes between the images colours.

$$\mathbf{T}_{LST} = \min_T ||\mathbf{I}_{NF}\mathbf{T} - \mathbf{I}_F|| \tag{5.4}$$

By contrast the MKT minimises the distance between $\mathbf{I}_{NF}\mathbf{T}$ and $\mathbf{I}_{NF}$, i.e no change, but with the additional constraint that we seek to have match the covariance structure of $\mathbf{I}_{NF}\mathbf{T}$ and $\mathbf{I}_F$. The covariance structure can be represented in more ways and we will consider another variation which is the autocorrelation (we will present results for the two cases in Section 5.4).

By analogy to the earth movers distance $||\mathbf{I}_{NF}\mathbf{T} - \mathbf{I}_{NF}||$ is the amount the data which we move and the covariances or autocorrelations match captures the idea that the data should occupy the target 'holes' (this is visualisation rather than an exact analogy).

The covariance is a 3×3 matrix calculated as follows:

$$\mathbf{\Sigma}_c = E[(\mathbf{I} - \mu_I)^t(\mathbf{I} - \mu_I)] = E[\mathbf{I}^t\mathbf{I}] - \mu_I\mu_{I^t} \tag{5.5}$$

where ^{t} denotes the matrix transpose operator, $E[]$ is the expected value of a matrix which has the expected values of every elements of the matrix and $mu_I = E[\mathbf{I}]$ is the expected value of $\mathbf{I}$,

The autocorrelation is a 3×3 matrix calculated as follows:

$$\mathbf{\Sigma}_a = E[\mathbf{I}^t\mathbf{I}] \tag{5.6}$$

where ^{t} denotes the matrix transpose operator and $E[]$ is the expected value of a matrix.

For the computation of MKT, we used the code of [111]. The MKT transform is calculated in Equation 5.7 using covariances of the images and in Equation 5.8 using the autocorrelation of the images.

$$\mathbf{T}_{MKT_c} = \mathbf{\Sigma}_{c_{NF}} \mathbf{\Sigma}_{c_F}^{\frac{1}{2}} (\mathbf{\Sigma}_{c_F}^{\frac{1}{2}} \mathbf{\Sigma}_{c_{NF}} \mathbf{\Sigma}_{c_F}^{\frac{1}{2}})^{-\frac{1}{2}} \mathbf{\Sigma}_{c_F}^{\frac{1}{2}} \tag{5.7}$$

$$\mathbf{T}_{MKT_a} = \mathbf{\Sigma}_{a_{NF}} \mathbf{\Sigma}_{a_F}^{\frac{1}{2}} (\mathbf{\Sigma}_{a_F}^{\frac{1}{2}} \mathbf{\Sigma}_{a_{NF}} \mathbf{\Sigma}_{a_F}^{\frac{1}{2}})^{-\frac{1}{2}} \mathbf{\Sigma}_{a_F}^{\frac{1}{2}} \tag{5.8}$$

where $\mathbf{\Sigma}_{c_{NF}}$ and $\mathbf{\Sigma}_{c_F}$ are the covariance matrices of the non-filtered and filtered images, $\mathbf{\Sigma}_{a_{NF}}$ and $\mathbf{\Sigma}_{a_F}$ are the autocorrelation matrices of the non-filtered and filtered images and $\mathbf{\Sigma}^{\frac{1}{2}}$ the square root of the matrix.

So unlike the original transform LST, where the optimisation was about minimising in a least-squares sense the distance between the colour-corrected image $\mathbf{I}_{NF}T$ and the filtered image $\mathbf{I}_F$ matrices, with MKT the optimisation is about matching the statistical distributions of the two matrices while minimising in a least-squares sense the distance between the input image $\mathbf{I}_{NF}$ and the colour-corrected one $\mathbf{I}_{NF}\mathbf{T}$. For this reason, the alignments of the pixels of $\mathbf{I}_{NF}$ and $\mathbf{I}_F$ should not have a real impact on the results of the optimisation.

5.3 New Datasets for Chromagenic Colour Constancy

In order to evaluate the performance of the chromagenic illuminant estimation algorithm and compare it to other methods, we collected sets of chromagenic RGB images. This type of dataset is different from the available benchmark RGB datasets in the sense that every scene is captured twice, with and without a coloured filter. All captured scenes come in pairs of images. The set of unfiltered images from these datasets can of course be used in the evaluation of other non-chromagenic illuminant estimation methods. In what follows, we present the data collection procedure and we introduce two new chromagenic images datasets.

5.3.1 Dataset Collection

A chromagenic dataset comes in pairs of chromagenic RGB images. Every pair of images represents one scene captured with and without a coloured filter. For the data collection we used a Nikon D5200 camera with an 18-105mm lens, coupled with Tiffen coloured filters. We place the filter in front of the camera objective. We used two different filters: a 85 red filter and a 81EF yellow filter. The transmittance of these filters is shown in Figure 5.1. Both filters do not cut off completely any part of the visible spectrum. The 85 filter has a higher transmittance in the red part of the spectrum and the yellow filter has a smoother transmittance across the spectrum.

Chromagenic illuminant estimation method works with a single coloured filter but we used two of them in order to compare the results and choose the best filter for future chromagenic colour constancy applications. The two filters were used separately to collect the two sets of images.

For every scene, we take three images: one black image with the camera cap on, a second image without a filter and a third image with the coloured filter. The black image is used in discounting the black level from the image pixels. The Black image (the dark current signal image), captured with the camera lens cap on, is one of the noise factors in the rendered scene image [100]. It is simply subtracted from the two other images.

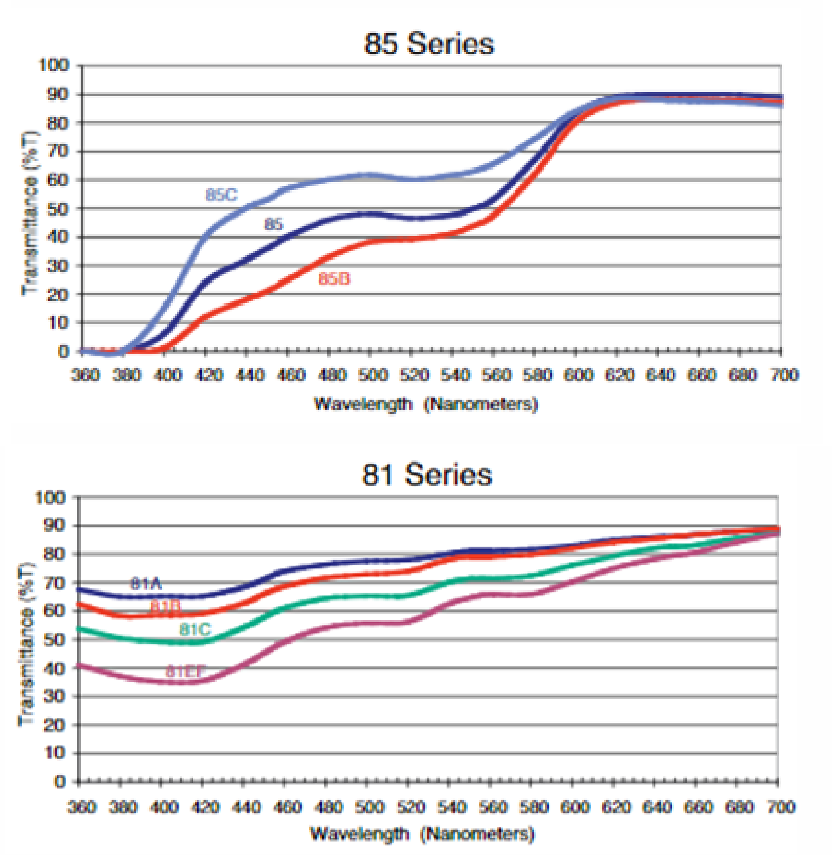


Fig. 5.1 Filters red 85 (upper graph) and yellow 81EF (lower graph, purple curve) transmittance in the visible spectrum. These filters were used in the collection of the chromagenic sets of images, from [8].

The three images of the same scene are taken consecutively to avoid any change in the light colour which can occur for example in outside scenes or scenes that contain shadows. Note, the intensity of the light – as long as it stays constant over the 3 captures – is not an important parameter, except for saturation (overexposure). This is because the linear transform relating filtered and unfiltered RGBs is independent of the intensity of the light. One challenging part of the data collection is to try to have only static objects in the scene and avoid objects like people walking or animals moving or cars in motion because we need to capture the same exact scene with and without a filter. Thus an extra precaution needs to be made based on that in particular when taking pictures outside. Our camera was mounted on a tripod to try and make the captured images ’as registered’ as possible (LST requires the images to be registered). All these elements makes the multi-illuminant dataset collection –in our case when using coloured filters– a long process as explained in [118]. In our case the data collection took approximately 10 minutes per scene.

A Macbeth ColorChecker chart is placed in every scene. We use it to determine the ground-truth, or correct answer of the illuminant colour in the scene, by calculating the RGB vector from the achromatic or grey patches of the colour chart. We used the calculation methodology of Shi and Funt [98], which is described in more detail in Section 4.1.2. The position of the colour chart in the scene is carefully chosen. It has to be in the part of the scene under the predominant light (not in the shadow if they are present in the scene).

One example of a scene with a minor shadow is shown in the 2$^{\text{nd}}$ row of Figure 5.2. There are some constraints when placing the ColorChecker: it needs to be facing the light source as well as the camera as much as possible in order to have an accurate illuminant colour measurement on the patches. In fact, by placing the chart parallel to the camera lens we optimise the reproduced patches surfaces in the image which makes the ground-truth illuminant calculation easier. The chart needs to be close enough to the camera without occupying a too large proportion of the scene, as the chart is masked when evaluating an illuminant estimation method.

We often used the camera automatic mode but we also sometimes used other modes like the exposure bracketing mode, sometimes called the High Dynamic Range (HDR) mode, which allows capturing the scene with different exposure levels to control saturation or overexposure: we capture a 3-image exposure stack per exposure bracketing value. Images are saved in a Raw format, then linearly processed with DcRaw [102] (no white-balance or gamma curve applied) and with a manual linear demosaicing with a simple interpolation of the Bayer pattern to produce linear JPEGs (3018×2010 pixels). A more detailed description of the images processing methodology is presented in Section 4.1.1 in Chapter 4.

5.3.2 UEA Chromagenic Dataset

The first collected dataset, the UEA (University of East Anglia) dataset, has two separate sub-sets of respectively 92 (red filter) and 86 pairs (yellow filter) of high quality RGB images.

The dataset was collected at different locations, mainly in Norwich in the UK but also Paris in France and Tunis in Tunisia. It has photographic images of indoor and outdoor scenes of various colour complexities and lights. Samples from the UEA dataset are presented in Figure 5.2. The images are the camera pipeline outputs (non-linear JPEG images) and the difference between the unfiltered image and the yellow-filtered counterpart is not so noticeable here due to the camera automatic white-balance (AWB). The illuminants of this dataset range from daylight to skylight and shadow and various artificial lights including LEDs lighting used in art galleries (see Figure 5.2).

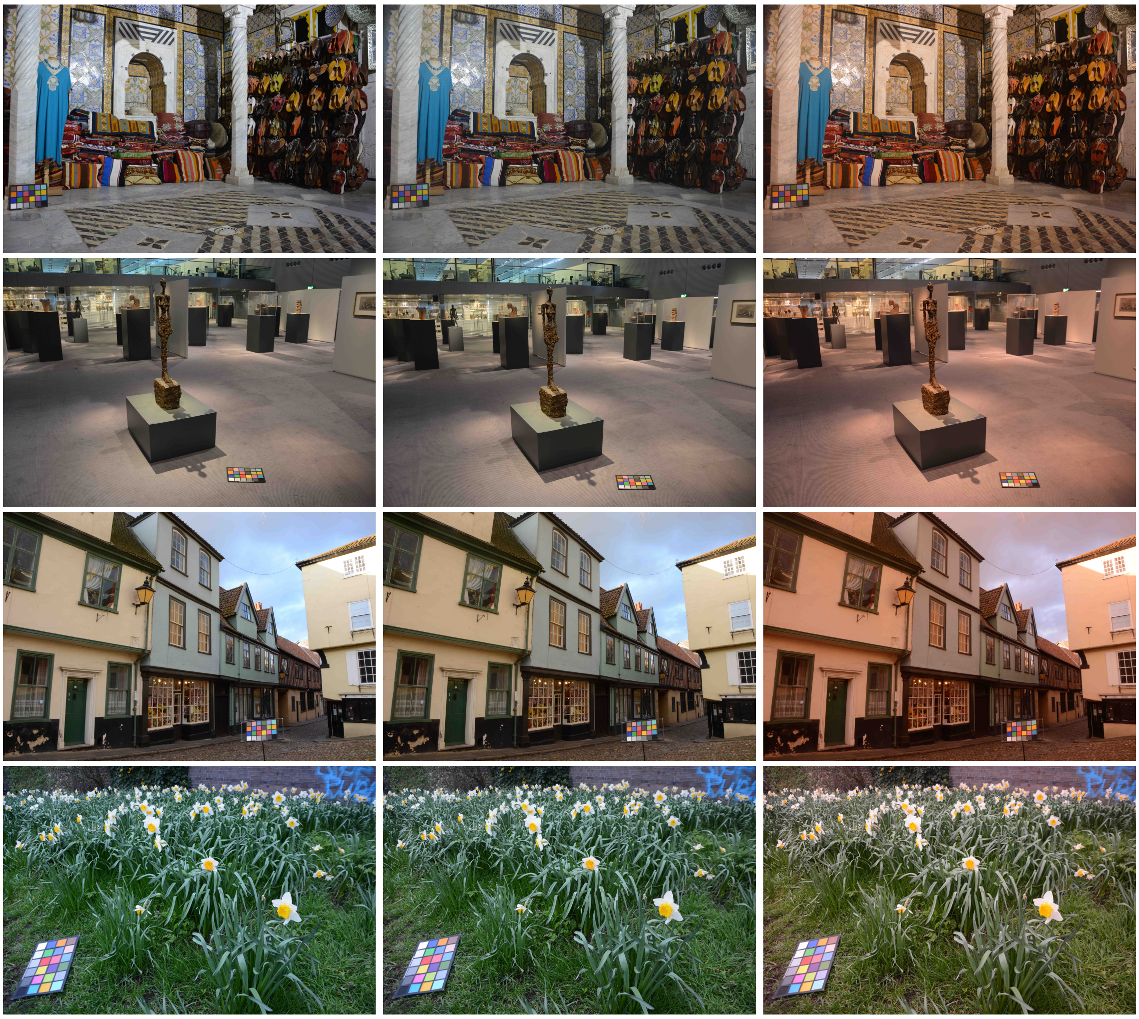


Fig. 5.2 Samples of indoor and outdoor scenes from the UEA chromagenic dataset. From left to right: unfiltered, yellow-filtered and red-filtered images. Notice the ColorChecker in every scene, and that these images are the camera pipeline outputs.

The rg chromaticity gamut of the ground-truth illuminants of the UEA dataset is shown in Figure 5.3. It shows a relatively sparse distribution which means that the dataset has a good variety of illuminants. The rgb chromaticities are normalised intensity-independent RGBs with $r = \frac{R}{R+G+B}$, the same for the chromaticities g and b.

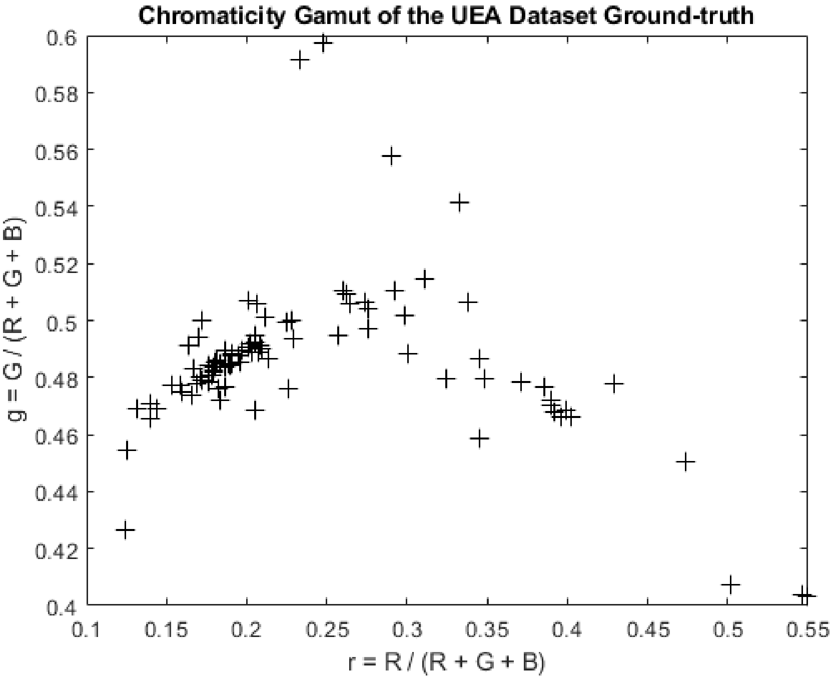


Fig. 5.3 Chromaticity gamut (r,g) of the UEA chromagenic dataset.

