Chapter 5

A Color Constancy Algorithm

5.1 Introduction

In a number of applications from machine vision tasks such as object recognition, image indexing and retrieval, to digital photography or new multimedia applications, it is important that recorded colors remain constant under changes in scene illumination. In particular, there exists the problem faced in mobile robotics when a place is newly visited. Besides the fact that objects may have changed their position, the whole room may be viewed under a completely different light, making a hard task the comparison of what is seen at that moment to what was recorded in the database in prior visits.

Furthermore, the movement of the camera around a place – a room or pathway – can also change the way colors in a scene are perceived. Hence, a preliminary step when using color must be to remove the pernicious effect of the illumination change. As already introduced in Chapter 2 and deeply studied in Chapter 4, this problem is usually referred to in literature as *color constancy*.

Generally, the color constancy task is to recover a *descriptor* for each surface in a scene as it would be seen by the camera under a canonic illuminant taken as a reference. This is similar to pose the color constancy problem as that of recovering an estimate of the color of the scene illumination from an image taken under an unknown illumination. As shown in Chapter 4, it is relatively straightforward to map image colors onto illuminant independent descriptors afterwards [FHH99, FHH01]. Therefore, *finding both a color mapping and the color of the scene illuminant are equivalent problems.* As will be seen later, there are plenty of approaches following either definition.

A lot of efforts were put in this problem in recent years [LM77, MW86, DI93, BF97, Buc80, Tom91, Sha85, For90, Bar99b, Bar00, BCF02, BMCF02, BFM00, FDF94a, Fin95a, Fin95b, Fin96, FH99, FH00, FHH99, FHH01, Sap99]. Part of the difficulty is that this problem is entangled with other confounding phenomena such as the shape of objects, viewing and illumination geometry, besides changes in illumination spectral power distributions and reflectance properties of the imaged objects.

Thus, in order to ease the problem many researchers [LM77, MW86, For90,

DI93] considered a simplified two–dimensional world in which all the objects were taken as flat, matte, Lambertian, uniformly illuminated surfaces, which is usually referred to as the *Mondrian* world. If any of those *a priori* requirements is not completely fulfilled it might be necessary to carry out a process to specifically remove some of the distorting effect such as *highlights* or *interreflections* [BLL96].

Let us briefly sketch some of the most important approaches, despite we compellingly refer to Chapter 2 for a complete state of the art on that issue. Land [LM77] assumed that every image contained a white patch. Another assumption [Buc80, GJT88] was that the average reflectance of all surfaces in a scene was *achromatic*, i.e., gray. In this case, the average color of the light leaving the surface would be the color of the incident illumination. Yet, another approach in [MW86] described lights and surfaces using finite–dimensional linear models to develop algebraic recovery schemes.

Other authors tried to exploit features not present in the idealized Mondrian world, such as *specularities* [Sha85], *shadows* [FF94], *varying illumination* [BFF97], or *mutual illumination* [FDH91], to recover information about the scene illumination. As reported in [FHH99, FHH01], the main drawback of all these algorithms is that their assumptions are often violated in real images. On the other hand, among those that can work on real images, their performance is still far from being good enough [FBM98].

As an alternative, there exist some algorithms which do not try looking for a unique answer, but for the *likeliest* solution. For example, the *gamut–mapping* algorithm developed by Forsyth [For90] and expanded later by Finlayson [FDF94a, Fin95a, Fin95b, Fin96, FH99, FH00] chooses a solution from the set of feasible solutions following diverse criteria of selection.

Other authors [BF97, Sap99] posed the problem in a probabilistic framework. More recently, Sapiro [Sap99] developed an algorithm based on the probabilistic Hough transform and Finlayson also used this framework in his *color–by–correlation* algorithm [FHH99, FHH01]. The neural network approach [CFB98] to color constancy can similarly be seen as a method of dealing with the inherent uncertainty of this problem. All these algorithms represent some improvements in the color constancy problem, but some further work must be still done to find a definitive solution [FHH99, FHH01].

In this Chapter, we propose a new color constancy algorithm based on the computation of the histogram of feasible mappings. The goal of this algorithm is to be used in tasks involved in mobile robotics where color and its stability are important issues. Thus, it is a must that this approach works only with the data present in the image pixels, besides employing as little *a priori* knowledge about illumination as possible, since this is the quotidian case. Our algorithm is capable of recovering a mapping which can change image colors as they would be seen under a canonic light by means of a single canonic image picturing a similar scene.

5.2 Outline of the Chapter

This Chapter is totally addressed to the description of our color constancy algorithm as well as the study of its performance in comparison to another well–known algorithm, namely, the Finlayson's 2D gamut–mapping. In Section 5.3 a

further look into some of the most interesting approaches for our taste and needs is carried out. Particularly, we present a deep insight into the gamut–mapping family of algorithms as well as a description of the color–by–correlation and the color&illuminant voting approaches. In Section 5.4 a discussion is held about the problems we must face in this Chapter and how color constancy can really help. Afterwards, our approach is fully developed in Section 5.5. Performance of this algorithm is tested in Section 5.6 side by side the aforementioned Finlayson's 2D gamut–mapping. Finally, in Section 5.7 we consider the work developed in this Chapter and some conclusions are summarized afterwards.

5.3 Related Previous Work

We review in this Section three of the most interesting color constancy algorithms in order to make clear the similarities and divergences to ours suggested later on. The algorithms described are, namely, gamut–mapping, color–by–correlation and color&illuminant voting.

To our knowledge, color–by–correlation is so far the algorithm that performs best, while the gamut–mapping family forms an important set of algorithms worth to take into consideration [FHH01, HF04]. While the former selects from a set of feasible candidates the illuminant that best correlates with the colors appearing in the images, the latter computes, in fact, an estimate of the illuminant. For us, this is a better solution in case the set of feasible illuminants is only approximately known, since color correlation can only find an illuminant out of a discrete set of feasible lights that must be known *a priori*, despite the correlation approach tends to win in that kind of experiments.

The third algorithm, named color&illuminant voting, has also been considered, though its performance is not as good as that of the others. Nevertheless, it illustrates a useful way to improve the gamut–mapping kind of algorithms, that is, by voting each recovered mapping and selecting only one accordingly to a meaningful poll.

5.3.1 Gamut–Mapping Algorithms

Given a set of image colors which were recorded picturing a scene under an unknown illumination, we want to find the mapping T which, when applied to these sensor values, produces the response that would be recorded under a canonic illuminant. Forsyth [For90] and Finlayson [FDF94a, Fin95a, Fin95b, Fin96, FH99, FH00] represented a change in illumination as a diagonal matrix $\mathbf{D} \in \mathcal{D}_3(\mathbb{R})$ mapping the gamut of image colors $\Gamma(\mathcal{I})$ to a gamut of *canonic* colors $\Gamma(\mathcal{C})$. The first stage of the algorithm is to find the set of all feasible matrices $\mathbf{D}$ such that $\forall \mathbf{c} \in \Gamma(\mathcal{I})$, $\mathbf{D}\,\mathbf{c}^t \in \Gamma(\mathcal{C})$, where $\mathbf{c} = (R, G, B)$ is the sensor response.

Canonic Gamut $\Gamma(\mathcal{C})$

The *canonic gamut* is defined to be the set of all feasible sensor responses obtained by viewing the set of all physically realizable surface reflectance under a canonic illuminant E_0. This set is approximated as a *convex combination* of

the set $C = \{\mathbf{c}_i^C\}_{i=1,\ldots,n}$ of all the responses to n surface

$$\Gamma(C) = \{\sum_{i=1}^{n} \omega_i \mathbf{c}_i^C \mid \mathbf{c}_i^C \in C\,,\ \omega_i \geq 0\ \forall i = 1,\ldots,n \text{ and } \sum_{i=1}^{n} \omega_i = 1\} \tag{5.1}$$

Since $\Gamma(C)$ is also a convex set, we only need to take into account the set $\mathcal{C}$, i.e., the group of points defining the convex hull of C. Therefore, $\Gamma(\mathcal{C}) = \Gamma(C)$.

Image Gamut $\Gamma(\mathcal{I})$

The *image gamut* is defined in a similar way from the set of image colors I recorded under an unknown illuminant. Then, all convex combinations of I, $\Gamma(I)$, could appear under this illuminant. Again, the convexity of $\Gamma(I)$ makes possible the mere consideration of the convex hull $\mathcal{I}$ instead of I.

Set of Feasible Mappings $\Gamma(\mathcal{C}/\mathcal{I})$

As said, a diagonal matrix $\mathbf{D}$ is a feasible solution to the color constancy problem if it maps a point in the image gamut somewhere onto the canonic gamut. The key idea of this approach is that of using only the points of the image and the canonic convex hull to compute the set of all *feasible mappings*.