5.3.3 Face Images Chromagenic Dataset

The second collected chromagenic dataset is a set of facial images (697 × 549 pixels). We were interested in considering the colour constancy problem with respect to faces as we know that face colour is exploited in many contemporary smartphone cameras. An example of a face-based illuminant estimation algorithm is presented in [119].

We built our face database in Chiba University in Japan during the Summer of 2018. Nine participants from Tsumura labratory [120] were subjects in our experiments. The dataset has 63 scenes in total. Every scene represents a set of face images of a subject taken under a specific light. The images were captured in a lighting room (from Horiuchi & Hirai

Laboratory [121]) and various lighting conditions were simulated using two Thouslite LED cubes [122] to modulate the colour temperature.

We call our face dataset the Kampo dataset as this dataset is also part of a project on the Kampo image-based diagnosis system [123] being carried out in the Tusmura Lab (see details at the end of this chapter).

Figure 5.4 shows two images of the same scene from the Kampo dataset. The scene shows the whole capture environment and has a ColorChecker chart. As before we use the chart to calculate the ground-truth illuminant colour for every scene. The two images seem very well aligned, this is because the experiment conditions were very well controlled : use of a tripod and a head holder.

We crop the images like in Figure 5.5 and we only keep the face in the image for the evaluation of the method. The Figure shows two subjects faces, the unfiltered images and the yellow and red filtered counterparts. The difference between the unfiltered image and the yellow counterpart is not so noticeable here due to the camera automatic white-balance (JPEG images). The face image –similar to a selfie– is what is needed for a Kampo medical diagnosis based on the image.

The ground-truth illuminants chromaticity gamut of the Kampo images dataset is shown in Figure 5.6. It shows a relatively sparse distribution which means that the dataset has a good variety of illuminants.

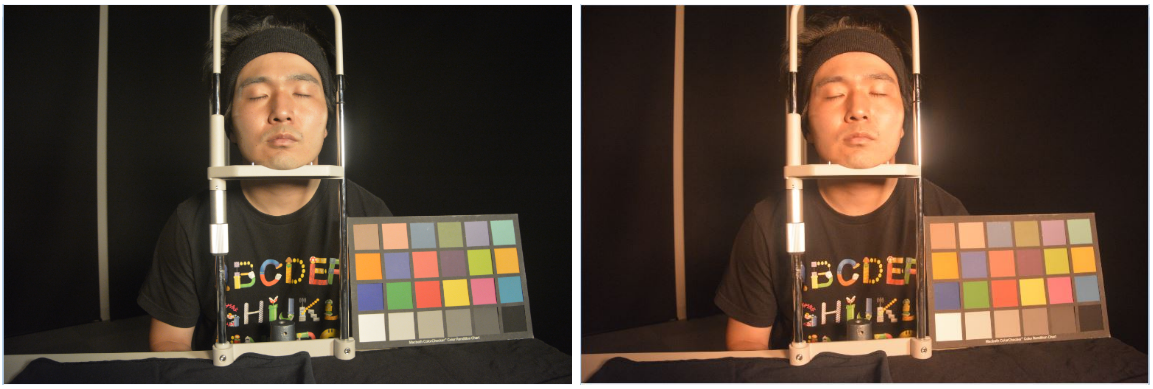


Fig. 5.4 Two images of the same scene from the dataset showing the ColorChecker chart, the left image is a normal capture and the right image was captured through a red filter, note that these 2 images are the camera pipeline outputs, from [9].

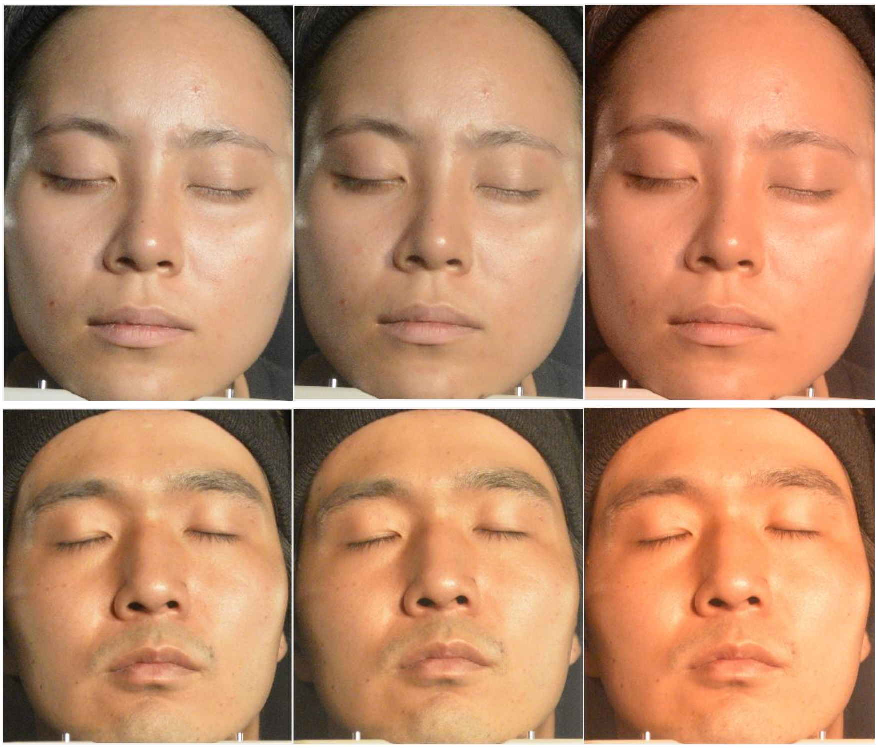


Fig. 5.5 Two sets of filtered and unfiltered images from the Kampo dataset. The 2 sets represent 2 scenes (2 subjects), from left to right: normal capture, image with yellow filter and image with red filter. The images are the camera pipeline outputs, from [9].

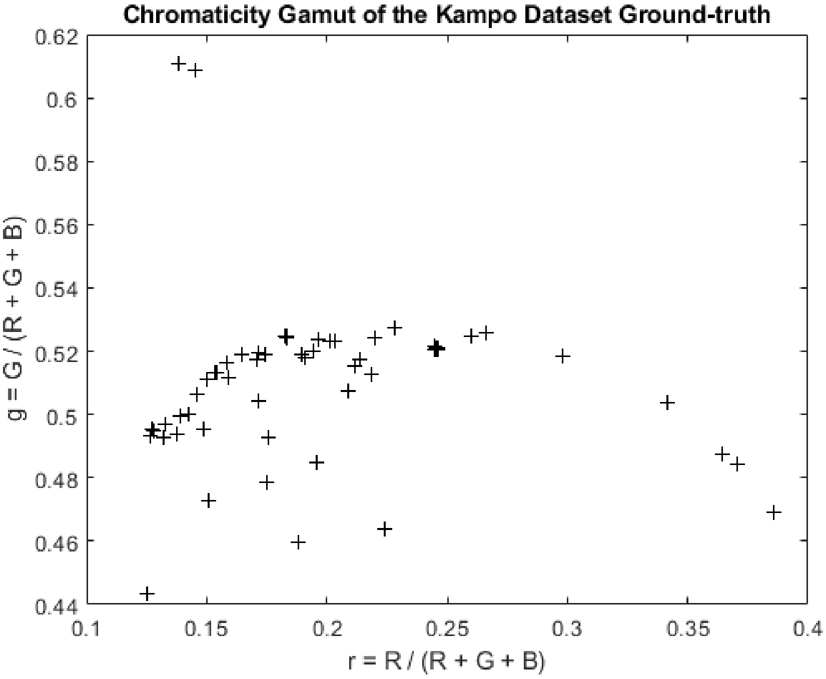


Fig. 5.6 Ground-truth chromaticities of the Kampo images dataset.

5.4 Experiments and Results

5.4.1 Experiments

Now, let us evaluate the effectiveness of our approach on the two chromagenic datasets: the UEA dataset of (sub-sets of 92 and 86 scenes) and the Kampo dataset which has 63 scenes, for the red and yellow-filtered images separately. We compare the results of the calibration step of the chromagenic illuminant estimation algorithm when using the Monge-Kantorovitch transform (MKT) considering both covariance and autocorrelation versus the least-squares transform (LST) method, where, for the absence of doubt, we are not registering the images.

Calculating the transform matrix that linearly maps the non-filtered image to the filtered counterpart is equivalent to solving a colour correction problem.

We are going to compare the colour correction afforded by an LST and an MKT against a ground-truth correction, computed by using the RGBs from the Macbeth Colorchecker in the scene. We extract images of the colour charts by cropping the original images. Then, we measure the 24×3 colour matrix $\mathbf{I}_{cc}$ using the 24 ColorChecker patches for the unfiltered and filtered images and we calculate the 3×3 least-squares transform $\mathbf{T}_{cc}$ which maps the unfiltered ColorChecker image $\mathbf{I}_{cc_{NF}}$ to its filtered image $\mathbf{I}_{cc_F}$ of the same scene, see Equation 5.9. The ground-truth best case colour correction afforded by a linear transform is calculated as:

$$\mathbf{I}_{cc_F} = \mathbf{I}_{cc_{NF}} \mathbf{T}_{cc} \tag{5.9}$$

Now let us compare the MKT transform and the LST transform against the ground-truth. LST is calculated by assuming the images are in pixel-wise registration (they aren't) and then calculating the Moore-Penrose inverse. The MKT transform is calculated in Equation 5.7 using covariances of the images and in Equation 5.8 using the autocorrelation of the images.

So, how do we evaluate how well the LST and MKT transforms work. Well, for the ground-truth method, we take the unfiltered image and multiply it by the colour correction transform calculated using the Macbeth ColorChecker images. That is we know what the corrected image should look like (and this may be slightly different than the actual filtered image). We now multiply the unfiltered image by the LST and MKT transforms. Compared to the ground-truth we calculate the error as:

$$\varepsilon = ||\mathbf{IT}_{cc} - \mathbf{IT}|| \, , \mathbf{T} \in \{\mathbf{T}_{LST}, \mathbf{T}_{MKT}\} \tag{5.10}$$

where $\mathbf{IT}_{cc}$ is the ground-truth best case colours corrected image using the ground-truth transform $\mathbf{T}_{cc}$ and $\mathbf{IT}$ is the result colour corrected image with MKT or LST. We calculate summary statistics using the set of ε for every dataset.

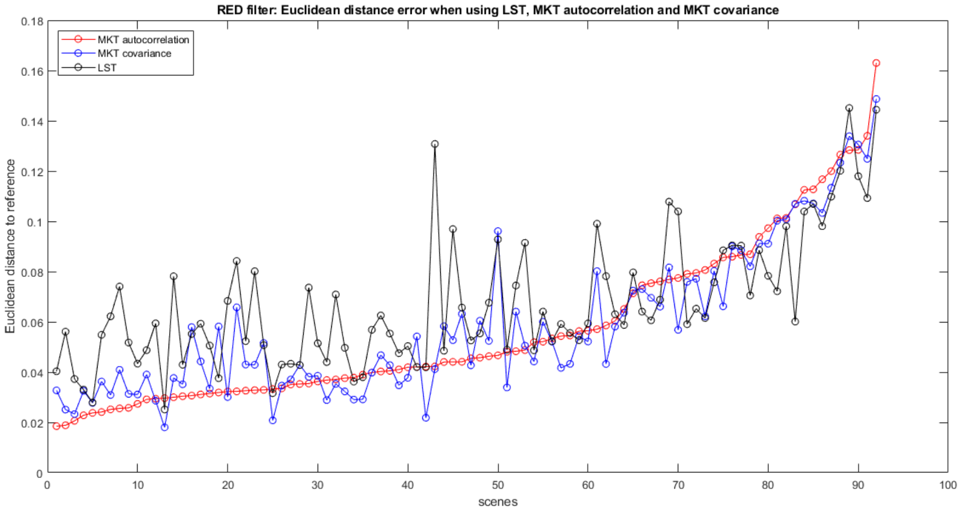
5.4.2 Results on the UEA Dataset

Figure 5.7 shows the distance error ε for the set of m chromagenic pairs of images expressing the effectiveness of the colour correction with LST versus the two MKT approaches based on the covariance and autocorrelation of the images with non-registered images for every scene from the dataset. The lower is the error ε, the more effective is the colour correction. The upper graph corresponds to the use of the red filter and the lower one to the yellow filter.

Note that the images pixels values are in the interval $[0,1]$ so a distance error of 0.01 corresponds to 1% of the image. The distance errors vary in the range $[0.015, 0.018]$. LST gives the largest errors for a larger number of scenes compared to the MKT approaches. The autocorrelation-based MKT gives the lowest errors for a larger number of scenes. The results with the two MKT approaches are comparable and different from the LST results and this is true in particular for the yellow filter.

Table 5.1 presents different summary statistics relative to the Euclidean distance errors $\{\varepsilon\}$ plotted in the previous graphs 5.7 for every pair of chromagenic images. The summary statistics include the root mean square (RMS) and the standard deviation (SD). In terms of median distance error, MKT autocorrelation gives better results than MKT covariance. The two filters give very similar results for this dataset, although the red filter seems to generate slightly smaller errors. In terms of RMS and mean distance errors, MKT autocorrelation works slightly better than MKT covariance. MKT covariance is more stable (lowest maximum and %95 quantile errors).

Visual examples of the results with the red filter are presented in Figures 5.8 and 5.10 where we show from left to right: the image corrected using the ColorChecker (the ground-truth i.e. the best possible solution), the image corrected using MKT covariance then MKT autocorrelation (our new approaches), and the image corrected with the LST (the original method). Our approach generates colors that are closer to the best possible solution. We show in Figure 5.9 the convex hulls in the chromaticity colour space for the first example (first row in Figure 5.8) for the output image with the best transform, MKT covariance and LST. The gamut of the MKT colour-corrected image is larger and has a closer surface to the best possible answer. We present similar results relative to the yellow filter in Figure 5.10 and Figure 5.11.



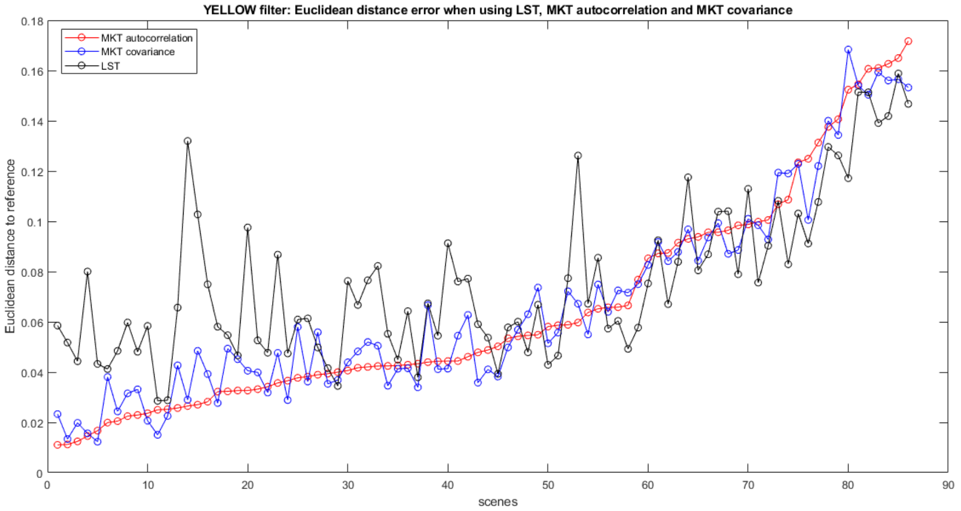


Fig. 5.7 UEA dataset: Euclidean distance when using MKT autocorrelation versus MKT covariance versus LST. Top: red filter. Bottom: yellow filter.

Table 5.1 UEA dataset: summary statistics for MKT versus LST approaches with the red filter (upper table) and the yellow filter (lower table) on the UEA dataset.