For each point in the convex hull of the image gamut, $\mathbf{c}_i^{\mathcal{I}} \in \mathcal{I}$, a set of mappings taking this point onto the convex hull of the canonic gamut $\mathcal{C}$ can be generated by dividing, componentwise, each point in $\mathcal{C}$ by any $\mathbf{c}_i^{\mathcal{I}} \in \mathcal{I}$,

$$\mathcal{C}/\mathbf{c}_i^{\mathcal{I}} = \{\mathbf{D}^j \in \mathcal{D}_3(\mathbb{R}) \mid \mathbf{D}_k^j = c_{j,k}^{\mathcal{C}}/c_{i,k}^{\mathcal{I}}\,,\ \forall \mathbf{c}_j^{\mathcal{C}} \in \mathcal{C} \text{ and } 1 \leq k \leq 3\} \tag{5.2}$$

Therefore, $\Gamma(\mathcal{C}/\mathbf{c}_i^{\mathcal{I}})$ is the set of all convex combinations of the set $\mathcal{C}/\mathbf{c}_i^{\mathcal{I}}$. Mappings in these sets represent feasible illuminants under which the surface could be viewed. It follows then that the *intersection* of all these sets of mappings gives rise to the set of *feasible mappings*

$$\Gamma(\mathcal{C}/\mathcal{I}) = \bigcap_{i=1}^{n} \Gamma(\mathcal{C}/\mathbf{c}_i^{\mathcal{I}}) \tag{5.3}$$

Illumination Gamut $\Gamma(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$

The set of feasible mappings can be further constrained if some restrictions on the range of illuminants that may occur in the real world are taken into account. Feasible illuminants can be characterized by considering a surface under a representative set of usual illuminants. If there are m such illuminants, then there will be a set $E = \{\mathbf{e}_i\}_{i=1,\ldots,m}$ of colors. The set of all convex combinations of the convex hull of E, denoted $\mathcal{E}$, is the *illuminant gamut* $\Gamma(\mathcal{E})$.

If $\mathbf{c}^{\mathcal{C}}$ is the color of the canonic illuminant, there exists a convex set of diagonal mappings $\Gamma(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$ taking this color to any other color in $\Gamma(\mathcal{E})$. Therefore, if $\mathbf{D}$ is a feasible mapping taking the image gamut into the canonic gamut, it must fulfill the illuminant constraint, i.e., its inverse mapping $\mathbf{D}^{-1}$ must take the canonic illuminant to the scene illuminant in the illuminant gamut. That is $\mathbf{D}^{-1} \in \Gamma(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$. If $\Gamma^{-1}(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$ is the *nonconvex* set of diagonal mappings taking illuminants in the illuminant gamut to the canonic gamut, then

$$\mathbf{D} \in \Gamma^{-1}(\mathcal{E}/\mathbf{c}^{\mathcal{C}}) \Rightarrow \mathbf{D}^{-1} \in \Gamma(\mathcal{E}/\mathbf{c}^{\mathcal{C}}) \tag{5.4}$$

Hence, the illuminant constraint can be enforced by intersecting the sets $\Gamma(\mathcal{C}/\mathcal{I})$ and $\Gamma^{-1}(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$. It must be noted that $\Gamma^{-1}(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$ is not a convex set, which implies the intersection is not as easy as it would be in the convex case.

Selection of One Single Mapping

Once the set of feasible mappings is found, it is necessary to select one single mapping out of the whole set to provide a unique estimate of the unknown illuminant. Forsyth employed the heuristic by which the volume of $\Gamma(\mathcal{I})$ mapped within $\Gamma(\mathcal{C})$ is maximum. Finlayson – following the Forsyth's approach – applied the heuristic of choosing the mapping which makes the images as colorful as possible, corresponding to maximizing the area of the canonic set occupied by the image gamut under the diagonal mapping.

In absence of any other information, it is more sensible to assume that all mappings and their corresponding illuminants are equally likely. With this assumption, it is natural to use the mean value of the mapping set as the estimate – as Barnard in [BCF02, BMCF02]. In [FH99] it was shown that, in general, the mean estimate does provide better color constancy results than the Forsyth's heuristic.

Afterwards, Finlayson [Fin96] reduced the dimension of the color data from $3D$ to $2D$ by using a kind of chromaticity coordinates he called *perspective color*. The goal of those coordinates was to improve the performance of the method in order to cope with images showing specularities, shape information, and varying illumination intensity. Theoretically, these factors only affect the intensity of the recovered illuminant, rather than its color. This observation leads to propose a similar algorithm where color (R, G, B) has been changed into the perspective coordinates $(R/B, G/B)$.

Since this transform conserves the convexity of gamuts, the Forsyth's algorithm can be used straightforward. So, the result of finding the feasible mappings $\Gamma(\mathcal{C}/\mathcal{I})$ is a set of diagonal mappings, no matter it is a $3D$ set found by the Forsyth's algorithm or the $2D$ set found in the perspective case. Later, Finlayson and Hordley [FH99, FH00] proved an important result, namely, the set of $3D$ diagonal matrices recovered using the Forsyth's algorithm, when projected into a $2D$ chromaticity space, is identical to the set of $2D$ diagonal matrices recovered by the Finlayson's algorithm using perspective color.

As a consequence, if the intensity of the scene illumination is not strictly necessary or recovered by an alternative way, it is far better working in a $2D$ chromaticity space because computations there are easier than in $3D$.

5.3.2 Color–by–Correlation

In a number of papers [FHH99, FHH01] Finlayson et al. propose a whole framework for color constancy which includes most of the existing algorithms. The key idea of the framework is that of recovering the illuminant from an image of a scene taken under an unknown light from a set of known feasible illuminants. Once the best correlated light is selected, it is relatively straightforward to transform image colors into illuminant independent descriptors.

In order to find out which is the illuminant belonging to the scene, the set of colors present in the image is compared to the set of colors that each illuminant generates, and the one with the greatest correlation, i.e., with most

common colors, is taken as the resulting illuminant. The set of colors generated by each feasible illuminant is coded in the columns of a *correlation matrix*. This approach is able to encompass various color constancy algorithms by the definition of different correlation matrices.

Formally, the idea is to find the illuminant $\hat{E}$ from a set of feasible lights which correlates better with the colors in an image I. First, the algorithm computes the image color histogram[1] $\mathcal{H}_I$ which is then *truncated* into $\tilde{\mathcal{H}}_I$, only showing what colors are present in the image. Since the color space has been partitioned into a finite set of n discrete cells representing all the feasible colors, histograms can be coded as vectors of dimension n, i.e., $\mathbf{h}\,,\ \tilde{\mathbf{h}} \in \mathbb{R}^n$, respectively.

Later, the *likelihood* of observing a certain color under a given light, for each of a set of m illuminants and n feasible colors, forms the elements $\mathbf{Q}_{ij}$ of a correlation matrix $\mathbf{Q} \in \mathcal{M}_{n\times m}(\mathbb{R})$. Therefore, the probability distributions for each light are coded in the columns $\mathbf{q}_i$ of matrix $\mathbf{Q}$.

Hence, a correlation measure between image colors in $\tilde{\mathbf{h}}$ and the color distributions for each illuminant in $\mathbf{Q}$ is needed to estimate the scene illuminant. Finlayson uses a *dot–product* between vectors. The elements of the vector $\tilde{\mathbf{h}}\,\mathbf{Q}$ are a measure of how strongly the image data correlates with each of the feasible illuminants. Since matrix $\mathbf{Q} = (\mathbf{q}_1^t \mid \cdots \mid \mathbf{q}_m^t)$, the correlation measure can be separately computed for the i^{th} feasible illuminant as the product $\tilde{\mathbf{h}}\,\mathbf{q}_i^t$.

The final stage in solving the color constancy problem is to recover an estimate of the scene illuminant based on the correlation information. Finlayson chooses the illuminant $\hat{E}$ that is the most highly correlated with the data in the image

$$\text{find } \hat{E} = E_k \quad \text{so that} \quad k = \underset{1 \le i \le m}{\text{argmax}} \ \{\tilde{\mathbf{h}}\,\mathbf{q}_i^t\} \tag{5.5}$$

To completely specify the solution, it is necessary to properly define the elements $\mathbf{Q}_{ij}$ of the correlation matrix. Those elements are related to the likelihood of finding a color under a certain illuminant. Finlayson proposes a Bayesian framework to define this likelihood function.

In a Bayesian color constancy framework, given an image I and the set of its colors $\mathcal{C}$, the probability $Pr(E \mid \mathcal{C})$ must be recovered in order to find the best illuminant E. If we knew the probability of observing a certain set of colors $\mathcal{C}$ under an illuminant E, $Pr(\mathcal{C} \mid E)$, the Bayes's rule would tell us how to calculate the corresponding probability $Pr(E \mid \mathcal{C})$, that is,

$$Pr(E \mid \mathcal{C}) = \frac{Pr(\mathcal{C} \mid E)\, Pr(E)}{Pr(\mathcal{C})} \tag{5.6}$$

Here, $Pr(E)$ is the probability that the scene illuminant be E and $Pr(\mathcal{C})$ is the probability of observing a set of image data $\mathcal{C}$. As these are constant for a given image under the hypothesis that colors are independent and illuminants are all equally likely, the above equation can be rewritten as

$$Pr(E \mid \mathcal{C}) \propto \prod_{\forall \mathbf{c} \in \mathcal{C}} Pr(\mathbf{c} \mid E) \tag{5.7}$$

[1] Despite perspective color coordinates are used in [FHH99, FHH01], other color coordinates could also be used instead.