Red filter	**RMS**	**Mean**	**Median**	**σ = SD**	**Max**	**Range** = **max - min**	**95% Quantile**
LST	0.0721	0.0675	0.0604	0.6920	0.1451	0.1200	0.1171
MKT autocorrelation	0.0651	0.0570	0.0448	0.6249	0.1630	0.1445	0.1259
MKT covariance	0.0654	0.0584	0.0519	0.6268	0.1487	0.1306	0.1223

Yellow filter	**RMS**	**Mean**	**Median**	**σ = SD**	**Max**	**Range** = **max - min**	**95% Quantile**
LST	0.0797	0.0713	0.0662	0.7641	0.1588	0.1588	0.1416
MKT autocorrelation	0.0749	0.0607	0.0444	0.7183	0.1717	0.1717	0.1601
MKT covariance	0.0750	0.0624	0.0510	0.7195	0.1683	0.1683	0.1539

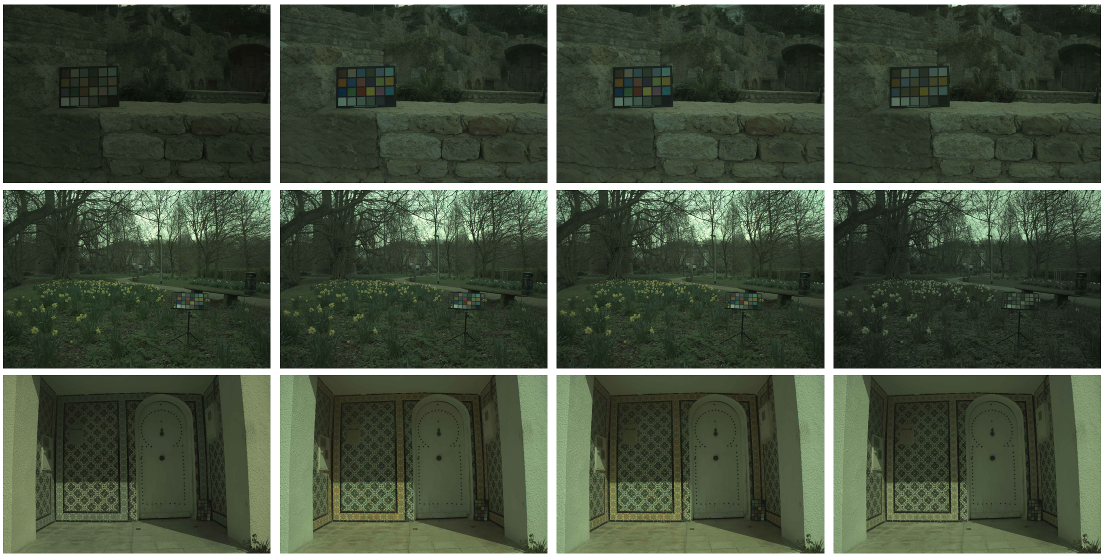


Fig. 5.8 UEA dataset: colour-corrected images for red filter. From left to right: best answer using T_{cc}, (the ColorChecker-based transform), output using MKT covariance, MKT autocorrelation and LST. Images are gamma corrected for a better comparison ($\sigma = 2.2$).

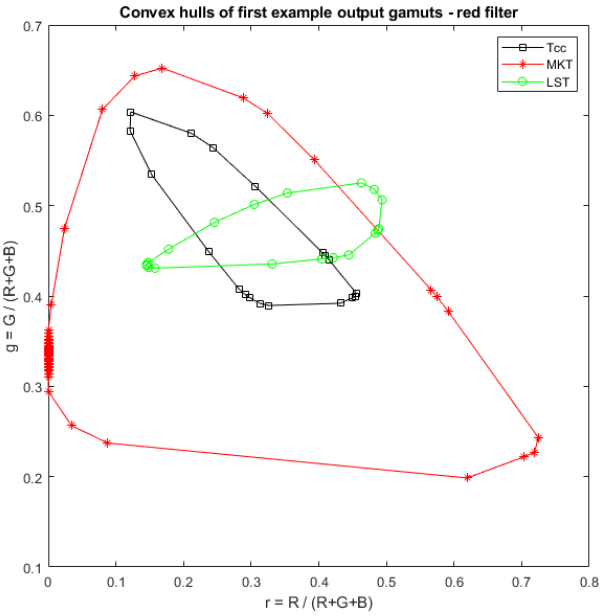


Fig. 5.9 UEA dataset: Convex hulls of first example output gamuts (with red filter) for T_{cc} versus MKT covariance versus LST.

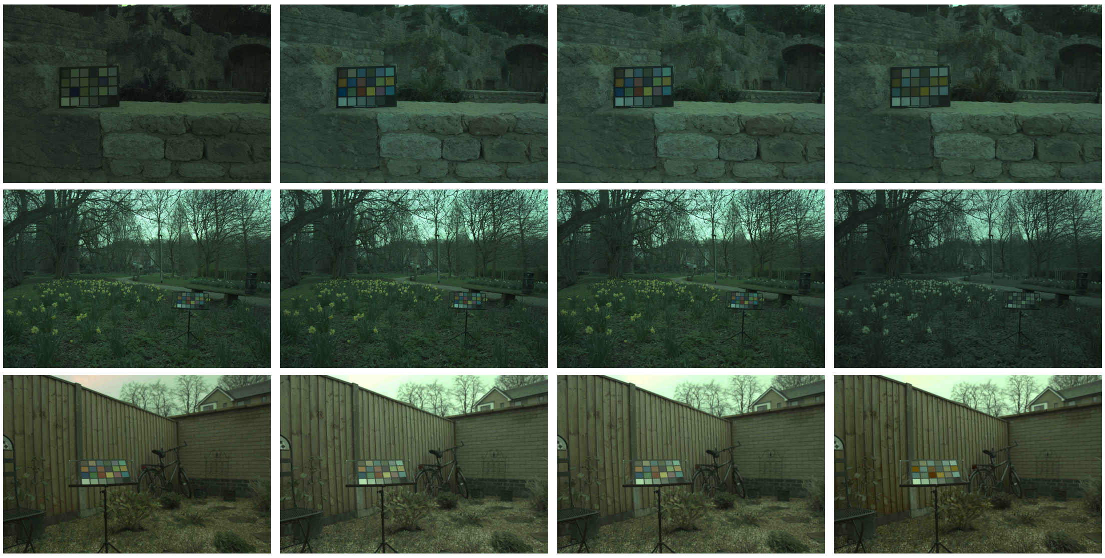


Fig. 5.10 UEA dataset: colour-corrected images for yellow filter. From left to right: best answer using T_{cc}, (the ColorChecker-based transform), output using MKT covariance, MKT autocorrelation and LST. Images are gamma corrected for a better comparison ($\sigma = 2.2$).

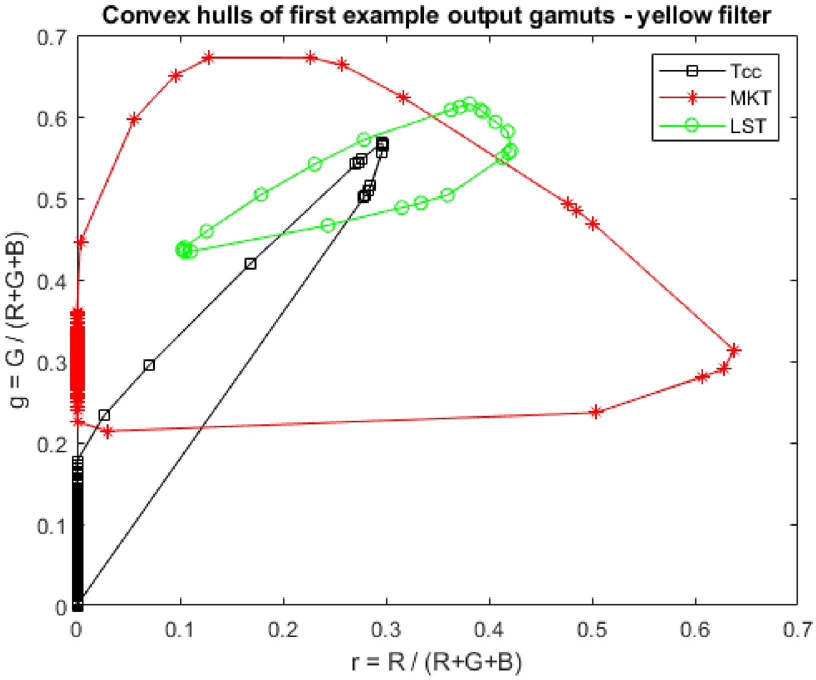


Fig. 5.11 UEA dataset: convex hulls of first example output gamuts (with yellow filter) for T_{cc} versus MKT covariance versus LST.

5.4.3 Results on the Kampo Dataset

Figure 5.12 shows the distance error ε for the m chromagenic pairs of images expressing the effectiveness of the colour correction with LST versus the two MKT approaches based on the covariance and autocorrelation of the images with non-registered images for every scene from the dataset. The lower is the error ε, the more effective is the colour correction. The upper graph corresponds to the use of the red filter and the lower one to the yellow filter.

Note that the images pixels values are in the interval $[0,1]$ so a distance error of 0.01 corresponds to 1% of the image. The errors vary in the range $[0.001, 0.014]$. LST gives the largest distance errors for a larger number of scenes compared to the MKT approaches. The autocorrelation-based MKT is best performing with the red filter and the covariance-based MKT is best performing with the yellow filter. The results with the two MKT approaches are comparable and different from the LST results with both filers.

Table 5.2 presents different summary statistics relative to the Euclidean distance errors $\{\varepsilon\}$ plotted in the previous graphs 5.12 for every pair of chromagenic images. The summary statistics include the root mean square (RMS) and the standard deviation (SD). In terms of median distance error, MKT autocorrelation gives better results than MKT covariance and the red filter seems to be more appropriate for this dataset. In terms of RMS and mean distance errors, MKT covariance works better especially when the red filter is used. The maximum errors prove that MKT autocorrelation can fail where MKT covariance is more stable.

Visual examples of the results with the red filter are presented in Figure 5.13 and 5.15 where we show from left to right: the image corrected using the ColorChecker (the ground-truth i.e. the best possible solution), the image corrected using MKT covariance then MKT autocorrelation (our new approaches), and the image corrected with the LST (the original method). Our approach generates colors that are closer to the best possible solution. We show in Figure 5.14 the convex hulls in the chromaticity colour space for the first example (first row in Figure 5.13) for the output image with the best transform, MKT covariance and LST. Clearly, the gamut of the MKT colour-corrected image is much more similar to the best possible solution. We present similar results relative to the yellow filter in Figure 5.15 and Figure 5.16.

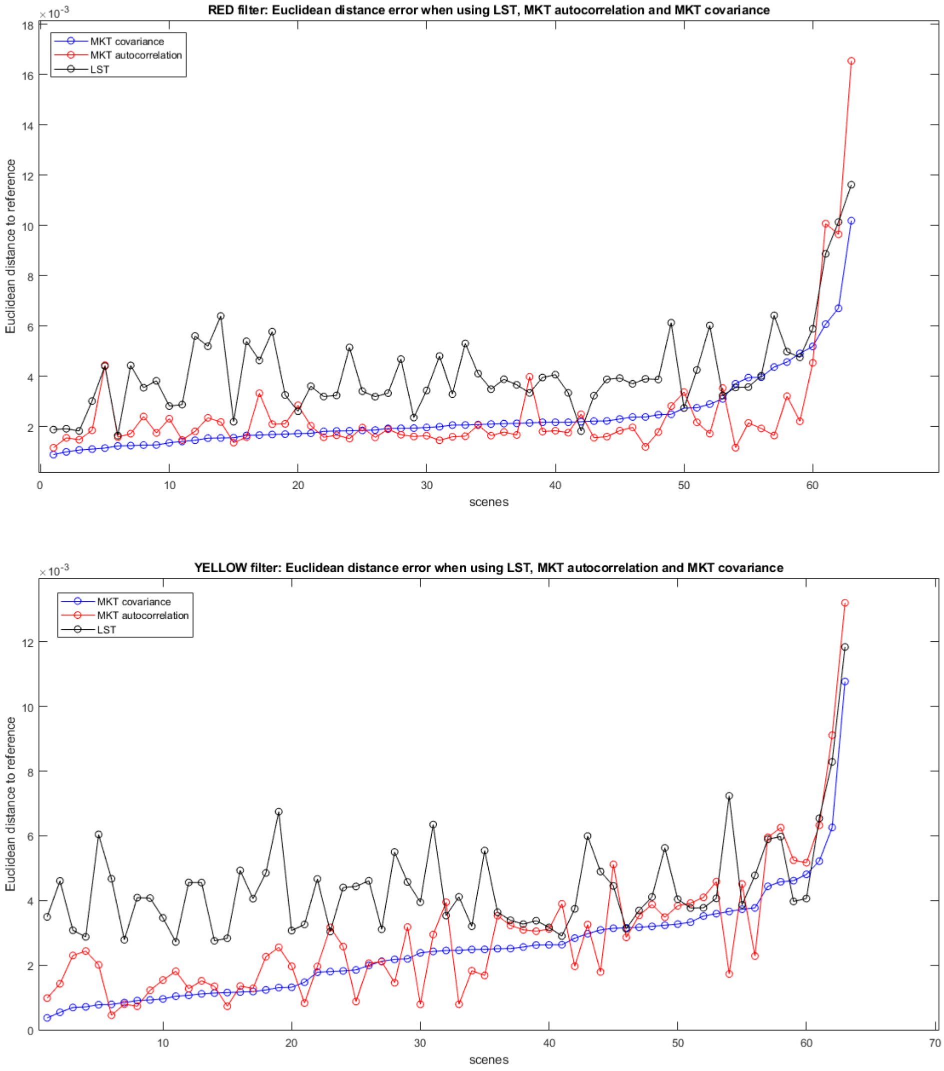


Fig. 5.12 Kampo dataset: Euclidean distance when using MKT autocorrelation versus MKT covariance versus LST. Top: red filter. Bottom: yellow filter.

Table 5.2 Kampo dataset: summary statistics for MKT versus LST approaches with the red filter (upper table) and the yellow filter (lower table) on the Kampo dataset.

Red filter	**RMS**	**Mean**	**Median**	**σ = SD**	**Max**	**Range** = **max - min**	**95% Quantile**
LST	0.0045	0.0042	0.0038	0.0036	0.0116	0.010	0.0073
MKT autocorrelation	0.0035	0.0025	0.0018	0.0274	0.0165	0.0154	0.0063
MKT covariance	0.0029	0.0024	0.0021	0.0228	0.0102	0.0093	0.0055

Yellow filter	**RMS**	**Mean**	**Median**	**σ = SD**	**Max**	**Range** = **max - min**	**95% Quantile**
LST	0.0047	0.0044	0.0041	0.0037	0.0118	0.0091	0.0069
MKT autocorrelation	0.0036	0.0029	0.0023	0.0284	0.0132	0.0128	0.0063
MKT covariance	0.0030	0.0025	0.0025	0.0237	0.0108	0.0104	0.0050

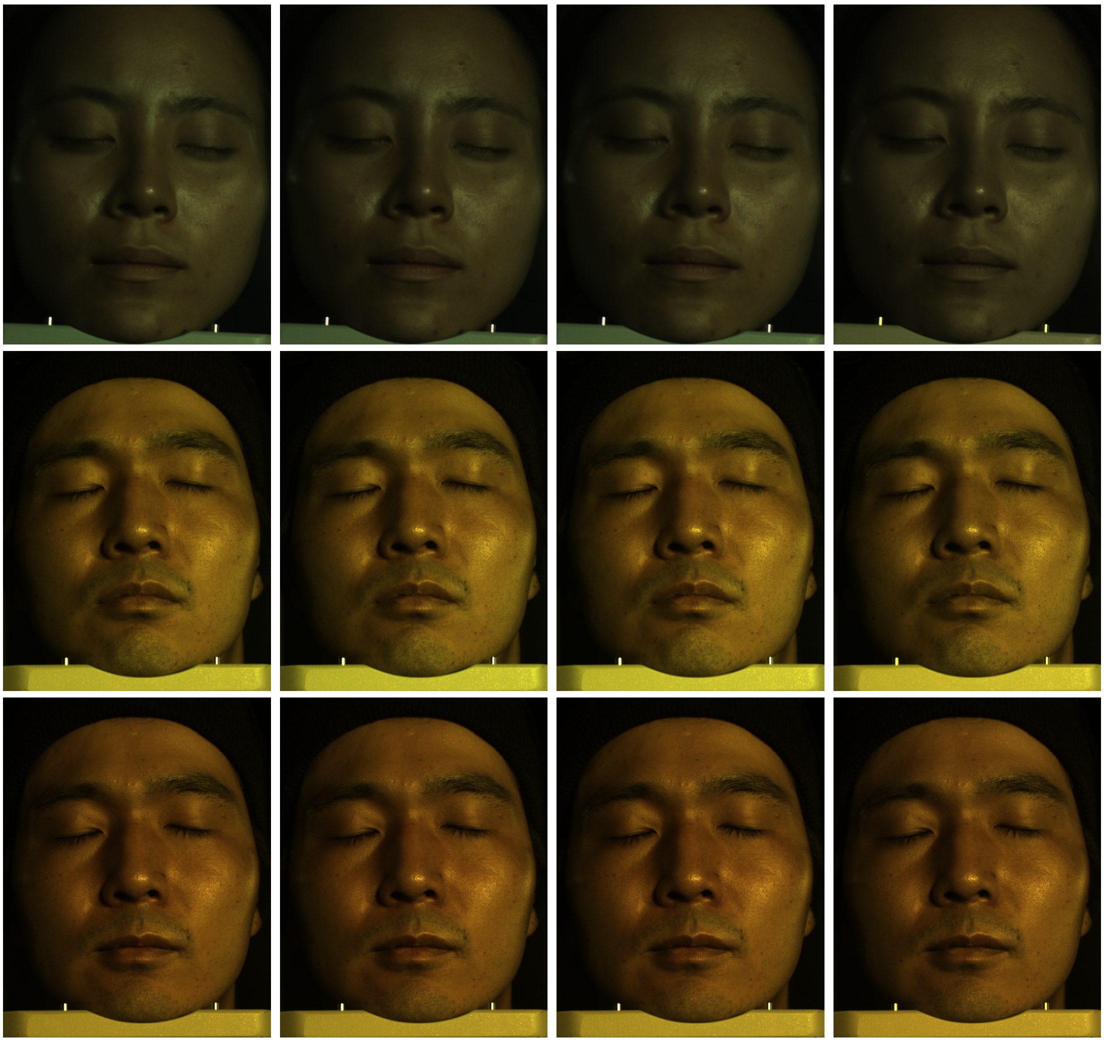


Fig. 5.13 Kampo dataset: colour-corrected images for red filter. From left to right: best solution using T_{cc}, (the ColorChecker-based transform), output using MKT covariance, MKT autocorrelation and LST.

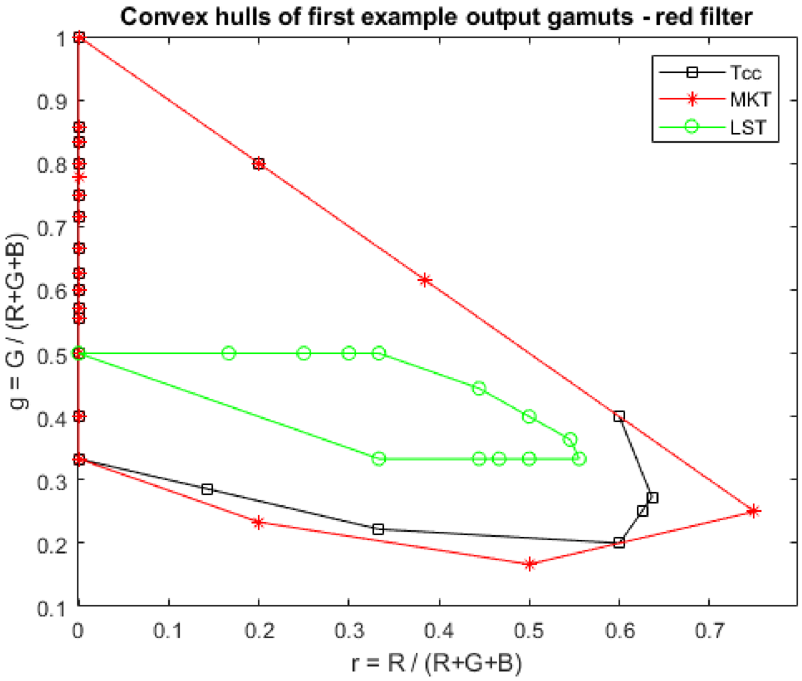


Fig. 5.14 Kampo dataset: convex hulls of first example output gamuts (with red filter) for T_{cc} versus MKT covariance versus LST.

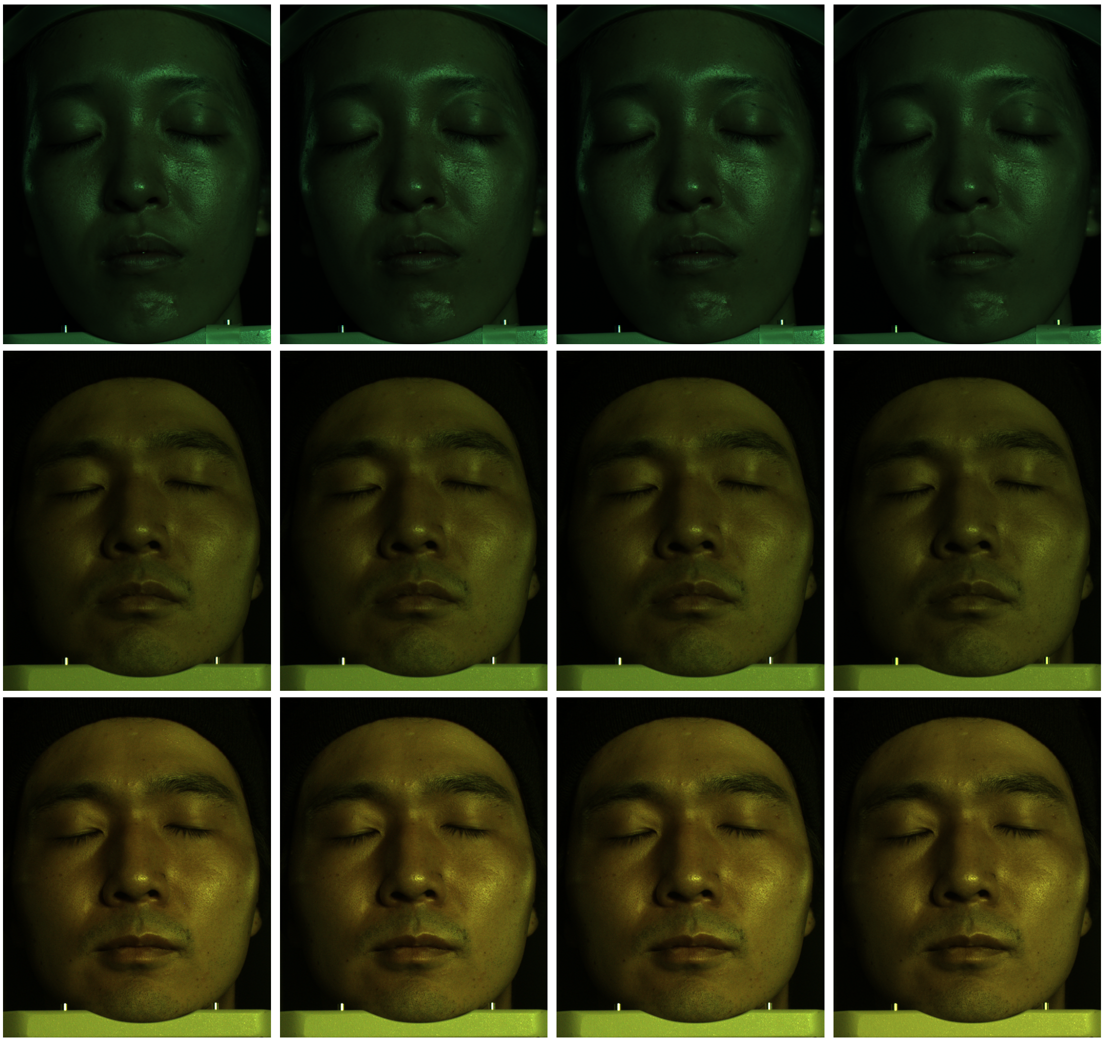


Fig. 5.15 Kampo dataset: colour-corrected images for yellow filter. From left to right: best solution using T_{cc}, (the ColorChecker-based transform), output using MKT covariance, MKT autocorrelation and LST.

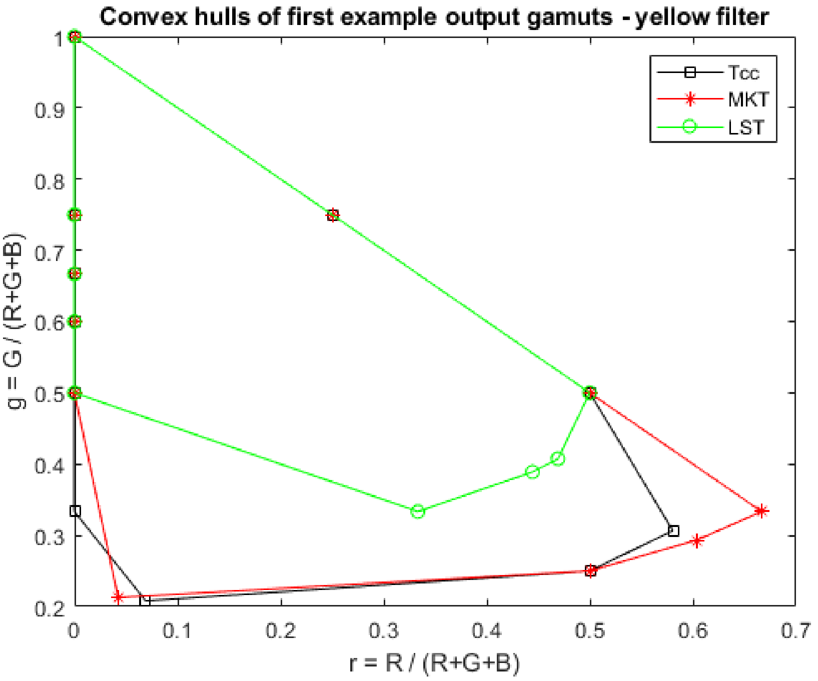


Fig. 5.16 Kampo dataset: convex hulls of first example output gamuts (with yellow filter) for T_{cc} versus MKT covariance versus LST.

5.5 Conclusion

We revisited in the colour correction part of the chromagenic illuminant estimation algorithm. Chromagenic algorithms work by relating the RGBs in a conventional image to a second image taken through a coloured filter. The mapping is generally found by least-squares regression. We show that the original approach has an important constraint : the least-squares transform (LST) is the best solution to the colour correction problem only when the pair of images are registered.

In this chapter, we introduced a new use of the Monge-Kantorovitch transform (MKT) in colour correction. The MKT transform allows the colour correction transform to be calculated without registration. An important contribution is the creation of two new datasets of chromagenic RGB images. Experiments demonstrated that the MKT supports better colour correction than the original approach LST (for unregistered images).

Chapter 6

Monge-Kantorovitch for Colour Stabilisation

We previously used the Monge-Kantorovich transform (MKT) as the preprocessing step for illuminant estimation and more precisely to make the preprocessing of the chromagenic method simpler. In this chapter, we propose using MKT to solve for the colour stabilisation problem.

Colour stabilisation –the change in the colour between frames in a video of the same scene – is a significant problem for video capture, e.g. for TV broadcasting [124] where the challenge is to have a real-time and efficient solution for reprocessing the video so that the content is stable across frames. In the TV industry devices called camera control units are used to keep the colours and other shooting components consistent over time. Colour stabilisation is also important in 3D stereoscopic cinema [125] and amateur video [126]. The change in the colours between two different frames of the scene can have various origins : the use of different cameras or a change in the same camera capture parameters (AWB, ISO, exposure,...) or a different predominant light in the scene depending on the perspective or simply a change in the viewing angle. Broadly, to solve the colour stabilisation problem we need to match content across frames of the same scene and make sure this content has the same colour.

In this chapter, we are interested in solving for colour stabilisation using a set of matching points from the images. The prior art finds the best colour and tonal mapping between the matched points and applies the same transform to one image to match the other. The flaw in this approach is that the matched points only match statistically. There are many false matches. Ideally, we would like to match the colours for the matched points in a manner where registration was not crucial.

We introduce two new methods for colour stabilisation based on the Monge-Kantorovitch transform (MKT) (also reviewed in the last chapter). We showed previously in the last chapter that MKT is less sensitive to a misalignment of the images than the standard least-squares transform. MKT also has a very low computational cost which makes it a good candidate to solve for real-time colour stabilisation. We compare our methods to recently presented algorithms and find the MKT approach delivers competitive performance, while being much simpler in formulation.

6.1 Colour Stabilisation

In colour stabilisation given two images (e.g.from two consecutive frames of the same scene) we wish to change the colour balance of one image to match the other so that the content that is common in both frames looks the same. The colour stabilisation problem has two parts: first we need to find the common content and second we need to convert the colours for this content in one image to match the other.

We propose using the Monge-Kantorovitch transform (MKT) to perform the colour conversion. Because the two images have similar content (frames of the same scene under different viewing angles or shooting parameters), the colour stabilisation problem is a special case of image transfer[127].

We chose to match similar content by finding correspondences points between two images. Matching points are found using the well known SIFT algorithm [91]. The outputs of SIFT are matching points of interest which are associated with image RGBs. If we can map the colours of the matching points - e.g. by applying a linear 3×3 matrix - we can apply this matrix to the whole image thereby solving the colour stabilisation problem if we make the matching content in the source image match the target.

One of the problems of this approach is that the SIFT pairs - while they may be broadly in correspondence - will contain many incorrect matches. Thus in the prior art the linear mapping is found using an expensive procedure such as RANSAC [92][90].

Our idea is to carry out the corresponding matching using the Monge-Kantorovitch transform and in so doing we side-step the need for exact pairwise matches. The MKT returns a linear transform which takes one image to another –or in this case one point set to another– so that their covariance structures match. Because the covariance in one point set is calculated independent of the other we remove the need for correspondence. Actually, the covariance structure of a point set can be represented in more ways and we will consider two variations. First we match covariance matrices. Second, we also consider matching the corresponding autocorrelations, see Section 5.2 for more details.

6.2 Monge-Kantorovitch Image Stabilisation

We propose to use the Monge-Kantorovitch transform to solve for colour stabilisation. Given two image frames of the same scene, a source image and a target image (also referred to as a reference image in [90]). On pair of images is shown in Figure 6.1.

The methods has two main steps: first the common content of the two images needs to be found. For that we use the SIFT algorithm to find correspondence points between the two images. More precisely SIFT gives a set of the correspondence pixels positions [3][4]. For the example in Figure 6.1, we plot the correspondence points in Figure 6.2.

The SIFT set of the pixels positions defines a vector of corresponding points for every image. We plot these points as 3-d point clouds in Figure 6.3.

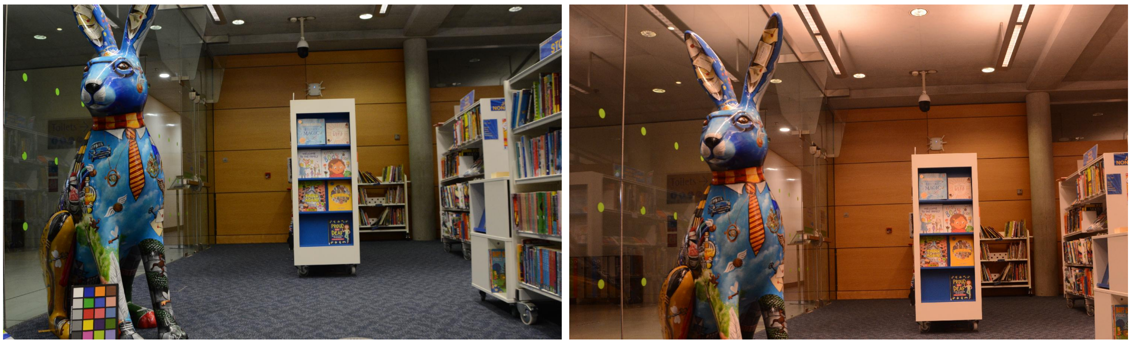


Fig. 6.1 A pair of images of the same scene under different capture conditions: source image (left) and target image (right).

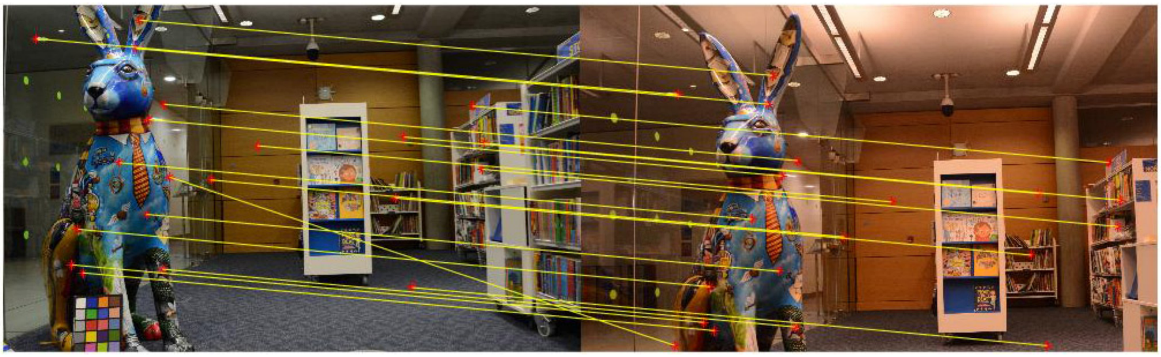


Fig. 6.2 SIFT Corresponding points for the source image (left) and target image(right). Only 20 pairs are plotted.

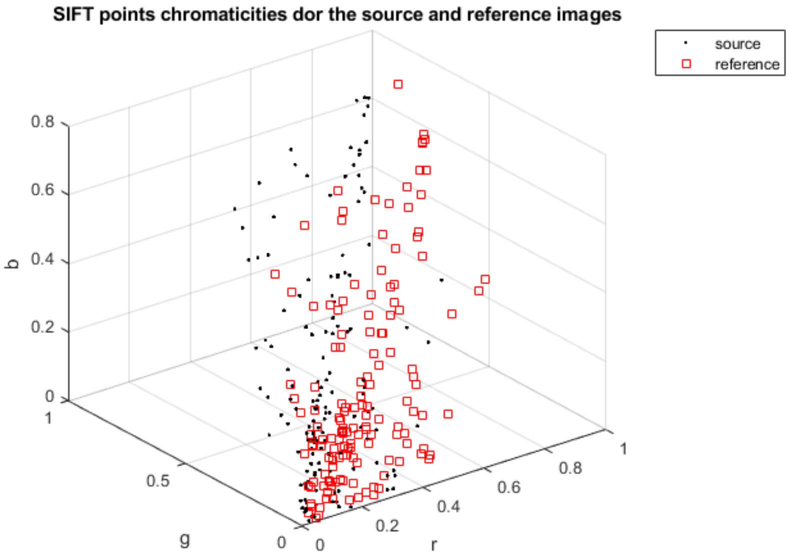


Fig. 6.3 rgb chromaticities gamut of the SIFT correspondence points for the source and target images. Only 100 pairs are plotted.