Now, the likelihood function can be defined as

$$\mathcal{L}(E \mid \mathcal{C}) = \sum_{\forall \mathbf{c} \in \mathcal{C}} log(Pr(\mathbf{c} \mid E)) \tag{5.8}$$

This way, the elements of the correlation matrix are defined as

$$\mathbf{Q}_{ij} = log(Pr(\mathbf{c}_i \mid E_j)) \tag{5.9}$$

and the correlation between image colors and illuminants can be expressed in terms of a log–likelihood as $\mathcal{L}(E_j \mid \mathcal{C}) = \tilde{\mathbf{h}}\,\mathbf{q}_j^t$. Therefore, it follows that Eq. (5.5) becomes a maximum–likelihood solution to the illuminant estimation problem

$$\hat{E} = \underset{1 \leq j \leq m}{\text{argmax}} \quad \{\mathcal{L}(E_j \mid \mathcal{C})\} \tag{5.10}$$

In [FHH99, FHH01] it is also claimed that several other color constancy algorithms can be slipped into this framework by changing the correlation matrix properly and some examples are provided, such as gray–world, gamut–mapping, color&illuminant voting and neural network approaches.

5.3.3 Color and Illuminant Voting

Let us now have a look into the Sapiro's *color&illuminant voting* algorithm [Sap99]. In order to estimate the illuminant in a given image this approach selects possible values for the reflectance $\mathbf{r}$ and the sensor data $\mathbf{y}$, which are used to compute a vote for the illuminant $\mathbf{e}$. Following the *Hough* transform, the estimated illuminant is assigned to the parameter with a maximal vote.

In contrast to the *gamut–mapping* algorithm, in that case it is the data for each of the pixels in the image that is used in the voting. In addition, this framework can handle images with more than one illuminant without any prior segmentation since more than a local maximum can be found in the parameter space.

First of all, let us describe the image formation process which is modeled by the bilinear equation, already explained in Section 4.8.1. Both illuminant power distributions and reflectance functions are described as a linear combination of a set of basis functions. The number and nature of these functions may be different in general and this approach referrers to quite well–established ones, i.e, those by Judd et al. [JMW64] for the illuminants, and those by either Cohen [Coh64] or Vrhel et al. [VGI94] for the reflectances.

This way, if $\{E_i(\lambda)\}_{i=1,\ldots,n}$ and $\{R_j(\lambda)\}_{j=1,\ldots,m}$ are respectively the illuminant and reflectance basis functions, any particular illuminant spectral power distribution $E(\lambda)$ and surface reflectance function $R(\lambda)$ can be approximated as

$$E(\lambda) \approx \sum_{i=1}^{n} e_i\, E_i(\lambda) \quad \text{and} \quad R(\lambda) \approx \sum_{j=1}^{m} r_j\, R_j(\lambda) \tag{5.11}$$

Any light function $I(\lambda)$ reflected by any surface with reflectance $R(\lambda)$ viewed under illuminant $E(\lambda)$ is the product of these two functions, that is, $I(\lambda) = E(\lambda)\, R(\lambda)$. Besides, an imaging device with p linearly independent spectral

sensitivities $\{S_h(\lambda)\}_{h=1,\dots,p}$ responds to the reflected light $L(\lambda)$ by producing p quantities $\{y_h\}_{h=1,\dots,p}$

$$y_h = \int_{\lambda_0}^{\lambda_1} I(\lambda)\, S_h(\lambda)\, d\lambda = \int_{\lambda_0}^{\lambda_1} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} e_i\, r_j\, E_i(\lambda)\, R_j(\lambda) \right] S_h(\lambda)\, d\lambda \qquad (5.12)$$

In order to simplify the notation and to separate the unknown values from the known functions, this work introduces the *bilinear model matrices* $\mathbf{K}_j$ defined in the following way

$$(\mathbf{K}_j)_{hi} = \int_{\lambda_0}^{\lambda_1} E_i(\lambda)\, R_j(\lambda)\, S_h(\lambda)\, d\lambda\,,\ j = 1, \dots, m \qquad (5.13)$$

The image formation model can then be written as

$$\mathbf{y}^t = \left(\sum_{j=1}^{m} r_j\, \mathbf{K}_j \right) \mathbf{e}^t = \mathbf{K}\, \mathbf{e}^t \qquad (5.14)$$

where $\mathbf{y} = (y_1, \dots, y_p)$ and $\mathbf{e} = (e_1, \dots, e_n)$. That is, from a set of values describing a surface reflectance $\mathbf{r} = (r_1, \dots, r_m)$, we can compute its appearance $\mathbf{y}$ under a certain illuminant $\mathbf{e}$ by computing the matrix $\mathbf{K}$ and projecting the illuminant vector $\mathbf{e}$ afterwards, i.e., $\mathbf{y}^t = \mathbf{K}\, \mathbf{e}^t$.

The basic illuminant estimation algorithm proposed in this approach also works in the reverse way, i.e., a matrix $\mathbf{K}$ is obtained for each color $\mathbf{y}$ and its pseudoinverse $\mathbf{K}^+$ is computed to backproject the color $\mathbf{y}$ into the illuminant space, $\mathbf{e}^t = \mathbf{K}^+\, \mathbf{y}^t$. Therefore, the illuminant $\mathbf{e}$ gets *voted*. To get an estimate of the true illuminant in the scene, a poll for each illuminant is calculated by adding up the votes from all the image colors. The illuminant with the maximum number of votes is selected as the scene illuminant.

For each sensor response $\mathbf{y}$ at each image pixel, many reflectance vectors $\mathbf{r}$ are selected to compute matrix $\mathbf{K}$ and so many votes are cast. The simplest way to proceed is to randomly select them, despite no specific information is used in that case. Nonetheless, a better strategy could be employed in case a probability distribution $P(\mathbf{r})$ is known or, even far better, if a conditional distribution $P(\mathbf{r} \mid \mathbf{y})$ is somehow obtained, since the possible surface reflectances would be thus conditioned by the data observed.

Besides constraining reflectances, this scheme is also able to improve its efficiency by restricting illuminants. If the set of feasible illuminants is known, then we only need to estimate the parameters which are closer to it. Therefore, any recovered illuminant $\mathbf{e}$ will be voted only if it represents one of these feasible illuminants.

The point in this work is, as a consequence, the introduction of the probabilistic Hough transform to estimate the illumination of an image as well as illustrating how to employ probability models and physical constraints as a pretty straightforward way to improve the precision of the recovery. Nevertheless, and despite its huge interest, the results exhibited in that work are still quite scarce in our opinion, which does not help to completely evaluate the actual performance of this algorithm in a more realistic problem.

5.4 Discussion

It is now the time to discuss the difficulties we found in the three algorithms that were described above. These problems and our aim to use color constancy in a mobile robotics context is what have conveyed us to propose the algorithm in the next Section.

First of all, those approaches, especially gamut–mapping and color–by–correlation, rely as a rule on the fact that the set of *all* possible colors seen under a canonic illuminant is, somehow, known and available *a priori*. Gamut–mapping takes advantage of them to constrain the set of feasible mappings, while color–by–correlation builds the correlation matrix up with them, implying that those colors must be known for all the illuminants taken into account.

Besides, the selection strategy is somehow blind because there is no information about what mappings among the feasible ones are likelier to be the correct result. Color&illuminant voting furthermore needs to know the relation between colors and surface reflectances to try a guess on the illumination, which seems quite unlikely in a real case[2].

Secondly, the set of all realizable illuminants needs to be known *a priori* in the gamut–mapping family of algorithms in order to restrict the feasible transformations and get an acceptable result. Besides, while this set is approximated as a *convex hull* in gamut–mapping, it is just a finite set in the color–by–correlation algorithm not covering any other intermediate illuminant that might be encountered. Moreover, the color&illuminant voting algorithm uses basis functions to precompute matrices $\mathbf{K}_j$, which might not exactly correspond to what is found in a real situation.

In short, before any of the previous color constancy algorithms can even be put to work, a rather big chunk of *a priori* knowledge about reflectances and lights is needed. In our opinion and experience, this is neither always available nor simple to get. To make our view clearer, we point out the lack of this kind of information in two basic tasks where a mechanism of color constancy would be helpful, namely, indexing and retrieval of images from a database and color–based recognition of objects [BFM00, FBM98].

It may be very difficult or simply impossible to have an *a priori* realistic database of surface and illuminant colors for both tasks. While indexing may be using images of unknown origin, the recognition task may be part of a higher layer of routines where light conditions are uncontrollable and unknown, such as the images gathered from a mobile robot moving through an unfamiliar environment.

Thus, our aim in this Chapter is to suggest a less information–dependent color constancy algorithm that straightforwardly relies on pixel colors and is capable of rendering images from an unknown illumination back to a task–dependent canonic illuminant. As a last requirement for our algorithm, it will be able to further restrict the set of feasible illuminations as a helpful step to find more accurate results for the color constancy problem, provided that information is fully at hand.