Now we want to match the points sets one to the other. The sets of points are denoted $\mathbf{F}_{src}$ and $\mathbf{F}_{tar}$. Let us assume exact correspondence of the points and solve the mapping between the two sets by a least-squares regression. The least-squares transform LST, denoted $\mathbf{T}_{LST}$, is defined as follows (Equation 6.1):

$$\mathbf{T}_{LST} = \mathbf{F}_{src}{}^{+}\mathbf{F}_{tar} \tag{6.1}$$

where $\mathbf{F}_{src}$ and $\mathbf{F}_{tar}$ are the $N \times 3$ sets of points from the source and target images, and $^{+}$ denotes the Moore-Penrose inverse.

We apply LST to the source image and we expect visually the result to be as close as possible to the target image, see Figure 6.4. Clearly the Figure shows that the colour stabilisation is poor, and this is because the matching is imperfect. In fact, the pairs of pixels not in correct correspondence are outliers and the least-squares transform is very sensitive to outliers. The calculated transform can quickly deviate from the ideal.

We replace the least-squares transform by the Monge-Kantorovitch transform (MKT) to calculate the mapping between the 2 sets of matching points given by SIFT using the covariances of the 2 sets. We again apply this new transform to the source image and show the result in Figure 6.4. The quality of the colour stabilisation in this case is more convincing.

We also consider in this work another representation of the covariance structure of a point set which is autocorrelation (see Section 5.2), and we present a 2$^{\text{nd}}$ approach for the

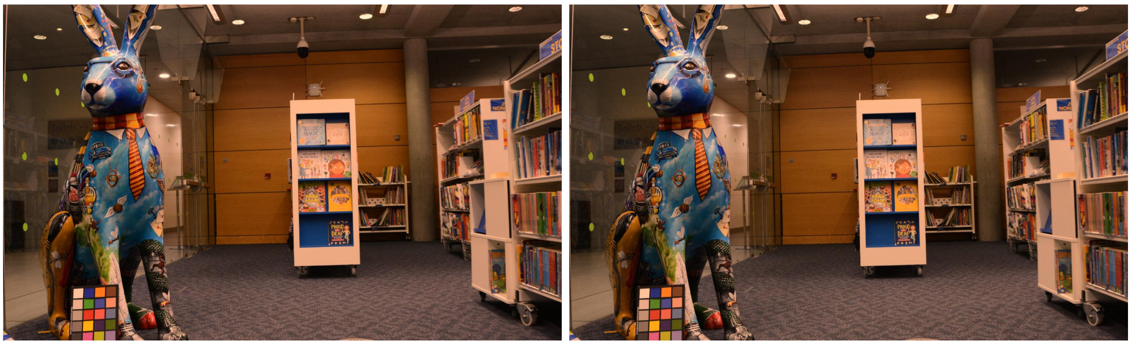


Fig. 6.4 Results of the colour stabilisation for LST (left) versus MKT (right) using the SIFT points.

MKT-based method using the autocorrelations of the point sets. One interesting property of MKT presented also in Section 5.2 is symmetry. The colour stabilisation methods that we consider in what follows are based on symmetric transforms.

6.3 Other Methods for Colour Image Stabilisation

We will compare our proposed method based on the Monge-Kantorovitch transform (MKT) to 3 other recent methods.

The first colour stabilisation method, described in [90], is based on the Scale-Invariant Feature Transform (SIFT). The SIFT algorithm is used to find the sets of matching points between the 2 images, the source and the target, and solve the mapping between the 2 sets by least-squares regression. The transform is then applied to the source image to match the target.

The second method by Vazquez-Corral and Bertalmío [90] is a 2-step approach using both the SIFT [91] and the RANdom SAmple Consensus (RANSAC) [92] to refine the sets of corresponding points and find the best linear transform matching the two sets of points. While this method can work well, the RANSAC process is computationally laborious. The method uses SIFT and RANSAC to find correspondence points in the 2 images, the source and the target, and calculate the least-squares transform with these points. The transform is then applied to the source image to match the target. More details about these 2 methods are presented in Chapter 2.

The last method that we consider is the colour transfer MKL by Pitié and Kokaram. [111] which is also based on the symmetric transform MKT. Colour transfer MKL finds the mapping of all the colours in one image to another (and not considering only the SIFT sets of points which correspond to the content that is similar). The method relates two images with a linear transform (again we are treating the colour stabilisation problem as one of mapping images, and not just the overlapping content). Further, using the colour transfer MKL approach for colour stabilisation between frames of the same scene is also novel.

We made the least-squares transform symmetric (see next section) for a fair comparison with the symmetric MKT-based approaches. That way all compared methods use symmetric transforms.

6.3.1 Symmetric Least-squares Transform Calculation

Given $\mathbf{F}_{src}$ and $\mathbf{F}_{tar}$, $N \times 3$ matrices of matching points between the source image $\mathbf{I}_{src}$ and target images $\mathbf{I}_{tar}$, we first calculate the products of $\mathbf{F}_{src}$ by the set of 3×3 symmetric matrices $\mathbf{U}_{ij}$ having 1 at positions $(i, j) and (j, i)$ and zeros elsewhere.

The $N \times 3$ matrix products are defined as follows:

$$\mathbf{A}_{11} = \begin{pmatrix} f_{r_1} & f_{g_1} & f_{b_1} \\ \vdots & \vdots & \vdots \\ f_{r_N} & f_{g_N} & f_{b_N} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad , \quad \mathbf{A}_{22} = \begin{pmatrix} f_{r_1} & f_{g_1} & f_{b_1} \\ \vdots & \vdots & \vdots \\ f_{r_N} & f_{g_N} & f_{b_N} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$\mathbf{A}_{12} = \begin{pmatrix} f_{r_1} & f_{g_1} & f_{b_1} \\ \vdots & \vdots & \vdots \\ f_{r_N} & f_{g_N} & f_{b_N} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad , \quad \mathbf{A}_{23} = \begin{pmatrix} f_{r_1} & f_{g_1} & f_{b_1} \\ \vdots & \vdots & \vdots \\ f_{r_N} & f_{g_N} & f_{b_N} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$
$$\mathbf{A}_{13} = \begin{pmatrix} f_{r_1} & f_{g_1} & f_{b_1} \\ \vdots & \vdots & \vdots \\ f_{r_N} & f_{g_N} & f_{b_N} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad , \quad \mathbf{A}_{33} = \begin{pmatrix} f_{r_1} & f_{g_1} & f_{b_1} \\ \vdots & \vdots & \vdots \\ f_{r_N} & f_{g_N} & f_{b_N} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.2}$$

Using the matrices $\{\mathbf{A}_{ij}\}$ we create the matrix $\mathbf{M}_{src}$ which is $3N \times 6$ defined such as each of its $3N \times 1$ column vector $\underline{M}_{src_{ij}}$ has the elements of $\mathbf{A}_{ij}$.

Then for every column vector $\underline{M}_{src_{ij}}$ we solve x in the following equation:

$$\underline{v}_{tar}x = \underline{M}_{src_{ij}} \qquad i, j \in [1,6] \tag{6.3}$$

where $\underline{v}_{tar}$ is $3N \times 1$ vector and has the R, G and B elements from $\mathbf{F}_{tar}$. $x_{ij} = \underline{M}_{src_{ij}} \backslash \underline{v}_{tar}$ is a solution of the equation for every column vector $\underline{M}_{src_{ij}}$.

Solving Equation 6.3 corresponds to solving $3N$ equations in $3N$ unknowns. The symmetric least-squares transform can be written as follow:

$$\mathbf{T}_{LST} = \begin{pmatrix} \underline{M}_{src_{11}} \backslash \underline{v}_{tar} & \underline{M}_{src_{12}} \backslash \underline{v}_{ref} & \underline{M}_{src_{13}} \backslash \underline{v}_{tar} \\ \underline{M}_{src_{21}} \backslash \underline{v}_{tar} & \underline{M}_{src_{22}} \backslash \underline{v}_{tar} & \underline{M}_{src_{23}} \backslash \underline{v}_{tar} \\ \underline{M}_{src_{31}} \backslash \underline{v}_{tar} & \underline{M}_{src_{32}} \backslash \underline{v}_{tar} & \underline{M}_{src_{33}} \backslash \underline{v}_{tar} \end{pmatrix} \tag{6.4}$$

6.4 Datasets for Colour Stabilisation

The first and second datasets are series of images of a variety of urban scenes captured in Barcelona (Spain) by Vazquez-Corral et al. [90] [128]. Every scene was captured twice under different angles (two perspectives). The images –the source and the target– are the outputs from the camera pipeline. In order to establish a ground-truth colour stabilised image, every scene was captured with perspective relative to the source image and capture parameters of the target image [90].

Changing the perspective to take the second capture of the scene results in two non-aligned images showing the same objects but with a small change in colour. The more the two perspectives are different, the more this difference in colour is noticeable and therefore the need for colour stabilisation becomes important. The colour change between the paired images may also be due to a change in the illuminant.

In the discussion that follows we will consider 3 images. First we have the image pair from an image sequences and it is this pair for which we want to stabilise the colour. We will talk about mapping the source image to the target (respectively, the first and second image in the pair). Of course we will not be able to map the colours exactly. However, a ground-truth image was captured for every scene with perspective relative to the source image and capture parameters of the target image [90]. We use the ground-truth image as the best possible colour stabilised image to evaluate the methods.

The first dataset has 10 images of 5 different scenes [90]. To emphasize the colour difference between the two images of the same scene for the purpose of performance evaluation, different AWB modes were used (see Figure 6.5). The source image was taken with AWB on, and the target with "illuminant = sunny". The images are in RGB JPEG format (3072×1728 pixels).

The second set of images has 40 scenes [128]. The images were taken using two camera models, Nikon D3100 (12-bits) and Canon EOS80D (14-bits). Manual mode was used in order to have full control over exposure time, white balance, ISO value, and aperture. The source, target and ground-truth images are in a JPEG format (4608×3072 pixels). All images were reduced to size 1152×768. The down-sampling might remove some details

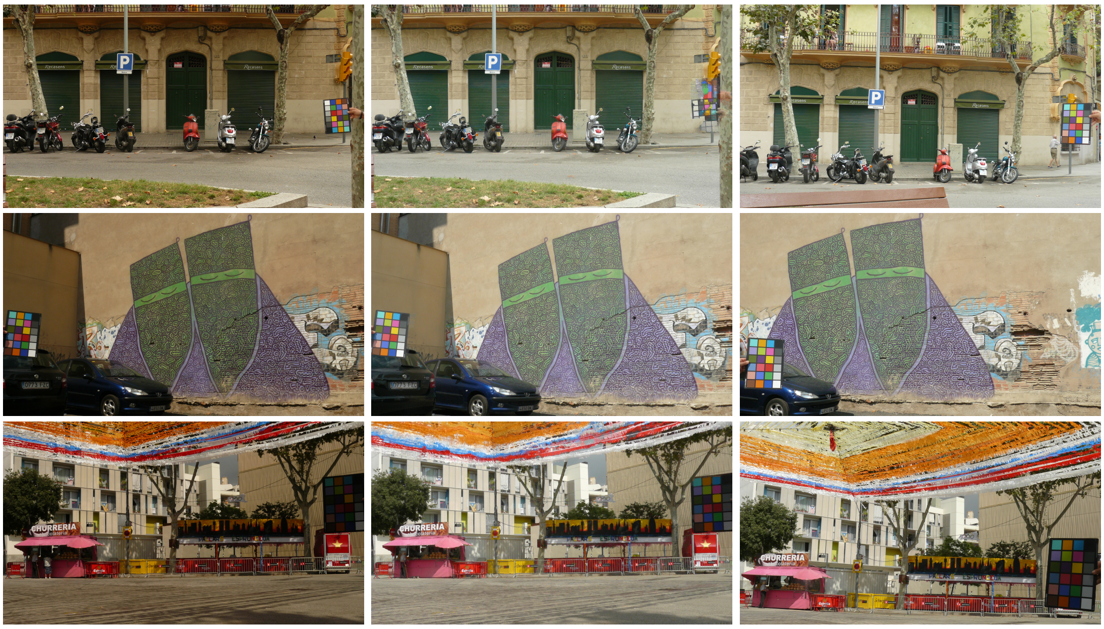


Fig. 6.5 Scenes from the first dataset. From left to right: source image, ground-truth image and target image.

and introduce spatial distortions of the image content but it globally preserves the content relations between the images.

In all our experiments the input images are rendered jpegs. The underlying colour stabilisation are carried out in a linearised space. Linearisation is carried out using the method described in the next section.

In the next section we will run a variety of colour stabilisation algorithms to map source image to the corresponding target image yielding what we call a *colour-stabilised image*. The performance evaluation of the algorithms consists in comparing the ground-truth image to the colour-stabilised image for every scene.

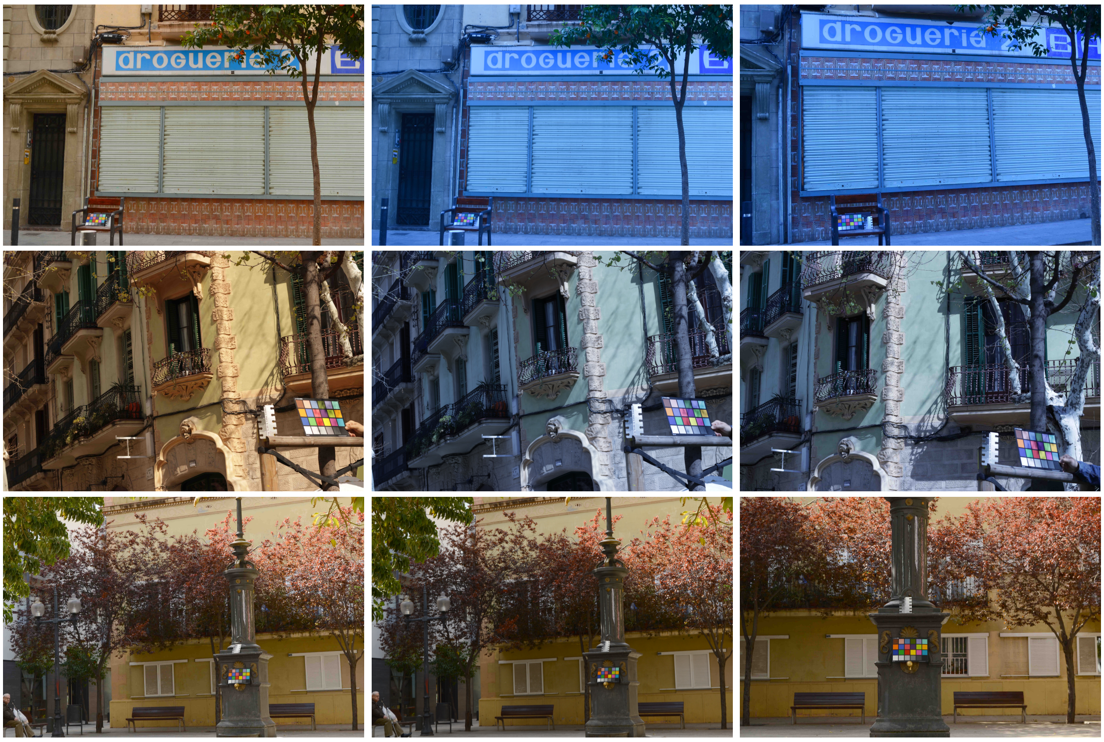


Fig. 6.6 Scenes from the second dataset. From left to right: source image, ground-truth image and target image.

6.5 Experiments and Results

We evaluate our two approaches –covariance and autocorrelation-based MKT colour stabilisation– against recent methods: the SIFT [3] and RANSAC-based colour stabilisation methods [90], the colour transfer MKL (based on the MKT symmetric transform). The SIFT and RANSAC-based methods use a least-squares transform computed with the SIFT and RANSAC output points that we made symmetric for a fair comparison of all the methods. The calculation method of the symmetric LST is described in Section 6.3.1. That way all compared methods use symmetric transforms.

For every scene, we first estimate the gamma-correction factor to linearise the images (see next section) using the SIFT+RANSAC algorithm outputs. Then we perform a reverse gamma-correction to linearise the images (the other camera pipeline processing steps are not reversed) and we calculate for 3 different SIFT threshold values ($th = \{0.005, 0.007, 0.009\}$) the SIFT outputs on the linearised images successively that we use in the different transforms calculation. As described in [90], to avoid the possible inaccuracies of the SIFT matches we first compute smoothed versions of the original images, after convolution with a Gaussian of standard deviation $\sigma = 5$. We use the SIFT code from [3][4].

We calculate the transform with each of the methods. We then apply the transform to the source (first image) image, we re-do the gamma-correction and we compare the colour-stabilised image to the ground-truth image using various metrics. We evaluate the performance of the methods and compare them with summary statistics.

6.5.1 Gamma Estimation

We first estimate the gamma value γ for every scene, using the methodology described in [90]. For each pair of images, source $\mathbf{I}_{src}$ and target $\mathbf{I}_{tar}$, we linearise the data (JPEGs outputs of the camera pipeline) with $\gamma = 2.2$ to run SIFT (for every threshold value $th = \{0.005, 0.007, 0.009\}$). The threshold filters the small features (in absolute value in the DoG scale space). With a small threshold the number of features increases. We obtain the correspondences points with SIFT. We refine these correspondences with RANSAC, which removes the matches that cause a large colour error and we obtain two matches matrices. The parameter values that we use for RANSAC are the same as in [90]): 5 points to fit the model, 1000 iterations, a threshold value of 0.09 for determining when a datum fits a model, and 0.001 as the minimum length of the solution vector.

We estimate the values of the gammas using the least-squares transform and by solving a polynomial regression:. For different values $g1$ and $g2$ ranging from 2 to 3, we use two matches matrices $\mathbf{F}_{src}$ and $\mathbf{F}_{tar}$ (JPEG pixels values), we first calculate the least-squares transform $\mathbf{T}_{LST}$ such as $\mathbf{F}_{tar}^{g2} = \mathbf{F}_{src}^{g1}\mathbf{T}_{LST}$. Then we solve the polynomial regression between $\mathbf{F}_{tar}^{g2}$ and $\mathbf{F}_{src}^{g1}T$. The best pair of gammas $g1$ and $g2$ is the one that gives the smallest value for the polynomial P coefficient with the highest degree i.e. error $min|P(1)-1|$ where $||$ is the absolute value. Finally by ratios equalisation, one gamma parameter (the largest) is set to 2.2 and the other to the estimated normalised value, by solving $2.2/g = g1/g2$ when $g1 > g2$.

Considering that SIFT and RANSAC were used to estimate the gamma values, we can expect the results to be slightly in favour of theses two methods. In fact we observed that estimating the gammas with the MKT approaches gives better results when evaluating these methods. In what follows, we use one estimation of the gammas $(g1, g2)$ based on the least-squares regression.

6.5.2 Colour Error Metrics

We rely for the performance evaluation and the comparison on different error metrics. For every scene, given the colour-stabilised and the ground-truth images we calculate the Euclidean distance image-difference, the colour-difference in the CIE76 space, the CIE 2000 colour-difference ΔE_{00} [86][87] (see 2.4.2 for more details about these state of the art metrics) and the improved Colour-Image-Difference (iCID) [129][130].

The Colour-Image Difference (CID) metric [130] was initially developed for optimising gamut-mapping and predicting its distortions. It contains local comparison terms for computing lightness-difference, lightness-contrast, lightness-structure, chroma-difference and hue-difference. Its framework comprises image normalization by an image-appearance model, features (Image-Difference Features IDFs) extraction, and features combination. The metric that we use is the improved version iCID [129] which was designed to avoid the artefacts of the CID-based gamut-mapping optimizations due to the gamut distortions (and avoid predictions inconsistent with the perception). The common artefacts are the lightness inversion and banding and the chromatic ringing and edges. The iCID metric (see Equation 6.5) has a total of 10 parameters: the weighting parameters $c_1, ..., c_7$, the window size k and the standard deviation for computing the Gaussian weights.The image comparison is operated in the perceptually uniform LAB2000HL colour space (HL stands for hue linear) [131].

$$iCID_P(\mathbf{I}_{src},\mathbf{I}_{tar}) = 1 - \frac{1}{|P|}\sum_{i\in P}[l_L(x_i,y_i).c_L(x_i,y_i).s_L(x_i,y_i)^{\alpha}.l_C(x_i,y_i).l_H(x_i,y_i).c_C(x_i,y_i).s_C(x_i,y_i)] \tag{6.5}$$

where P is an index set with the pixels positions and $|P|$ is the cardinal number of P. The pixels (x_i,y_i) in the window (sub-region) are extracted from the LAB2000HL images $\mathbf{I}_{src}$ and $\mathbf{I}_{tar}$. l_L is the lightness-difference comparison term, c_L is the lightness-contrast, s_L is the lightness-structure, l_C chroma-difference, l_H is the hue-difference, c_C is the chroma-contrast and s_C is the chroma-structure.

6.5.3 Results

Results on 10 scenes Barcelona Series Dataset

In this section we present the quantitative and qualitative results of the colour stabilisation methods on the two datasets of images. We evaluate the performance of the methods with different error metrics.

For the first dataset of 10 scenes, we present the mean and median error results in Table 6.1. our MKT-based approaches give as good results as RANSAC, in particular the MKT autocorrelation method. The ΔE^*_{00} and iCID mean errors for MKT autocorrelation and RANSAC are very close and the median errors are the same. MKT autocorrelation seems to be a good alternative to the computationally expensive method RANSAC.

Considering the output images of the methods –we present four examples in Figure 6.7– we can say that our approaches deliver overall convincing visual results, where the colour transfer MKL, SIFT and sometimes RANSAC fail. On the first example, RANSAC and the autocorrelation-based MKT give similar visual results. On the second example, correlation-based MKT succeeds to restore the saturated red of the motorbike while the result with RANSAC is less convincing. The covariance-based MKT image shows a stronger yellow hue on the facades of the buildings which is different from the colour in the ground-truth image. One the third example, the orange of the truck is well restored using RANSAC.

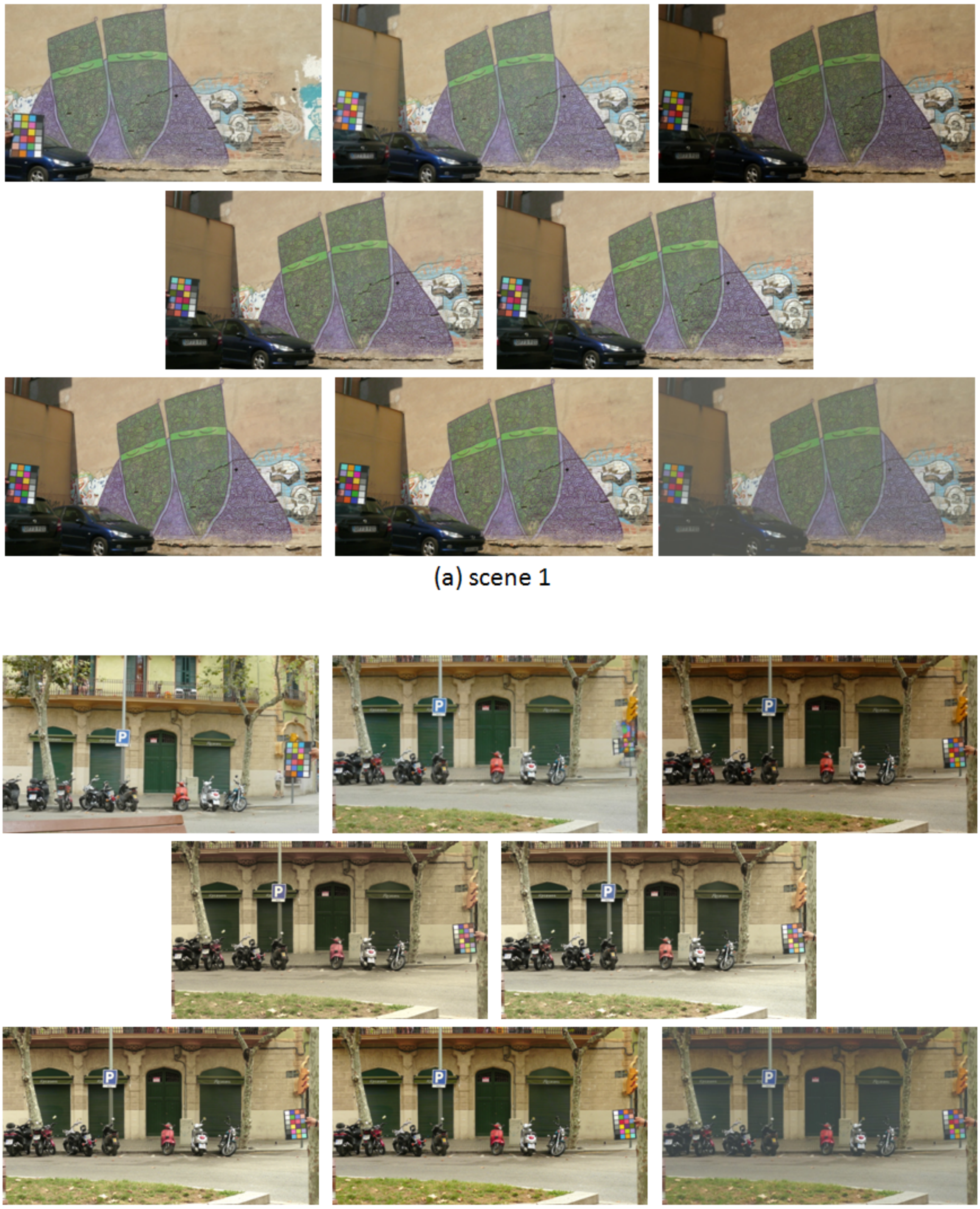
Table 6.1 Quantitative performance evaluation of the colour stabilisation methods on the first dataset: error measures with four different metrics. Top table: mean errors. Bottom table: median errors. In brackets, the best threshold parameter values.

Mean error	**SIFT** (0.009)	**RANSAC** (0.009)	**MKT autocorrelation** (0.009)	**MKT covariance** (0.007)	**Colour transfer MKL**
Euclidian distance	0.107	0.095	0.100	0.106	0.104
CIE76 error	15.955	14.487	14.280	15.044	15.657
ΔE^*_{00}	12.192	11.066	10.995	11.553	12.267
iCID	0.554	0.538	0.536	0.548	0.588

Median error	**SIFT** (0.009)	**RANSAC** (0.009)	**MKT autocorrelation** (0.009)	**MKT covariance** (0.007)	**Colour transfer MKL**
Euclidian distance	0.103	0.099	0.099	0.102	0.0971
CIE76 error	14.807	13.580	13.661	14.540	13.511
ΔE^*_{00}	11.967	10.497	10.708	12.225	11.508
iCID	0.531	0.531	0.531	0.532	0.578

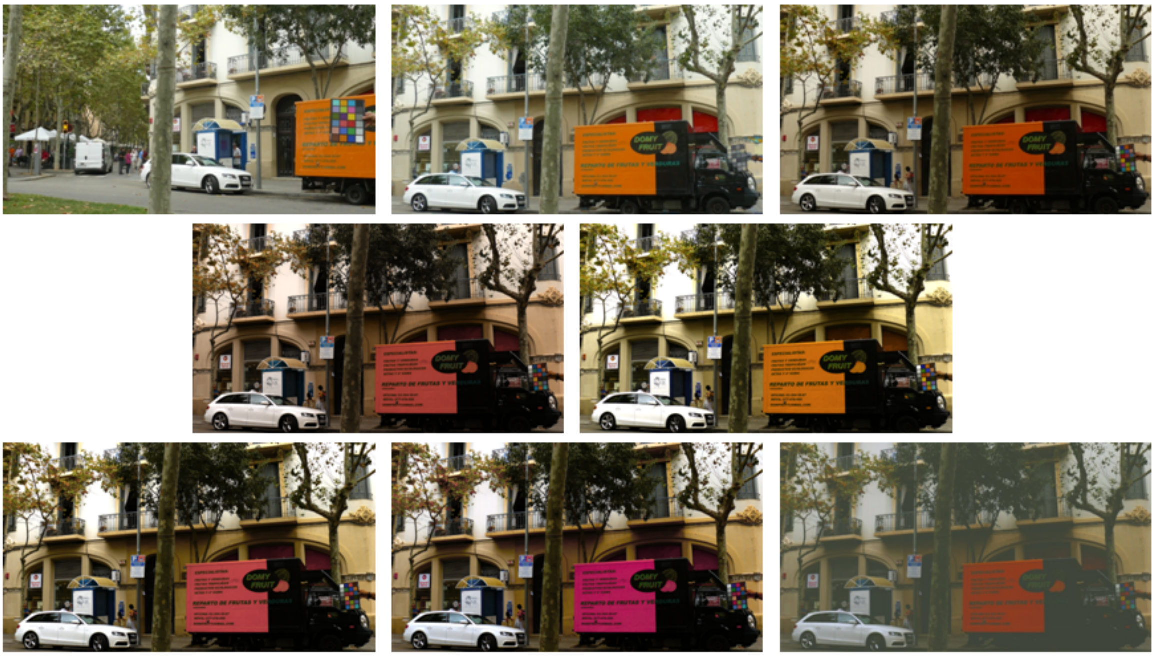
However both RANSAC and our two approaches fail regarding the walls colour. On the last example, our approaches work better than the other methods.

Fig. 6.7 Qualitative performance evaluation of the colour stabilisation methods on the first dataset: output images of four scenes (a)(b)(c) and (d). For every scene, from left to right: (Top row) Target, ground-truth and source images, (middle row): SIFT, RANSAC(bottom row) MKT autocorrelation, MKT covariance and colour transfer MKL. SIFT parameter is th = 0.005

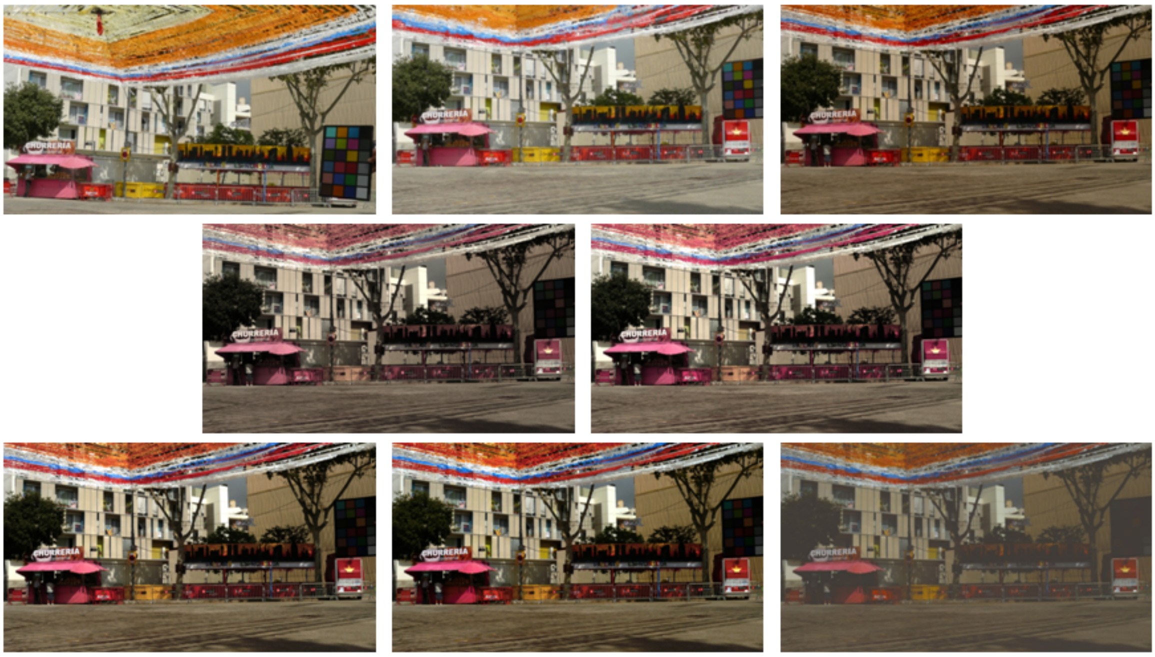


(a) scene 1

(b) scene 2



(c) scene 3



(d) scene 4

Results on 40 scenes Barcelona Series Dataset

For the second dataset of 40 scenes, we present the mean and median error results in Table 6.2. We calculate the error in the centre area of the images for every scene. The mean and median errors are in the same range as for the results on the first dataset. The Euclidean distance, ΔE^*_{00} and iCID errors are smaller with this dataset for all methods. The SIFT and RANSAC method handles very well this dataset and gives much better results in terms of mean and median errors than on the first dataset. Our MKT-based approaches do not give very good quantitative results on this dataset, due partially to the loss of information when resizing the images.