[2] If we knew this, we would already had a surface descriptor and color constancy would be solved.

5.5 The Color Constancy Algorithm

This Section is completely devoted to the introduction of our color constancy algorithm. As mentioned before, the main goal of this algorithm is to recover colors in an image as they would be seen under a canonic illumination. This capacity may help to identify objects in actual images based on the *a priori* knowledge collected in a database. An important restriction in our case is that this method must be applicable in tasks involved in mobile robotics, where little control and information about the lighting conditions are available. The algorithm must thence rely on the more volatile data provided by image pixels.

As soon will be seen, this algorithm has a number of similarities to the other three just taken into account. The main idea is inherited from the gamut–mapping approach, namely, finding a single map relying on the raw data only provided by images. Nevertheless, the procedure followed by our approach is not of the kind that is employed by the gamut–mapping methods, which consists in a chain of intersections between convex hulls representing successive sets of feasible color maps that most of the times must be relaxed to get an acceptable result.

We additionally bring into play the idea already developed in both color&illuminant voting and color–by–correlation that gives to each feasible mapping a particular weight, or more precisely expressed, a likelihood. However, our approach touches upon this aspect in a very different way than the way is done by the correlation matrix in color–by–correlation. We explicitly compute an approximation to the histogram of feasible color maps out of the sets of colors, instead of guessing the illuminant that correlates best with the colors in the image.

In order to decide what maps could transform the colors from an image onto those under a canonic light we need a proper scheme to estimate those mappings like in the color&illuminant voting approach, where for each color in the scene an illuminant is tried out. The more an illuminant is voted, the likelier it is the scene illuminant. The present algorithm has further developed the idea of recovering an estimate from raw color data as well as the selection of a certain mapping based on the number of times it appears. It must be stated here there is a close equivalence between mappings and illuminants that will make us consider them as the same thing.

The ideas and first versions of our approach, as well as some prior results, have already been published in a number of papers [VLS03a, VLS03b, VLS04]. Nevertheless, again we describe our approach in a more theoretical manner in the next paragraphs, whilst new and more extensive results that have been lately developed are shown in the Section of results.

5.5.1 Some Definitions

We now sketch the basis of our color constancy algorithm. Let I_c and I_a be the *canonic* and the *actual* images, respectively, picturing similar scenes under two different illuminations. The number of objects in the scenes do not matter since no segmentation is needed, being the algorithm based only on raw data from image pixels. Our aim is to find the likeliest color transformation T which maps the pixel colors of image I_a as close onto those of image I_c as possible.

First, we introduce some notation and definitions. We note as $\mathcal{I} \subset \mathbb{R}^d$ a set

of colors, where d is the dimension of the color space. The origin of this set can be from either a specified color gamut or an image. Then, we can get the color histogram from $\mathcal{I}$, noted as $\mathcal{H}_{\mathcal{I}}$. If a mapping $\mathrm{T} \in <\mathrm{T}>$ is applied to each color in $\mathcal{I}$, its transformed set $\mathrm{T}(\mathcal{I})$ is obtained as a result. $<\mathrm{T}>$ is the set of all *feasible color mappings*. Analogously, $\mathrm{T}(\mathcal{H}_{\mathcal{I}})$ represents the transformation of the histogram $\mathcal{H}_{\mathcal{I}}$ by the mapping T.

In general, given two color sets, $\mathcal{I}_a$ and $\mathcal{I}_c$, a *model of color change* consists in a mapping $\mathrm{T} \in <\mathrm{T}>$ so that

$$\begin{array}{rcl} \mathrm{T}\colon \mathcal{I}_a & \longrightarrow & \mathcal{I}_c \\ s & \longmapsto & \mathrm{T}(s) = q \end{array} \tag{5.15}$$

where $s \in \mathcal{S}$ and $q \in \mathcal{Q}$ are two *corresponding* colors. Besides, the set of feasible mappings can be defined as

$$<\mathrm{T}> = \{\mathrm{T} = \Delta(\mathcal{S}, \mathcal{Q}) \mid \forall \mathcal{S} \subset \mathcal{I}_a \text{ and } \forall \mathcal{Q} \subset \mathcal{I}_c\} \tag{5.16}$$

where Δ is a *mapping estimation* scheme – a function or an algorithm – computing one single mapping T out of two corresponding color sets $\mathcal{S}$ and $\mathcal{Q}$.

5.5.2 Likelihood Function

The color constancy algorithm must select the likeliest transformation $\hat{\mathrm{T}}$ from the set $<\mathrm{T}>$. That is, if a likelihood function $\mathcal{L}_\Delta(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c)$ is defined for every mapping $\mathrm{T} \in <\mathrm{T}>$, the algorithm must select the mapping that

$$\hat{\mathrm{T}} = \underset{\mathrm{T} \in <\mathrm{T}>}{\operatorname{argmax}} \{\mathcal{L}_\Delta(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c)\} \tag{5.17}$$

Any likelihood function $\mathcal{L}_\Delta$ can be related to a probability function Pr in the way $\mathcal{L}_\Delta(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c) = log(\Pr(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c))$, noting that the mapping which maximizes $\mathcal{L}_\Delta(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c)$ will also maximize $\Pr(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c))$, and *vice versa*.

Hence, we must first get a value for the probability of any mapping T. As an estimate of the probability distribution of $\Pr(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c)$ we use the *histogram* of the set of feasible mappings, noted as $\mathcal{H}_{<\mathrm{T}>}$. In order to compute the frequencies for those maps we employ the colors in the sets $\mathcal{I}_a$ and $\mathcal{I}_c$, as well as the mapping estimator Δ. *The point is that the likelier a mapping is, the more frequent it should appear in the histogram* $\mathcal{H}_{<\mathrm{T}>}$, and the other way round too.

It is obvious that a particular mapping T can be produced from different groups of corresponding colors. Thus, it is useful to define the following set $\Delta^{-1}(\mathrm{T}) = \{(\mathcal{S}, \mathcal{Q}) \in 2^{\mathcal{I}_a} \times 2^{\mathcal{I}_c} \mid \Delta(\mathcal{S}, \mathcal{Q}) = \mathrm{T}\}$ as the set of all pairs $(\mathcal{S}, \mathcal{Q})$ giving rise to a certain mapping T by using the mapping estimator Δ. The set $2^{\mathcal{I}} = \{A\colon \forall A \subset \mathcal{I}\}$ stands for the *power set* of $\mathcal{I}$, i.e., the set of all subsets of $\mathcal{I}$.

The set $\Delta^{-1}(\mathrm{T})$ is equivalent to T and can be taken instead of the latter since it is true that $\Delta^{-1}(\mathrm{T}) = \Delta^{-1}(\mathrm{T}') \Leftrightarrow \mathrm{T} = \mathrm{T}'$. Hence, the probability $\Pr(\mathrm{T} \mid \mathcal{I}_a, \mathcal{I}_c)$ is computed as $\Pr(\Delta^{-1}(\mathrm{T}) \mid \mathcal{I}_a, \mathcal{I}_c)$.

In practice, since our color sets are discrete, $\Delta^{-1}(\mathrm{T})$ can be thought as a finite disjoint union of singletons $\{(\mathcal{S}, \mathcal{Q})\}$, where each set is a combination of colors from sets $\mathcal{I}_a$ and $\mathcal{I}_c$, respectively. Besides, to compute the probability

$\Pr((\mathcal{S},\mathcal{Q}) \mid \mathcal{I}_a,\mathcal{I}_c)$ each singleton $\{(\mathcal{S},\mathcal{Q})\}$ gets further divided into two independent sets, namely, $\mathcal{S} \subset \mathcal{I}_a$ and $\mathcal{Q} \subset \mathcal{I}_c$. Therefore,

$$\Pr((\mathcal{S},\mathcal{Q}) \mid \mathcal{I}_a,\mathcal{I}_c) = \Pr(\mathcal{S} \mid \mathcal{I}_a) \cdot \Pr(\mathcal{Q} \mid \mathcal{I}_c) \tag{5.18}$$

Then, we get that

$$\Pr(\mathrm{T} \mid \mathcal{I}_a,\mathcal{I}_c)) = \sum_{\forall(\mathcal{S},\mathcal{Q})\in\Delta^{-1}(\mathrm{T})} \Pr(\mathcal{S} \mid \mathcal{I}_a) \cdot \Pr(\mathcal{Q} \mid \mathcal{I}_c) \tag{5.19}$$

That is, the frequency of the bin corresponding to a mapping T in the histogram of feasible mappings $\mathcal{H}_{<\mathrm{I}>}$ is computed adding the product of the frequencies corresponding to all pairs $\mathcal{S}$ and $\mathcal{Q}$ that gave rise to the mapping T by way of the mapping estimation scheme Δ.