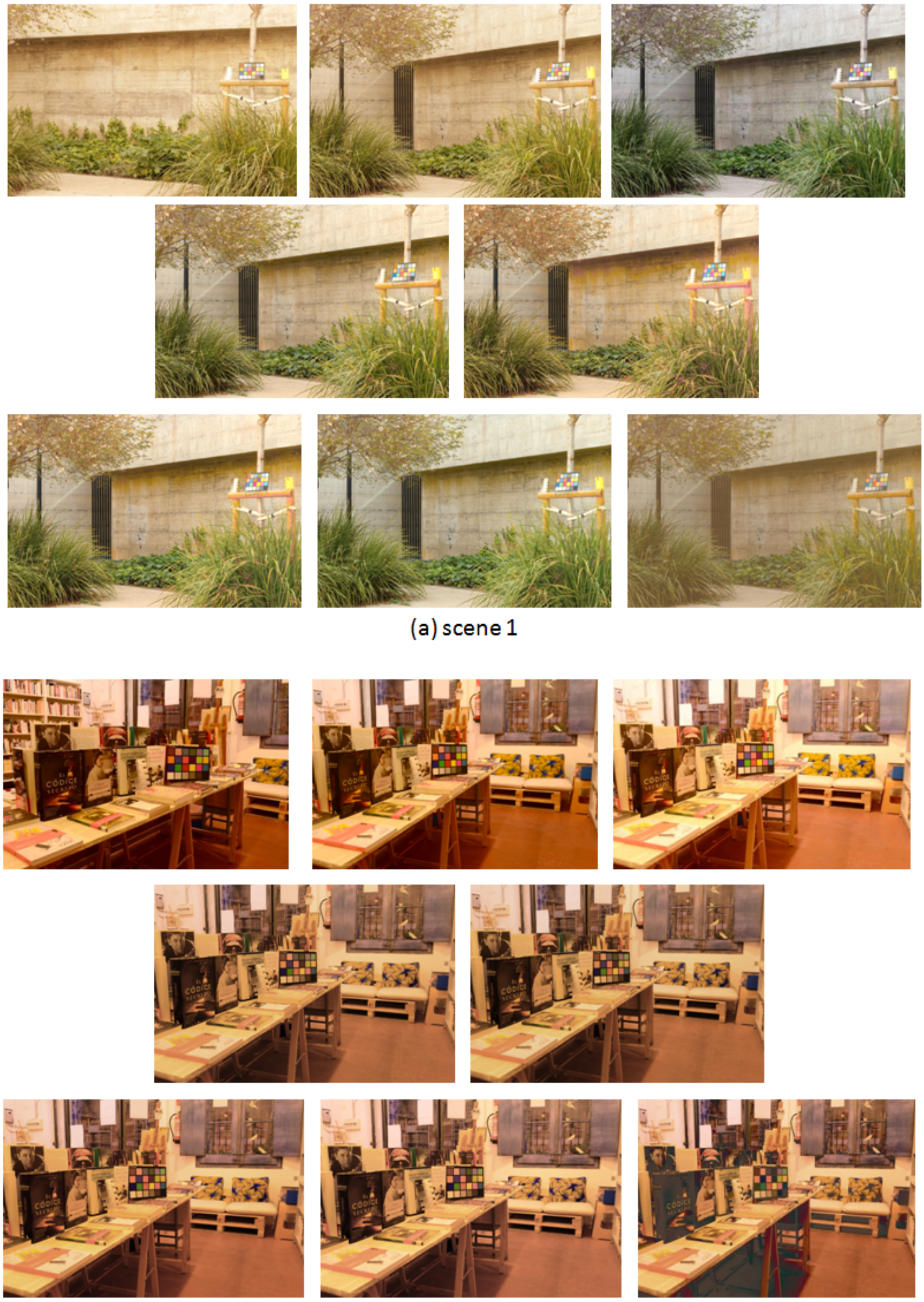
In what follows we present the output images of the six methods including our two MKT-based approaches. The results are overall good with our approaches that we recall are cost effective compared to the RANSAC algorithm. On the first example, RANSAC and autocorrelation-based MKT present artefacts on the wall and plants in the images. However, the colours in the rest of the image are well recovered. On the second example, SIFT, RANSAC and our methods do not give this artefact. However, the dark edges of the window panes are less visible with these methods. On the last example, our approaches give very convincing visual results.

Table 6.2 Quantitative performance evaluation of the colour stabilisation methods on the second dataset: error measures with four different metrics. Top table: mean errors. Bottom table: median errors. In brackets, the best threshold parameter values.

Mean error	**SIFT** (0.007)	**RANSAC** (0.007)	**MKT autocorrelation** (0.005)	**MKT covariance** (0.007)	**Colour transfer MKL**
Euclidian distance	0.084	0.079	0.085	0.100	0.110
CIE76 error	14.290	13.534	15.634	18.894	17.115
$\boldsymbol{\Delta E^{*}_{00}}$	9.749	9.218	10.686	12.740	12.273
iCID	0.405	0.393	0.429	0.476	0.482

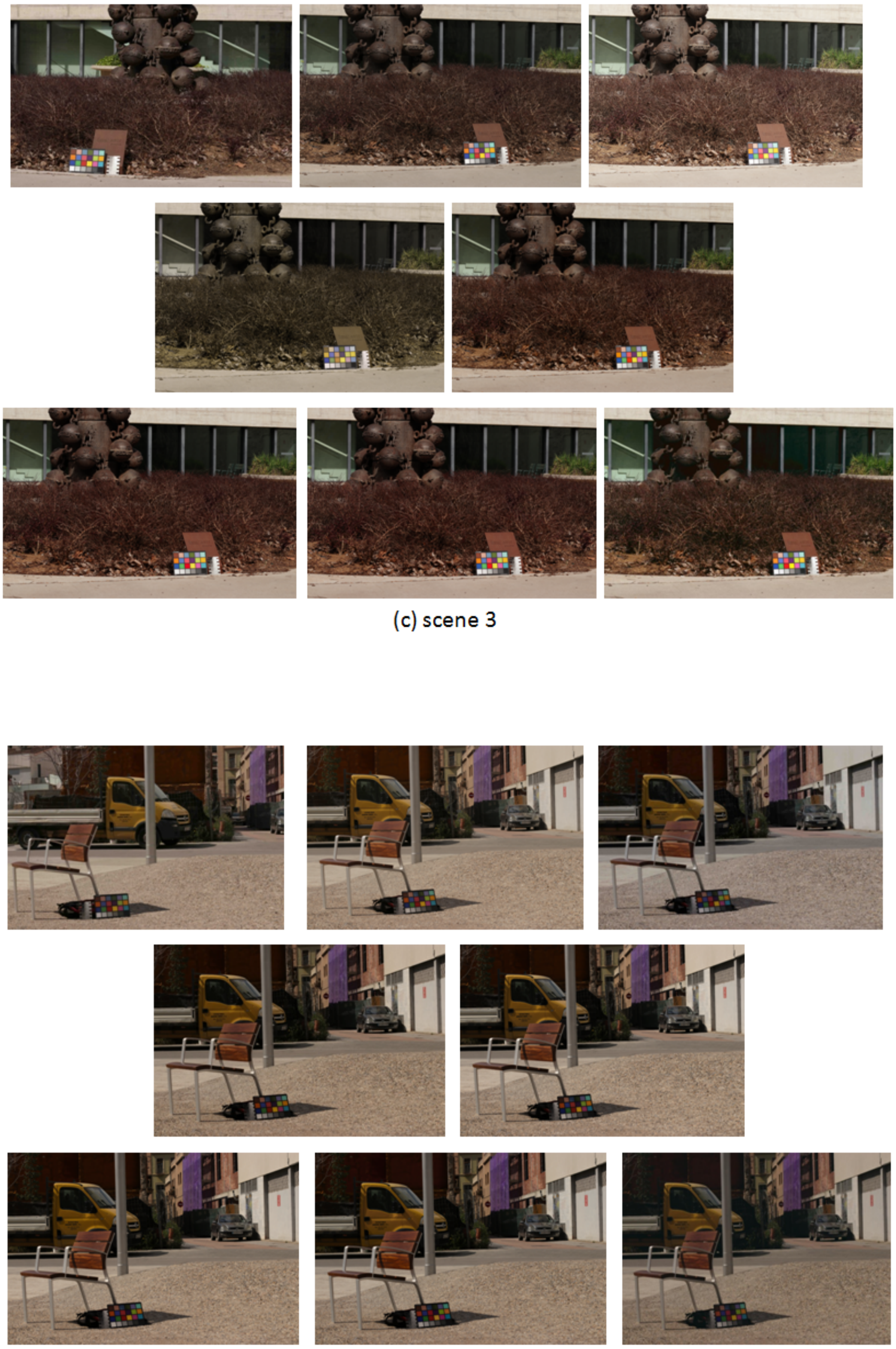
Median error	**SIFT** (0.007)	**RANSAC** (0.007)	**MKT autocorrelation** (0.005)	**MKT covariance** (0.007)	**Colour transfer MKL**
Euclidian distance	0.067	0.061	0.072	0.091	0.098
CIE76 error	11.991	11.909	14.124	17.267	15.297
$\boldsymbol{\Delta E^{*}_{00}}$	8.277	7.439	9.234	11.225	11.452
iCID	0.353	0.359	0.398	0.437	0.483

Fig. 6.8 Qualitative performance evaluation of the colour stabilisation methods on the second dataset: output images of four scenes (a)(b)(c) and (d). For every scene, from left to right: (Top row) Target, ground-truth and source images, (middle row): SIFT, RANSAC(bottom row) MKT autocorrelation, MKT covariance and colour transfer MKL. SIFT parameter is th = 0.005



(a) scene 1

(b) scene 2



(c) scene 3

(d) scene 4

6.6 Conclusion

We introduced in this chapter a new use of the Monge-Kantorovitch transform (MKT) in colour stabilisation. We consider the problem using a set of matching points: we need to transfer the colour gamut from a target image to a source image. The algorithms we introduced comes in two steps: first it computes the SIFT matching points between two images of the same scene, and second it computes the MKT transform using the autocorrelation or covariances matrices of the matching points. Finally the colour stabilised result image is obtained by applying the transform to the source image.

We compare the algorithms, the autocorrelation and covariance-based MKT, to other recent methods of colour transfer: SIFT and RANSAC [90], colour transfer MKL [111]. The performance evaluation on two datasets of images showed that our method performs well and gives comparable quantitative and visual results to the computationally expensive RANSAC.

Chapter 7

Conclusion and Future Work

In the presented work we considered two important colour processing operations: colour constancy and colour stabilisation. First we examined the ColorChecker dataset for illuminant estimation. We discovered a problem with the dataset ground-truth –correct answers of the illuminants colours–. We showed that not only different sets of ground-truth exist but also that their calculation methodologies are not correct. We investigate the sources of errors in the calculations and the impact on the performance evaluation and rankings of the algorithms. Finally based on the methodology described by Shi and Funt we re-processed the dataset images and re-calculated the ground-truth that we named the RECommended set and we present in this work as well as on the colorconstancy.com web-site extensive results of the algorithms evaluation in terms of angular errors.

This project let us emphasize the importance of the accuracy of the ground-truth and the repeatability of the evaluation procedure when evaluating the performance of algorithms and comparing them.

As a future work, we would like to examine the learning methods for illuminant estimation. The ground-truth is used for learning the model and therefore the performance of a learning algorithm is truly affected by the accuracy of the ground-truth. We would like to re-evaluate these methods on the ColorChecker dataset using the REC ground-truth and give a final answer for the evaluation and rankings.

In the next contribution of this thesis, we focused on a specific method of illuminant estimation for colour constancy which is the chromagenic method. In particular, we re-visited the pre-processing step of the method which is chromagenic colour correction. In the chromagenic method, the scene is captured twice with a without a coloured filter. A transform that describes the relationship between the two images is calculated. A calibration step is needed : for a set of filtered and unfiltered images, a set of transforms is calculated. Each transform is relative to a pair of images and goes in pair with an RGB colour vector which is the ground-truth illuminant in the captured scene. For a new pair of captured images, the best transform which gives the best mapping between the two images is selected and gives the estimate of the illuminant. The original chromagenic method uses the least-squares regression to calculate the initial set of transforms and solve for the best transform. However, the least-squares transform works only when the images are registered, and registering images is a costly processing task, We proposed to replace it by the Monge-Kantorovitch transform (MKT) which works without the need for image registration. We test our approach on two sets of chromagenic images that we collected: the Kampo dataset and the UEA dataset. Candidly, we did not actually use our method to try and solve for the illuminant. Rather, we became very interested in the general potential of the MKT. The results in chapter 5 showed we could register colours without registering the images.

Of course, as a follow-up for this project we would like to use MKT for chromagenic illuminant estimation and perform colour constancy. Although we expect our approach to work well without the need to register the images, we believe additional tuning is needed to develop a performing chromagenic algorithm to estimate the illuminant which can work well in the case of real images and compete with state of the art and recent methods.

Another idea we would like to demonstrate, is about improving the Kampo diagnosis system by performing MKT chromagenic colour constancy on the system input face images. The advantage of the image-based system is to emit a medical diagnosis by taking a face image (similar to a selfie) in any uncontrolled lighting conditions (the scene can have shadow, indoor artificial lighting, multiple light sources). The evaluation of the method is possible by comparing the diagnosis emitted by the image-based system to a traditional diagnosis made by a Kampo doctor.

In the last contribution of this thesis we use MKT to solve for colour stabilisation –a further processing of the camera pipeline output image– which is particularly useful in live

TV broadcasting and amateur video. We evaluated our approach on two datasets of images and we compared it to other competitive methods like MKL for colour transfer and the computationally expensive RANSAC. The MK-based method that we propose performs well and by optimising the code, it is possible to have in the future a better performing algorithm.

Bibliography

[1] K. Barnard, L. Martin, B. V. Funt, and A. Coath, “A data set for color research,” *Color Research and Application*, vol. 27, no. 3, pp. 147–151, 2002.

[2] S. W. Oh and S. J. Kim, “Approaching the computational color constancy as a classification problem through deep learning,” *Pattern Recognition*, vol. 61, pp. 405–416, 2017.

[3] A. Vedaldi and B. Fulkerson, “VLFeat - An open and portable library of computer vision algorithms,” in *ACM International Conference on Multimedia*, no. 1, pp. 1–4.

[4] ——, “VLFeat: An open and portable library of computer vision algorithms,” http://www.vlfeat.org/overview/sift.html, accessed: January-2019.

[5] G. D. Finlayson, G. Hemrit, A. Gijsenij, and P. V. Gehler, “A Curious problem with using the colour checker dataset for illuminant estimation,” in *Color and Imaging Conference*, 2017, pp. 64–69.

[6] G. Hemrit, G. D. Finlayson, A. Gijsenij, P. V. Gehler, S. Bianco, B. V. Funt, M. S. Drew, and L. Shi, “Rehabilitating the ColorChecker dataset for illuminant estimation,” in *Color and Imaging Conference*, 2018, pp. 350–353.

[7] G. Hemrit, G. D. Finlayson, A. Gijsenij, P. V. Gehler, S. Bianco, M. S. Drew, B. V. Funt, and L. Shi, “Providing a Single Ground-truth for Illuminant Estimation for the ColorChecker Dataset,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1286–1287, 2019.

[8] “Tiffen filters,” www.tiffen.com, accessed: 2018-07.

[9] G. Hemrit, F. Matsushita, M. Uchida, J. Vazquez-Corral, H. Gong, N. Tsumura, and G. D. Finlayson, “Using the Monge-Kantorovitch Transform in Chromagenic Color Constancy for Pathophysiology,” in *Computational Color Imaging Workshop*, 2019, pp. 121–133.

[10] G. D. Finlayson, R. Zakizadeh, and A. Gijsenij, “The reproduction angular error for evaluating the performance of illuminant estimation algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1482 – 1488, 2016.

[11] C. Fredembach and G. D. Finlayson, “The bright-chromagenic algorithm for illuminant estimation,” *Journal of Imaging Science and Technology*, vol. 52, no. 4, pp. 137–142, 2008.

[12] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, 1781.

[13] S. Tominaga and B. A. Wandell, “Standard surface-reflectance model and illuminant estimation,” *Journal of the Optical Society of America A*, vol. 6, no. 4, pp. 576–584, 1989.

[14] S. K. Nayar, K. Ikeuchi, and T. Kanade, “Surface reflection: physical and geometrical perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 611–634, 1991.

[15] A. Hurlbert, “Colour vision: Is colour constancy real?” *Current Biology*, vol. 9, no. 15, pp. R558–R561, 1999.

[16] E. H. Land, “The Retinex Theory of Color Vision,” *Scientific American*, vol. 237, no. 6, pp. 108–128, 1977.

[17] D. Slater and G. Healey, “The illumination-invariant recognition of 3D objects using local color invariants,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 206–210, 1996.

[18] Yilmaz Alper, Javed Omar, and Shah Mubarak, “Object tracking: A Survey,” *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, 2006.

[19] L. J. Li, R. Socher, and L. Fei-Fei, “Towards total scene understanding: Classification, annotation and segmentation in an automatic framework,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[20] B. V. Funt and G. D. Finlayson, “Color constant color indexing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 522–529, 1995.

[21] T. Gevers and A. W. M. Smeulders, “Color-based object recognition,” *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[22] J. Von Kries, “Beitrag zur physiologie der gesichtsempfindung,” *Arch. Anat. Physiol*, vol. 2, pp. 505–524, 1878.

[23] G. D. Finlayson, M. S. Drew, and B. V. Funt, “Diagonal Transforms Suffice for Color Constancy,” in *Proceedings of the International Conference on Computer Vision*, 1993, pp. 164–171.

[24] ——, “Color constancy: generalized diagonal transforms suffice,” *Journal of the Optical Society of America A*, vol. 11, no. 11, pp. 3011–3019, 1994.

[25] J. Roca-Vila, C. Alejandro Parraga, and M. Vanrell, “Chromatic settings and the structural color constancy index,” *Journal of Vision*, vol. 13, no. 4, 2013.

[26] S. E. Lynch, M. S. Drew, and G. D. Finlayson, “Colour constancy from both sides of the shadow edge,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 899–906.

[27] G. D. Finlayson, M. S. Drew, and B. V. Funt, “Spectral sharpening : sensor transformations for improved color constancy f,” *Journal of the Optical Society of America A*, vol. 11, no. 5, pp. 1553–1563, 1994.

[28] E. H. Land and J. J. McCann, “Lightness and retinex theory,” *Journal of the Optical Society of America*, vol. 61, no. 1, pp. 1–11, 1971.

[29] G. Buchsbaum, “A Spatial processor model for object colour perception,” *Journal of the Franklin Institute*, vol. 310, pp. 1–26, 1980.

[30] G. D. Finlayson and E. Trezzi, “Shades of gray and colour constancy,” in *Color and Imaging Conference*, no. 1, 2004, pp. 37–41.

[31] J. Vazquez-Corral, M. Vanrell, R. Baldrich, and F. Tous, “Color constancy by category correlation,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1997 – 2007, 2012.

[32] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, “Bayesian color constancy revisited,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[33] A. Gijsenij and T. Gevers, “Color Constancy Using Natural Image Statistics,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[34] S. Bianco, C. Cusano, and R. Schettini, “Color constancy using CNNs,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 81–89.

[35] J. T. Barron, “Convolutional color constancy,” in *IEEE Proceedings of the International Conference on Computer Vision*, 2016.

[36] J. T. Barron and Y.-T. Tsai, “Fast fourier color constancy,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[37] W. Shi, C. C. Loy, and X. Tang, “Deep Specialized Network for Illuminant Estimation,” in *European Conference on Computer Vision*, 2016, pp. 371–387.

[38] Y. Hu, B. Wang, and S. Lin, “FC4: Fully Convolutional Color Constancy with Confidence-weighted Pooling,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2017, pp. 4085–4094.

[39] G. D. Finlayson, P. M. Hubel, and S. D. Hordley, “Colour by Correlation,” in *Color and Imaging Conference*, 1997, pp. 6–11.

[40] G. D. Finlayson, S. D. Hordley, and I. Tastl, “Gamut constrained illuminant estimation,” *International Journal of Computer Vision*, vol. 67, no. 1, pp. 93–109, 2006.

[41] R. T. Tan, K. Ikeuchi, and K. Nishino, “Color constancy through inverse-intensity chromaticity space,” *Journal of the Optical Society of America A*, vol. 21, no. 3, pp. 321–334, 2004.

[42] H.-C. Lee, “Method for computing the scene-illuminant chromaticity from specular highlights,” *Journal of the Optical Society of America A*, vol. 3, no. 10, pp. 1694–1699, 1986.

[43] B. V. Funt, M. S. Drew, and J. Ho, “Color Constancy from Mutual Reflection,” *International Journal of Computer Vision*, vol. 24, pp. 5–24, 1991.

[44] Y. Qian, J.-K. Kamarainen, J. Nikkanen, and J. Matas, "On Finding Gray Pixels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8062–8070.

[45] G. D. Finlayson, S. D. Hordley, and P. Morovic, "Chromagenic Colour Constancy," in *10th Congress of the International Colour Association*, 2005, pp. 547–550.

[46] G. D. Finlayson, P. Morovic, and S. D. Hordley, "The chromagenic colour camera and illuminant estimation," in *Color and Imaging Conference*, 2005, pp. 20–24.

[47] G. D. Finlayson, S. D. Hordley, and P. Morovic, "Colour constancy using the chromagenic constraint," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 1079–1086.

[48] G. D. Finlayson, C. Fredembach, and M. S. Drew, "Detecting illumination in images," in *IEEE 11th International Conference on Computer Vision*, 2007.

[49] D. Cheng, A. Kamel, B. Price, S. Cohen, and M. S. Brown, "Two illuminant estimation and user correction preference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, 2016, pp. 469–477.

[50] E. Hsu, M. T., P. S., S. Avidan, and F. Durand, "Light Mixture Estimation for Spatially Varying White Balance," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 70, 2008.

[51] K. Barnard, "Improvements to Gamut Mapping Colour Constancy Algorithms," in *European Conference on Computer Vision*, vol. 1842, 2000, pp. 390–403.

[52] C. Rosenberg, T. Minka, and A. Ladsariya, "Bayesian color constancy with non-gaussian models," *Advances in Neural Information Processing Systems, MA: MIT Press*, vol. 16, 2003.

[53] R. Tan, K. Nishino, and K. Ikeuchi, "Illumination chromaticity estimation using inverse-intensity chromaticity space," in *IEEE conference on Computer Vision and Pattern Recognition*, 2003, pp. 673–680.

[54] J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, 2007.

[55] A. Gijsenij and T. Gevers, "Datasets and Results per Datasets," www.colorconstancy.com, accessed: 10-February-2017.

[56] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Automatic color constancy algorithm selection and combination," *Pattern Recognition*, vol. 43, no. 3, pp. 695–705, 2010.

[57] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics and scene semantics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 687–698, 2011.

[58] J. Van De Weijer, C. Schmid, and J. Verbeek, "Using high-level visual information for color constancy," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[59] A. Gijsenij, T. Gevers, and J. Van De Weijer, "Generalized gamut mapping using image derivative structures for color constancy," *International Journal of Computer Vision*, vol. 86, no. 2, pp. 127–139, 2010.

[60] B. V. Funt and W. Xiong, "Estimating illumination chromaticity via support vector regression," in *Color and Imaging Conference*, 2004, pp. 47–52.

[61] A. Chakrabarti, K. Hirakawa, and T. Zickler, "Color constancy with spatio-spectral statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1509–1519, 2012.

[62] M. S. Drew and H. R. V. Joze, "Exemplar-based colour constancy," in *The British Machine Vision Conference*, vol. 26, 2012, pp. 1–12.

[63] D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Effective learning-based illuminant estimation using simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, no. 2, 2015, pp. 1000–1008.

[64] S. Bianco, C. Cusano, and R. Schettini, "Single and Multiple Illuminant Estimation Using Convolutional Neural Networks," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 1–14, 2017.

[65] I. Rafegas, M. Vanrell, L. A. Alexandre, and G. Arias, "Understanding Trained CNNs by Indexing Neuron Selectivity," *Pattern Recognition Letters*, pp. 1–9, 2019.

[66] A. Ng, "Neural Networks and Deep Learning," https://www.coursera.org/learn/neural-networks-deep-learning/home/welcome, accessed: February-2020.

[67] D. M. Hawkins, "The Problem of Overfitting," *Journal of Chemical Information and Computer Sciences*, pp. 1–12, 2004.

[68] Z. Lou, T. Gevers, N. Hu, and M. P. Lucassen, "Color Constancy by Deep Learning," in *Proceedings of the British Machine Vision Conference*, 2015, pp. 7–10.

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.

[70] S. Beigpour, C. Riess, J. Van De Weijer, and E. Angelopoulou, "Multi-illuminant estimation with conditional random fields," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 83–96, 2014.

[71] E. I. Ershov, A. V. Belokopytov, and A. V. Savchik, "Problems of dataset creation for light source estimation," 2020.

[72] C. S. McCamy, H. Marcus, and J. G. Davidson, "A color-rendition chart," *J. Appl. Photogr. Eng.*, vol. 2, no. 3, pp. 95–99, Summer 1976.

[73] Ç. Aytekin, J. Nikkanen, and M. Gabbouj, "INTEL-TUT dataset for camera invariant color constancy research," *CoRR*, vol. abs/1703.09778, 2017.

[74] F. Laakom, J. Raitoharju, A. Iosifidis, J. Nikkanen, and M. Gabbouj, "INTEL-TAU : A Color Constancy Dataset," *arXiv preprint arXiv:1910.10404*, 2019.

[75] D. Cheng, D. K. Prasad, and M. S. Brown, “Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution,” *Journal of the Optical Society of America A*, vol. 31, no. 5, pp. 1049–1058, 2014.

[76] “It 8 colour chart,” http://www.targets.coloraid.de/, accessed: 2020-September.

[77] F. Ciurea and B. V. Funt, “A large image database for color constancy research,” in *Color and Imaging Conference*, vol. 5, 2003, pp. 160–164.

[78] N. Banić, K. Koščević, and S. Lončarić, “Unsupervised Learning for Color Constancy,” in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2017, pp. 1–15.

[79] “Datacolor spydercube,” www.datacolor.com/photography-design/product-overview/spydercube/, accessed: 2020-07-25.

[80] J. Vazquez-Corral, C. A. Parraga, M. Vanrell, and R. Baldrich, “Color constancy algorithms: Psychophysical evaluation on a new dataset,” *Journal of Imaging Science and Technology*, vol. 53, no. 3, 2009.

[81] B. Arad and O. Ben-Shahar, “Sparse Recovery of Hyperspectral Signal from Natural RGB Images,” in *European Conference on Computer Vision*, 2016, pp. 19–34.

[82] A. Gijsenij, T. Gevers, and M. P. Lucassen, “Perceptual analysis of distance measures for color constancy algorithms,” *Journal of the Optical Society of America A*, vol. 26, no. 10, pp. 2243–2256, 2009.

[83] S. D. Hordley and G. D. Finlayson, “Reevaluation of color constancy algorithm performance,” *Journal of the Optical Society of America A*, vol. 23, no. 5, pp. 1008–1020, 2006.

[84] G. D. Finlayson, “Color Constancy and a Changing Illumination,” in *Proc. SPIE 2179, Human Vision, Visual Processing, and Digital Display V*, 1994, pp. 338–343.

[85] CIE, “Commission Internationale de L’Eclairage (CIE), Colorimetry,” Tech. Rep. 15.2, 1986.

[86] G. Sharma, W. Wu, and E. N. Dalal, “The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations,” *Color Research and Application*, vol. 30, pp. 21–30, 2005.

[87] CIE, “Improvement to Industrial Colour-Difference Evaluation.” Tech. Rep. 5, 2001.

[88] M. R. Luo, G. Cui, and B. Rigg, “The development of the CIE 2000 colour-difference formula: CIEDE2000,” *Color Research and Application*, vol. 26, no. 5, pp. 340–350, 2001.

[89] “Rush hour test set from the derf’s test media collection,” https://media.xiph.org/video/derf/, accessed: 2020-September.

[90] J. Vazquez-Corral and M. Bertalmío, “Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications,” *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4564–4575, 2014.

[91] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[92] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[93] C.-S. C. . Y.-P. H. . J.-B. Cheng, “RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229 – 1234, 1999.

[94] M. S. Brown and D. G. Lowe, “Unsupervised 3D object recognition and reconstruction in unordered datasets,” in *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, 2005, pp. 56–63.

[95] G. D. Finlayson, H. Gong, and R. B. Fisher, “Color Homography: Theory and Applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 20–33, 2019.

[96] B. V. Funt and M. Mosny, “Removing outliers in illumination estimation,” in *Color and Imaging Conference*, 2012, pp. 105–110.

[97] A. Gijsenij, T. Gevers, and J. Van De Weijer, “Computational color constancy: survey and experiments,” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2475–2489, 2011.

[98] L. Shi and B. V. Funt, “Re-processed Version of the Gehler Color Constancy Dataset of 568 Images,” https://www2.cs.sfu.ca/~colour/data/shi_gehler/, accessed: 10-February-2017.

[99] J. T. Barron, Personal communication, 2017.

[100] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, “Color Image Processing Pipeline in Digital Still Cameras,” *IEEE signal processing magazine*, no. January 2005, pp. 34–43, 2005.

[101] S. Bianco, “About your paper ‘a curious problem with using the colour checker dataset for illuminant estimation’,” Personal communication by email, january 12, 2018.

[102] D. Coffin, “Decoding raw digital photos in linux,” www.dechifro.org/dcraw/, accessed: 2016-10-01.

[103] Y. Hu, “Fc4: Fully convolutional color constancy with confidence-weighted pooling. faq and code repository,” http:https://github.com/yuanming-hu/fc4, accessed: June-2020.

[104] F. Fang, H. Gong, M. Mackiewicz, and G. D. Finlayson, “Colour Correction Toolbox,” in *13th AIC Congress*, 2017, pp. 13–18.

[105] A. Maus, “Delaunay triangulation and the convex hull of n points in expected linear time,” *BIT Numerical Mathematics*, vol. 24, no. 2, pp. 151–163, 1984.

[106] J. M. DiCarlo, F. Xiao, and B. a. Wandell, "Illuminating Illumination," in *9th Color Imaging Conference*, 2001, pp. 27–34.

[107] G. D. Finlayson, S. D. Hordley, and P. Morovic, "Chromagenic filter design," in *10th Annual Congress of the International Colour Association*, 2005, pp. 1023–1026.

[108] "Kodak wratten filters," https://www.kodak.com/en/motion/page/wratten-2-filters, accessed: 2020-09.

[109] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[110] Y. Rubner, C. Tomasi, and L. J. Guibas, "Earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[111] F. Pitie and A. Kokaram, "The linear Monge-Kantorovitch linear colour mapping for example-based colour transfer," in *IET 4th European Conference on Visual Media Production*, 2007, pp. 23–23.

[112] J. Rabin and N. Papadakis, "Convex color image segmentation with optimal transport distances," in *Scale Space and Variational Methods in Computer Vision. SSVM 2015. Lecture Notes in Computer Science*, vol. 9087, 2015, pp. 256–269.

[113] S. Cohen and L. J. Guibas, "Earth Mover's Distance under transformation sets," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, no. September, 1999, pp. 1076–1083.

[114] Y. Rubner, "Perceptual Metrics for Image Database Navigation," Ph.D. dissertation, 1999.

[115] L. Vasershtein, "Markov processes on countable product space describing large systems of automata (in Russian)," *Problemy Peredachi Infomatsii*, vol. 5, no. 3, pp. 64–73, 1969.

[116] L. Kantorovich, "On translation of mass(in Russian)," *Doklady. Acad. Sci. USSR*, vol. 37, pp. 227–229, 1942.

[117] I. Olkin and F. Pukelsheim, "The Distance between Two Random Vectors with Given Dispersion Matrices," in *Linear Algebra and its Applications*, 1982, pp. 257–263.

[118] I. Boyadzhiev, S. Paris, and K. Bala, "User-assisted image compositing for photographic lighting," *ACM Transactions on Graphics*, 2013.

[119] S. Bianco and R. Schettin, "Color Constancy Using Faces," in *IEEE Conference on Computer Vision and Pattern Recognition*, no. June 2014, 2012, pp. 65–72.

[120] "Tsumura laboratory, chiba university in japan," mi.tj.chiba-u.jp/index-e.html, accessed: 2020-September.

[121] "Horiuchi & hirai laboratory, chiba university in japan," http://dippix.tp.chiba-u.jp/staff_e.html, accessed: 2020-September.

[122] "Thouslite led cube," http://www.thouslite.com/products/16.html?jdfwkey=poobe1, accessed: 2020-September.

[123] F. Matsushita, K. Kiyomitsu, K. Ogawa, and N. Tsumura, "System for evaluating pathophysiology using facial image," in *Color and Imaging Conference*, 2017, pp. 274–279.

[124] ITU-R, "Parameter values for the HDTV standards for production and international programme exchange," *Broadcasting service (television)*, 2015.

[125] B. Mendibru, *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*, 2009.

[126] M. L. Gleicher and F. Liu, "Re-Cinematography: Improving the Camerawork of Casual Video," *ACM Transactions on Multimedia Computing*, vol. 5, no. 1, pp. 1–28, 2008.

[127] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.

[128] R. G. Rodriguez, J. Vazquez-Corral, and M. Bertalmío, "Color matching images with unknown non-linear encodings," *IEEE Transactions on Image Processing*, vol. 29, pp. 4435 – 4444, 2020.

[129] J. Preiss, F. Fernandes, and P. Urban, "Color-image quality assessment: From prediction to optimization," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1366–1378, 2014.

[130] I. Lissner, J. Preiss, P. Urban, M. S. Lichtenauer, and P. Zolliker, "Image-Difference Prediction : From Grayscale to Color," *Image Processing*, vol. 22, no. 2, pp. 435 – 446, 2013.

[131] I. Lissner and P. Urban, "Toward a Unified Color Space for Perception-Based Image Processing," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1153 – 1168, 2012.