$\Pr(\mathcal{S} \mid \mathcal{I}_a)$ and $\Pr(\mathcal{Q} \mid \mathcal{I}_c)$ are computed using as approximations the frequency of the corresponding bins in histograms $\mathcal{H}_{\mathcal{I}_a}$ and $\mathcal{H}_{\mathcal{I}_c}$, namely, $\mathcal{H}_{\mathcal{I}_a}(s)$ and $\mathcal{H}_{\mathcal{I}_c}(q)$, $\forall s \in \mathcal{S}$ and $\forall q \in \mathcal{Q}$, respectively. For example, in case $\mathcal{S} = \{s\}$ it is straightforward that $\Pr(\mathcal{S} \mid \mathcal{I}) \approx \mathcal{H}_{\mathcal{I}}(s)$. In general, if $\mathcal{S} = \{s_i\}_{i=1,\ldots,n}$, the probability of the set under the hypothesis of independence of colors is

$$\Pr(\mathcal{S} \mid \mathcal{I}) \approx \prod_{i=1}^{n} \mathcal{H}_{\mathcal{I}}(s_i) \tag{5.20}$$

Some spurious peaks may usually appear in $\mathcal{H}_{<\mathrm{I}>}$, which might mislead the algorithm. In order to improve the robustness of Eq. (5.19) a measure of similarity between the transformed set $\mathrm{T}(\mathcal{I}_a)$ and the canonic set $\mathcal{I}_c$ is taken into account, which helps to evaluate the performance of a particular mapping T. We use the Swain&Ballard intersection–measure between histograms defined in [SB90] as

$$\cap(\mathcal{H}_1,\mathcal{H}_2) = \sum_k \min\{\mathcal{H}_1(k),\mathcal{H}_2(k)\} \in [0,1] \tag{5.21}$$

The advantage of this similarity measure[3] is that if compared to other matching functions it is very fast to compute [SC96a]. Besides, if the histograms are sparse and colors equally probable, this method is a robust way of comparing images [SB90, SC96a]. This step helps in practice to eliminate outlier mappings among the set of candidates.

As a consequence, we finally define our likelihood function by joining the probability of a mapping and its performance into one single expression as follows

$$\mathcal{L}_\Delta(\mathrm{T} \mid \mathcal{I}_a,\mathcal{I}_c) = log(\cap(\mathrm{T}(\mathcal{H}_{\mathcal{I}_a}),\mathcal{H}_{\mathcal{I}_c})) \cdot \Pr(\mathrm{T} \mid \mathcal{I}_a,\mathcal{I}_c)),\ \forall \mathrm{T} \in <\mathrm{T}> \tag{5.22}$$

where $\mathrm{T}(\mathcal{H}_{\mathcal{I}_a})$ is the transformation of the histogram $\mathcal{H}_{\mathcal{I}_a}$ by a mapping T and $\Pr(\mathrm{T} \mid \mathcal{I}_a,\mathcal{I}_c))$ is the frequency of T in the histogram $\mathcal{H}_{<\mathrm{I}>}$ computed with Eq. (5.19).

5.5.3 Color Coordinates

As explained in Chapter 4, colors can be represented as vectors in $\mathbb{R}^d$, where $d = 3$ in the case of (R,G,B) coordinates or $d = 2$ in a *chromaticity space*. In

[3] A distance measure can be similarly defined as $\mathcal{D}(\mathcal{H}_1,\mathcal{H}_2) = 1 - \cap(\mathcal{H}_1,\mathcal{H}_2) \in [0,1]$.

our case, to alleviate the problems found in images with specularities or shape information and to reduce at the same time the computational burden, we use the *perspective color* $(r, g) = (R/B, G/B)$ proposed by Finlayson in [Fin96] which discards intensity. Physically, those coordinates consist in normalizing the response of one sensor to be the unit, usually the blue component.

One might imagine that these coordinates are highly sensitive to noise. However, this is not the case in practice since pixel values are quantized. As a further advantage, Finlayson and Hordley proved in [FH00] that the set of feasible mappings computed in a $3D$ space and projected into a $2D$ space afterward is the same as the one directly computed in a $2D$ space.

5.5.4 Color Change Model

The problem of finding a general color change model was extensively studied in Chapter 4. As a result, it was stated that any color change could be described using a homogeneous linear relation after truncating the continuos solution to the problem.

In short, if a vector $\mathbf{r}$ represents the surface reflectance having color $\mathbf{y}_a$ under some light conditions encompassed in matrix $\mathbf{K}_a$ and we want to know how this surface would look like – $\mathbf{y}_c$ – under a canonic illumination $\mathbf{K}_c$, we have that

$$\left.\begin{array}{rcl} \mathbf{y}_a^t & = & \mathbf{K}_a\,\mathbf{r}^t \\ \mathbf{y}_c^t & = & \mathbf{K}_c\,\mathbf{r}^t \end{array}\right\} \Longrightarrow \mathbf{y}_c^t = \left(\mathbf{K}_c\,\mathbf{K}_a^{-1}\right)\,\mathbf{y}_a^t = \mathbf{T}\,\mathbf{y}_a^t \tag{5.23}$$

where $\mathbf{T} \in \mathcal{M}_d(\mathbb{R})$ is a squared matrix coping with the color change, mapping colors from the actual light conditions onto those under the canonic illumination. Thus, the mapping T is specified in coordinates as follows

$$\begin{array}{rcl} \mathrm{T}\colon \mathcal{I}_a & \longrightarrow & \mathcal{I}_c \\ \mathbf{s} & \longmapsto & \mathrm{T}(\mathbf{s}) = \mathbf{T}\,\mathbf{s}^t = \mathbf{q}^t \end{array} \tag{5.24}$$

A reasonable tradeoff between simplicity and performance is attained employing a *diagonal model* [For90, FHH99, FHH01, Fin96], as it is described in Chapter 4. This model assumes that color sensors are completely uncorrelated and any change in the light falling into them equates to independently scaling each channel, that is, $\mathbf{T} = diag(t_1, \ldots, t_d)$. Equivalently, $\mathbf{T}$ can be also expressed as a vector $\mathbf{t} = (t_1, \ldots, t_d) \in \mathbb{R}^d$, so that, $\mathbf{T} = \mathbf{t}\,\mathbf{I}_d$.

This is the color change model we are using throughout the rest of Chapter, though the whole color constancy scheme can also work with more complete color mappings. Nevertheless, as we will see soon, the more complete the model, the more information and time to process it.

5.5.5 Mapping Estimation

Once established the sort of color coordinates and color change model, we need to further define the mapping estimation scheme. The idea is to recover an estimate of the mapping T employing two subsets of colors from both $\mathcal{I}_a$ and $\mathcal{I}_c$.

Formally, the mapping estimator Δ is a function that computes a mapping $\mathrm{T} \in <\mathrm{T}>$ from two sets $\mathcal{S} = \{s_i\}_{i=1,\ldots,n} \subset \mathcal{I}_a$ and $\mathcal{Q} = \{q_i\}_{i=1,\ldots,n} \subset \mathcal{I}_c$, i.e.,

$$\begin{array}{rcl} \Delta\colon 2^{\mathcal{I}_a} \times 2^{\mathcal{I}_c} & \longrightarrow & <\mathrm{T}> \\ (\mathcal{S}, \mathcal{Q}) & \longmapsto & \Delta(\mathcal{S}, \mathcal{Q}) = \mathrm{T} \end{array} \tag{5.25}$$

Before using this function, we need to express it in coordinates. As seen in the previous Section, a mapping T can be seen as a matrix $\mathbf{T}$ so that $q = \mathrm{T}(s)$ is equivalent to $\mathbf{T}\,\mathbf{s}^t = \mathbf{q}^t$. In the most general way, it is not possible to find a matrix $\mathbf{T}$ out of just a pair of corresponding vectors $\mathbf{s}$ and $\mathbf{q}$ since we need, at least, as many pairs as the space dimension d.

Taking advantage of the nature of $\mathbf{T}$ and two sets of $n \geq d$ *one–to–one* corresponding vectors, namely, $\{\mathbf{s}_i\}_{i=1,\ldots,n}$ and $\{\mathbf{q}_i\}_{i=1,\ldots,n}$, it is true that

$$\mathbf{T}\,\left(\mathbf{s}_1^t \mid \cdots \mid \mathbf{s}_n^t\right) = \left(\mathbf{q}_1^t \mid \cdots \mid \mathbf{q}_n^t\right) \tag{5.26}$$

If $\mathbf{S} = (\mathbf{s}_1^t \mid \cdots \mid \mathbf{s}_n^t) \in \mathcal{M}_{d\times n}(\mathbb{R})$ and $\mathbf{Q} = (\mathbf{q}_1^t \mid \cdots \mid \mathbf{q}_n^t) \in \mathcal{M}_{d\times n}(\mathbb{R})$ are introduced by joining vectors columnwise, finding a matrix $\mathbf{T}$ consists in solving the linear system $\mathbf{Q} = \mathbf{T}\,\mathbf{S}$, where $\mathbf{T}$ is the unknown.

The usual method to solve this system is by a least squares approach, that is, minimizing the residual of the solution $\|\mathbf{T}\,\mathbf{S} - \mathbf{Q}\|_2^2$, which can be numerically done by the Moore–Penrose's pseudoinverse $\mathbf{S}^+ = \mathbf{S}^t\,\left(\mathbf{S}\,\mathbf{S}^t\right)^{-1}$ or using the SVD decomposition of matrix $\mathbf{S}$. Therefore, the version of Eq. (5.25) expressed in coordinates is

$$\begin{array}{rcl} \Delta\colon \mathcal{M}_{d\times n}(\mathbb{R}) \times \mathcal{M}_{d\times n}(\mathbb{R}) & \longrightarrow & \mathcal{M}_d(\mathbb{R}) \\ (\mathbf{S}, \mathbf{Q}) & \longmapsto & \Delta(\mathbf{S}, \mathbf{Q}) = \mathbf{Q}\,\mathbf{S}^+ = \mathbf{T} \end{array} \tag{5.27}$$

In case the diagonal model is used, the above function is greatly simplified, using a vectorial notation as follows

$$\begin{array}{rcl} \Delta\colon \mathbb{R}^d \times \mathbb{R}^d & \longrightarrow & \mathbb{R}^d \\ (\mathbf{s}, \mathbf{q}) & \longmapsto & \Delta(\mathbf{s}, \mathbf{q}) = (\frac{q_1}{s_1}, \ldots, \frac{q_d}{s_d})^t = \mathbf{t} \end{array} \tag{5.28}$$

The method for carrying out the mapping computations between two sets of colors consists in first computing the histograms of both sets and applying the estimator Δ to every possible pair of color sets afterwards. An important point of the mapping estimation scheme Δ is that it does not need the colors in $\mathcal{S}$ to be really in a one–to–one correspondence with those in $\mathcal{Q}$. Rather, all possible correspondences between the two sets are checked over, each one giving a feasible mapping as a result. It seems clear, at this point, that computing all possible combinations will not be easy in practice and that some heuristic will be necessary in order to speed up computations.

The first choice is to employ a random selection among colors with the highest frequencies in order to reduce the number of possible combinations to check up on. Other ways to reduce the amount of computations would involve the knowledge of the distribution $Pr\,(\mathcal{Q} \mid \mathcal{S})$ or the selection of colors in $\mathcal{S}$ and $\mathcal{Q}$ which are in similar categories. Nevertheless, we are not going into further questions about speeding up Hough–like schemes or such as stopping criteria.

In our case, only colors corresponding to bins with nonzero frequency were taken into account to reduce the number of combinations. As a result, the amount of computations to produce the histogram of feasible mappings $\mathcal{H}_{\langle \mathrm{I} \rangle}$ was reduced to less than $O\left(m^2\right)$, where $m << n$ is the number of histogram bins and n the number of colors in a set. Thus, the number of computations is far smaller than theoretically, which makes the algorithm fully affordable in practice.

5.5.6 Mapping Selection

Once a set of possible mappings is derived and after computing the value of their likelihood, the best mapping among them must be selected. As said in Eq. (5.17), in principle the mapping $\hat{\mathrm{T}}$ with the highest likelihood is the one chosen as the proper mapping. However, other selection methods can also be applied to improve the resulting mapping. Next is a succinct description of them.

Max As proposed in Eq. (5.17), the mapping with the highest likelihood is selected.

Mean The selected mapping comes from the mean among the candidates after weighting each one by its likelihood value. Due to the discretization of histograms, it might be helpful to use this scheme instead to aim better when selecting the rightest mapping.

Constrained Max The same as **Max** but the set of feasible mappings $<\mathrm{T}>$ has been restricted using the set of feasible illuminants $\Gamma^{-1}(\mathcal{E}/\mathbf{c}^{\mathcal{C}})$, as proposed by Finlayson. To avoid the problem of nonconvexity of this set, it is approximated by its convex closure, which reduces the complexity of intersections.

Constrained Mean The same as **Mean** after using the illuminant restriction explained in the previous paragraph.

Center of Mass This consists in computing the center of mass of the convex hull computed from all the values in the set of feasible mappings $<\mathrm{T}>$.

5.6 Experiments and Results

The goal in this Section has to do with the comparison of performances between our color constancy algorithm and Finlayson's gamut–mapping. First, we illustrate a bit the kind of images used to produce those results as well as the sort of experiments carried out, which basically consists in computing a distance between the appearance of images before and after the color constancy stage modifies the colors of the images. At the end of the Section, we summarize the whole set of results obtained and some conclusions about them are taken into account.

5.6.1 Image Database Description

The database[4] considered here is partially shown in Fig. 5.1. It consists of a set of 220 images from 20 objects viewed under the same 11 illuminants considered as the canonic lights in Chapter 4. The methodology for the capture of data and further details about the database are described in [BMFC02]. An important point of this database is that the authors also took care of spectrally analyzing the kind of lights that were utilized to produce these images. Thus, the spectra of illuminants and their color are also available at the same site, which helps to build up the set of feasible illuminants.

[4] `http://www.cs.sfu.ca/~color/data/objects_under_different_lights/index.html`

Basically, these images picture single colorful objects over a black background to avoid interreflections from the environment. The pose of the objects varied every time illuminants were changed, which makes images slightly different. Some of the objects are exhibited in Fig. 5.2 as an example of how their appearance changes under the 11 lights employed. This database has also been used in a number of publications about color constancy such as [Bar99b, BFM00].

5.6.2 Experiment Description

The aim of this Chapter is to propose an alternative *color constancy* (CC) algorithm useful in case little information about lights in the scene is available. It tries to map the set of colors in a certain image onto those one could perceived under some canonic conditions. To do that, we employ a *canonic image* of the scene as the reference to which to map this set. That image provides the set of canonic colors. In order to compare the results obtained by our approach the same experiments have also been carried out by The Finlayson's 2D–GM algorithm since it is the most similar one to ours among the different CC approaches that exist.

The basic idea of the experiment consists in comparing colors of each scene with those of the canonic image before and after the CC step is applied. This way, we can measure the differences between these two cases and then compute the amount of reduction in color dissimilarity among all the images belonging to the same scene. This process is done for the two algorithms using one canonic image for each set of images. The greater the reduction of color dissimilarity, the more colors in the processed images resemble to those in the canonic one.

For each algorithm, various heuristics were tried with small changes in the way the final map was extracted from the feasible set of mappings. In our algorithm we used CM – center of mass –, $Mean$, Max, and the constrained variation of the latter two, namely, $CMean$ and $CMax$. As explained in Section 5.5.6, CM computes the final mapping as the center of mass of the convex hull extracted from the set of feasible mappings. $Mean$ calculates the weighted mean of the feasible mappings, while Max searches for the mapping with the highest likelihood. The constrained versions of $Mean$ and Max further reduce the set of feasible mappings by applying to this set the Finlayson's constraint on illuminants. This is endeavored whenever the set of feasible lights is available.

In the case of Finlayson's 2D–GM we also employed the computations of the center of mass ($FinCM$), the mean ($FinMean$), and an alternative version ($FinHord$) of the 3D mean proposed by Hordley and Finlayson in [FH99, FH00]. In all these cases the measure of likelihood of the mappings was not used, since neither was this information taken into account by the original gamut–mapping algorithm.

We used two measures of color dissimilarity to estimate the performance of the two CC methods. First, we made use of the difference between the color histograms from the canonic images and the processed images, respectively. These measures are useful to gather how close those histograms get after the CC stage. The intersection–based distance proposed by Swain&Ballard [SB90] was employed as the measure between color histograms.

Whenever this type of similarity is applied, we can also compute a further result corresponding to the use of true mappings which is computed directly

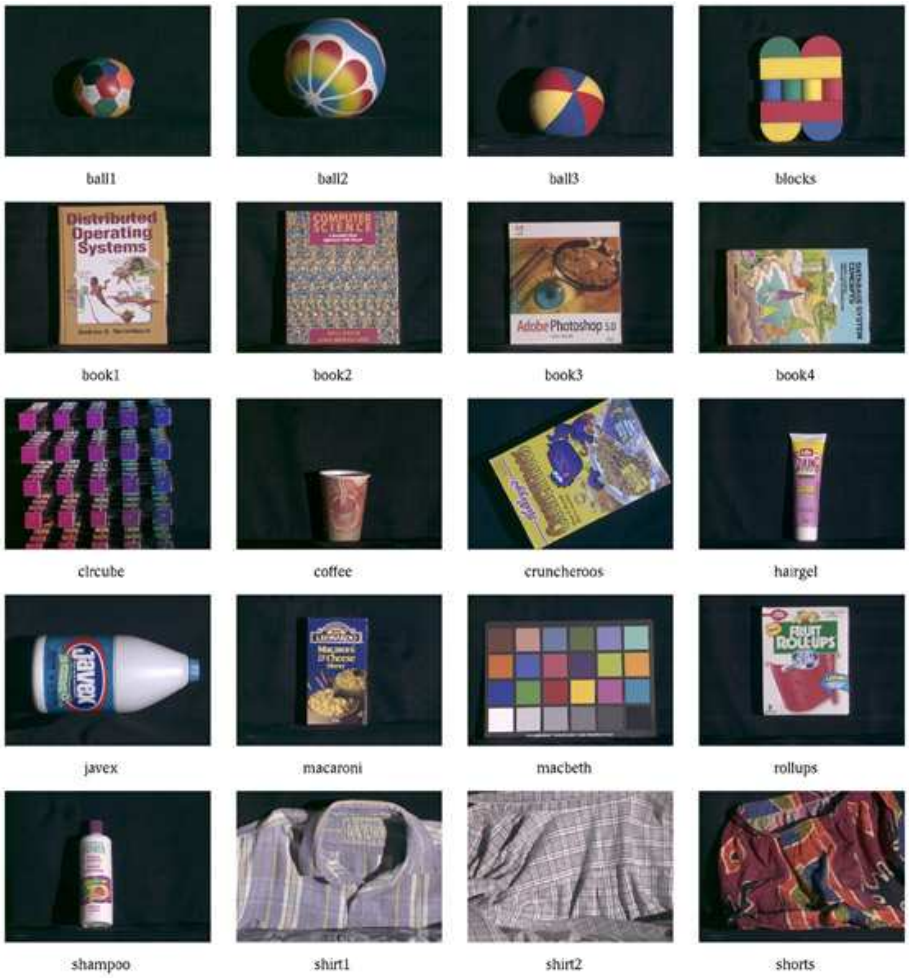


Figure 5.1: Images in Simon Fraser University database.

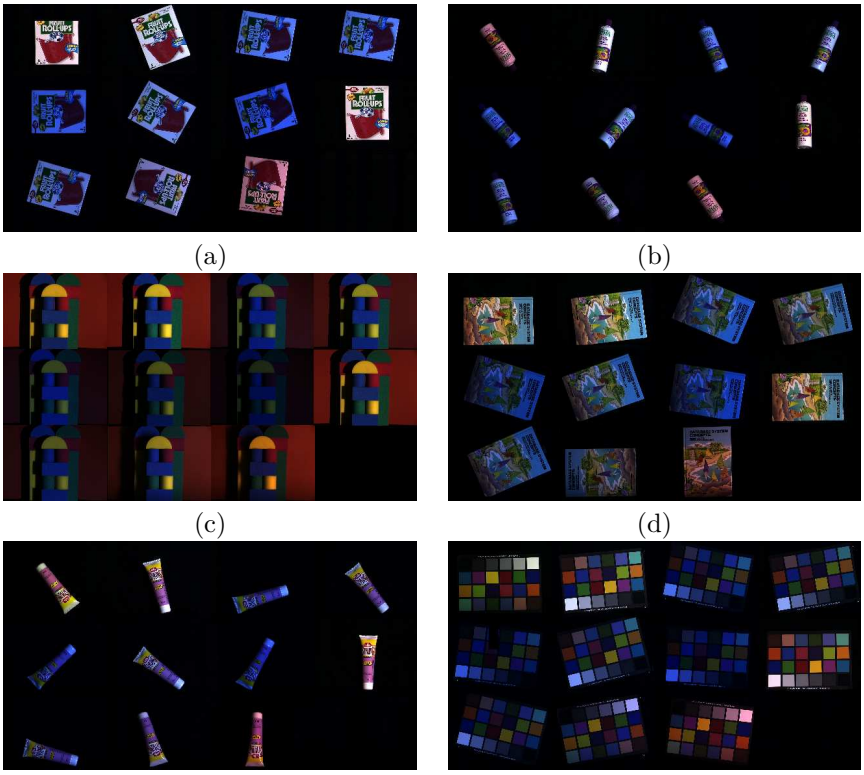


Figure 5.2: Sets of some objects under all light variation: (a) rollups, (b) shampoo, (c) blocks, (d) book4, (e) hairgel, and (f) macbeth.

from the values of the true illuminants kindly provided by the authors of the database. Thus, we can compare what we would get in case of having this information with the one that our CC step really computes. It is kind of a *limit* for the performance of the color change model. Those results are specifically labeled as $True$.

Secondly, we also used the RMSE measure of error, which is computed pixelwise as a distance between images. Since objects in the images were appreciably moved from one illuminant to the other, applying this distance between the canonic and the processed image would be totally erroneous in our opinion since part of the distance would be only on account of the different position, rather than the different color between images.

As a consequence of this lack we took a different path and applied two mappings to each image, the one corresponding to the true illuminants and the other extracted from the CC algorithm. If the CC process has been perfect, the two maps will be exactly the same and so will the two transformed images, as a consequence.

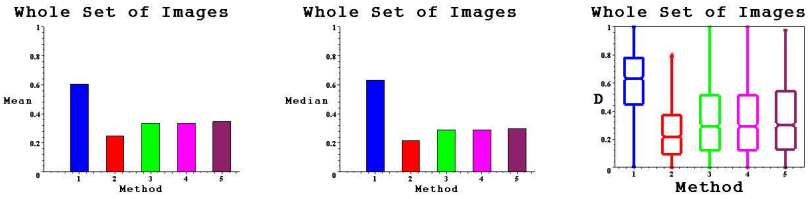
The results of these experiments are summarized in Fig. 5.3 through Fig. 5.6. Plots in Fig. 5.3 and Fig. 5.4, respectively, show the results corresponding to Finlayson's 2D–GM and to our algorithm, where the color difference has been computed with the Swain&Ballard distance. On the other hand, Fig. 5.5 and Fig. 5.6 illustrate the results for the two algorithms using this time the RMSE distance instead.

These lot of graphics consist of three plots each, being first the one accounting for the mean distances globally computed for all the images in the database. The second plot corresponds to the median, while the third depicts the distribution of distances by means of a boxplot[5]. Each color corresponds to a version of the algorithm as it is indicated in the caption of those Figures.

In addition to the graphics described above, the numeric values corresponding to those Figures are also summarized in a group of Tables. Namely, Table 5.1, Table 5.2, and Table 5.3 correspond to the results of the Swain&Ballard distance, while Table 5.4, Table 5.5, and Table 5.6 belong to the RMSE distance. Additional information provided by these tables encompasses standard deviation as well as percent reduction in the mean distance, which is located at the lower row. On the other hand, columns account for the results obtained for each version of the two algorithms taken into consideration.

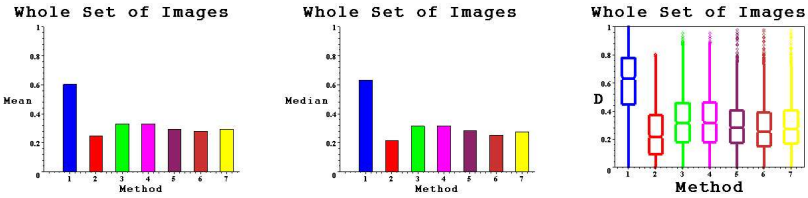
Finally, in Fig. 5.7 through Fig. 5.10 some further examples of images obtained after the color constancy phase are also given. In these groups of images, the upper row represents an object under 4 different lights. The first image at the lower row is the canonic image of the set, whereas the rest of the row corresponds to the transformed images obtained from those in the upper row. In Fig. 5.7 and Fig. 5.9 objects were not moved and only the light was changed, while in Fig. 5.8 and Fig. 5.10 both position and lights were altered.

[5] A boxplot is a statistic tool to describe, at first glance, a distribution of values. The box is dimensioned by the 25% and 75% quartiles, whereas the notch corresponds to the median (50% quartile). The tails span from the minimum to the maximum values. Points correspond to outliers if the distribution were Gaussian.



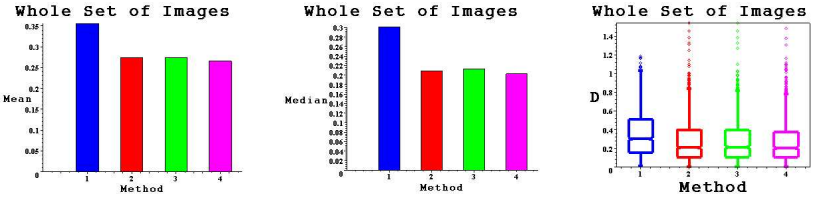
(a) (b) (c)

Figure 5.3: Results using the S&B distance and the Finlayson's 2D–GM algorithm. Blue: *None*. Red: *True*. Green: *FinCM*. Magenta: *FinMean*. Violet: *FinHord*. (a) Mean, (b) median, and (c) boxplot.



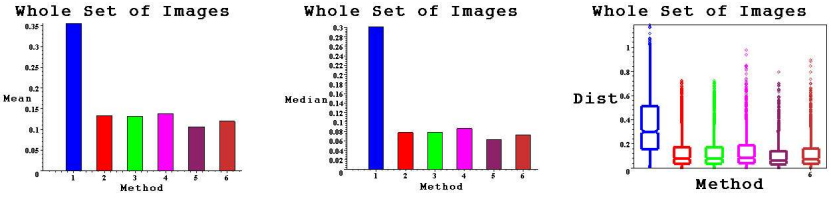
(a) (b) (c)

Figure 5.4: Results using the S&B distance and our algorithm. Blue: *None*. Red: *True*. Green: *CM*. Magenta: *Mean*. Violet: *Max*. Orange: *CMean*. Yellow: *CMax*. (a) Mean, (b) median, and (c) boxplot.



(a) (b) (c)

Figure 5.5: Results using the RMSE distance and the Finlayson's 2D–GM algorithm. Blue: *None*. Red: *FinCM*. Green: *FinMean*. Magenta: *FinHord*. (a) Mean, (b) median, and (c) boxplot.



(a) (b) (c)

Figure 5.6: Results using the RMSE distance and our algorithm. Blue: *None*. Red: *CM*. Green: *Mean*. Magenta: *Max*. Violet: *CMean*. Orange: *CMax*. (a) Mean, (b) median, and (c) boxplot.

	None	True
Mean	**0.6036**	**0.2488**
StD	0.2300	0.1788
Q25%	0.4465	0.0945
Q50%	0.6326	0.2172
Q75%	0.7801	0.3719
% Red.		**58.78**

Table 5.1: Results using the S&B distance, original images, and true illuminants.

	FinCM	FinMean	FinHord
Mean	**0.3367**	**0.3357**	**0.3464**
StD	0.2524	0.2513	0.2547
Q25%	0.1214	0.1223	0.1278
Q50%	0.2908	0.2907	0.2998
Q75%	0.5121	0.5115	0.5423
% Red.	**44.22**	**44.38**	**42.61**

Table 5.2: Results using the S&B distance and the Finalyson's 2D–GM algorithm.

	CM	Mean	Max	CMean	CMax
Mean	**0.3301**	**0.3301**	**0.2956**	**0.2805**	**0.2945**
StD	0.1945	0.1939	0.1721	0.1796	0.1782
Q25%	0.1791	0.1804	0.1735	0.1491	0.1687
Q50%	0.3151	0.3174	0.2822	0.2742	0.2110
Q75%	0.4589	0.4613	0.4029	0.3884	0.4038
% Red.	**45.31**	**45.31**	**51.03**	**53.53**	**51.21**

Table 5.3: Results using the S&B distance and our algorithm.

	None
Mean	**0.3559**
StD	0.3001
Q25%	0.2450
Q50%	0.1554
Q75%	0.5119

Table 5.4: Results using the RMSE distance and original images.

	FinCM	FinMean	FinHord
Mean	**0.2732**	**0.2738**	**0.2655**
StD	0.2185	0.2136	0.2121
Q25%	0.1053	0.1082	0.1082
Q50%	0.2081	0.2124	0.2028
Q75%	0.4001	0.3947	0.3753
% Red.	**23.24**	**23.07**	**25.40**

Table 5.5: Results using the RMSE distance and the Finalyson's 2D–GM algorithm.

	CM	Mean	Max	CMean	CMax
Mean	**0.1330**	**0.1313**	**0.1382**	**0.1060**	**0.1200**
StD	0.1448	0.1408	0.1416	0.1131	0.1262
Q25%	0.0353	0.0355	0.0407	0.0303	0.0355
Q50%	0.0775	0.0779	0.0859	0.0625	0.0722
Q75%	0.1702	0.1722	0.1886	0.1411	0.1612
% Red.	**62.63**	**63.11**	**61.17**	**70.32**	**66.28**

Table 5.6: Results using the RMSE distance and our algorithm.

5.6.3 Some Considerations about the Results

Let us have a look upon the results obtained in this Chapter. First of all, we should state that things work fine since the results we attained for the Finlayson's 2D–GM – $RMSE \approx 0.27$ – are fully comparable to those reported in [FHH99, FHH01] for the same method and type of similarity measure – $RMSE \approx 0.21$ –, despite the database is not the same. After this verification, some more things can be drawn for the figures.

In the group of results evaluated by way of the Swain&Ballard distance in Table 5.2 and Table 5.3, the Finlayson's 2D–GM best result is achieved using the $FinMean$ heuristic to select the proper mapping, while in ours, the best result is obtained using the $CMean$ method. It can be seen that the constrained selection provides slightly better results than the unconstrained one, since the feasible set is reduced in an appropriate way after applying the set of feasible lights. Despite this good performance, it is not always easy to collect all the needed set of illuminants in poorly known environments.

When comparing the best results achieved by the two algorithms considered in this Chapter, it must be noticed that our percent reduction – 53.53% – is slightly higher than that of Finlayson's – 44.38% –, and is pretty close to the best results attainable by the diagonal model of color change using the true values of illuminants, i.e., 58.78% of reduction.

The RMSE group of results in Table 5.5 and Table 5.6 show similar conclusions. The values obtained by our algorithm are definitely better than those of Finlayson's 2D–GM and are totally similar to the best color constancy algorithm, namely, color–by–correlation [FHH99, FHH01] – $RMSE \approx 0.11$ –, which is a fairly good result since our algorithm works with less or even none information about light unlike the Finlayson's approach, depending on the selection heuristic. Once more, constrained selections give better results than unconstrained ones.

It is also interesting when using the RMSE distance to note that the performance as a function of the selection scheme is not exactly the same as if the Swain&Ballard histogram distance were employed. For instance, in the RMSE case, $Mean$ provided better marks than Max in our algorithm. Something alike happens with $FinHord$ and $FinMean$ in the Finlayson's algorithm. Whilst $FinMean$ is the best and $FinHord$ the worst in the RMSE case, it is just the other way in the Swain&Ballard case.

Reasons for such a behavior may be multiple. On the one hand, the RMSE distance is applied between two images coming from just one image, which has undergone two different color changes, while in the Swain&Ballard case the distance is computed between the histograms of the canonic and the transformed images, i.e., only one color transformation has been applied and, moreover, the canonic image is not synthesized but real. On the other hand, distances between histograms may depend on the discretization of the color space. Due to the fact that the two distances seem to indicate the same trend for the results obtained so far, we accept both as equally helpful measures, despite our belief that a distance in the histogram space appreciates best the color dissimilarity between images.

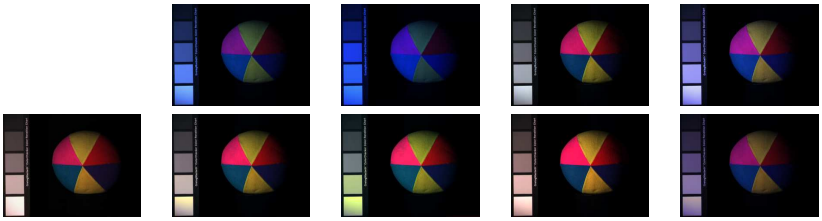


Figure 5.7: Color constancy results: Ball set with light variation.

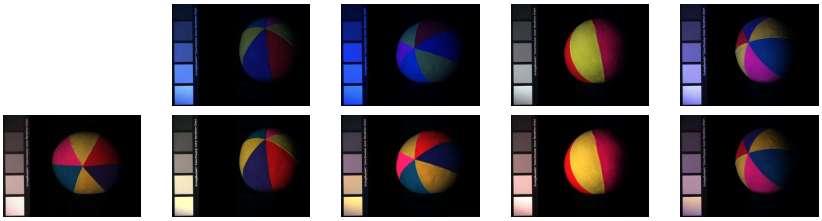


Figure 5.8: Color constancy results: Ball set with pose and light variation.

Figure 5.9: Color constancy results: Tide set with light variation.

Figure 5.10: Color constancy results: Tide set with pose and light variation.

5.7 Conclusions

This Chapter is devoted to the description of our color constancy algorithm which is able to render the image colors back to those that would been seen under the conditions of a canonic light. This algorithm is based on the computation of the histogram for all feasible maps arising between two sets of colors, namely, the one belonging to the actual image and the canonic set. In order to have a tool to select the most suitable mapping from all feasible transformations, a measure of likelihood for each of them is suggested. The likelihood function depends both on the number of times a particular transform is recovered from these sets of colors and its effectiveness in the task of recovering the most similar colors to the canonic set. This is done by computing a measure of similarity between transformed and canonic histograms.

A set of experiments using a pretty nourished number of images of objects under different illuminations were undergone in order to evaluate the performance of the proposed algorithm. These experiments consist in measuring the similarity there exists between the images in the set and those taken as the canonic ones in two specific situations, i.e., before and after the color constancy stage is carried out on them. Two types of similarity measures are employed to compute the difference between images. Analogous results are also obtained from the most similar color constancy method we found, namely, Finlayson's 2D gamut–mapping, and are compared to those obtained by our algorithm, which outperformed Finlayson's as well as attaining similar results to the color–by–correlation approach, the best color constancy algorithm to our knowledge.

The advantage of our algorithm with respect to the others is that ours is able to work with the information provided by only a canonic image as the reference to colors, rather than using an *a priori* correlation matrix that must span for any color and for any light. Besides, it is not restricted just to a discrete group of illuminants, but rather to a piece of a continuous set. Constraints on the set of feasible lights can easily be incorporated to the selection heuristic, improving the performance of the algorithm as a consequence. The behaviour of our color constancy algorithm is similar to the best known color constancy algorithms but, in contrast, it requires less information and is thus more adequate in tasks where there exists little control and knowledge on light sources, such as mobile robotics.