Color Constancy using 3D Scene Geometry derived from a Single Image

Noha M. Elfiky, Theo Gevers, Arjan Gijsenij and Jordi Gonzàlez

***Abstract*—The aim of color constancy is to remove the effect of the color of the light source. As color constancy is inherently an ill-posed problem, most of the existing color constancy algorithms are based on specific imaging assumptions (e.g. grey-world and white patch assumption).**

In this paper, 3D geometry models are used to determine which color constancy method to use for the different geometrical regions (depth/layer) found in images. The aim is to classify images into stages (rough 3D geometry models). According to stage models; images are divided into stage regions using hard and soft segmentation. After that, the best color constancy methods is selected for each geometry depth. To this end, we propose a method to combine color constancy algorithms by investigating the relation between *depth*, local *image statistics* and *color constancy*. Image statistics are then exploited per depth to select the proper color constancy method. Our approach opens the possibility to estimate multiple illuminations by distinguishing nearby light source from distant illuminations.

Experiments on state-of-the-art data sets show that the proposed algorithm outperforms state-of-the-art single color constancy algorithms with an improvement of almost 50% of median angular error. When using a perfect classifier (i.e, all of the test images are correctly classified into stages); the performance of the proposed method achieves an improvement of 52% of the median angular error compared to the best-performing single color constancy algorithm.

I. Introduction

The color of objects is largely dependent on the color of the light source. Therefore, the same object recorded by the same camera but under different illumination conditions may vary in its measured color appearance. This color variation may negatively affect the result of subsequent image and video processing methods for different applications such as object recognition, tracking and surveillance. The aim of color constancy is to remove the effect of the color of the light source. A considerable number of color constancy algorithms have been proposed, see [1], [2], [3] for reviews. Traditionally, pixel values are exploited to estimate the illumination. Examples of such methods include approaches based on low-level features [4], gamut-based algorithms [5], and methods using learning [3]. Recently, methods that use derivatives (*i.e.,* edges) and even higher-order statistics have been proposed [6].

In general, color constancy algorithms are based on specific assumptions about the illumination or properties of the object reflectance. As a consequence, none of them can be considered as universal. Therefore, different methods have been proposed to select or combine color constancy approaches. Higher level visual information is considered only recently [6], [7], [8]. In [6], the image is modeled as a mixture of semantic classes, such as sky, grass, roads and buildings. Illumination estimation is steered by different classes by evaluating the likelihood of the semantic content. Similarly, indoor-outdoor image information is used [7]. Alternatively, image statistics are used in [8] to select the most useful color constancy method. It is shown that images with similar image statistics will benefit from the same color constancy algorithm.

Recently, the work in [9] and [10] shows the importance of using stage models for selecting the proper color constancy algorithm per stage. Stage models are 3D geometry models of a scene. The method in [10] shows the importance of exploiting local depth information for each stage geometry over using the global geometry [9]. Alternatively, Jose et al. [11] proposed a different approach to address multi-illuminations. The method uses color and texture features to compute the nearest neighbor regions from the training data. This is done for each region in the test image. The illumination estimation for each (test) region is based on histogram matching using the nearest (training) region. Finally, the region estimates are combined in a single estimated illumination for the entire image.

In contrast to previous work, our contribution is to exploit the relation between *depth*, *image statistics* and *color constancy*. It has been shown that image statistics are influenced by depth patterns [12], e.g. the signal-to-noise ratio generally decreases as the depth increases [13] while the scale changes when viewing scenes from different depths [14]. Furthermore, attributes like signal-to-noise and scale are inherently correlated with color constancy [8].

Therefore, in this paper, the relationships between depth, local image statistics and color constancy algorithms are investigated. The aim is to compute the $3D$ scene geometry from a single image. In this way, the depth layers are obtained. Then, image statistics are exploited (per layer/depth) to select the color constancy method with best expected performance. Color constancy will be applied per depth layer allowing multiple light sources per scene.

The outline of the paper is as follows: we give the motivation of our approach in section II, then in section III we briefly outline the color constancy framework and $3D$ scene geometry classification. Section IV presents the datasets and error metrics. In section V and VI, we outline the proposed. Section VII presents the experimental results on standard datasets. Finally, section VIII concludes the paper.

II. Motivation

The lighting conditions in a scene may differ at varying depth layers. For example, nearby objects may be illuminated by a different light source than distant ones. If we can derive the scene geometry, this will result in important depth cues. In this section, we motivate our approach by inferring a relation between the depth of a scene and color constancy.

Image statistics and scene depth: The relation between depth patterns and natural image statistics is studied in [14], [15]. They show that in case of a dominant structure (object or background) gradient histograms correspond to a decaying power-law distribution. When depth increases, the size of objects will decrease showing less texture. Hence, object distance can be associated by a power-law.

In this paper, to model natural image statistics, the integrated Weibull distribution is taken as the representation of a decaying power-law [16]:

$$\omega(x) = C \exp(-\frac{1}{\gamma}|\frac{x}{\beta}|^{\gamma}) \quad (1)$$

where x is the edge response in a single color channel of a Gaussian derivative filter, C is a normalization constant, β is the scale parameter (i.e. the width) of the distribution and γ is the shape parameter (i.e. the peakedness) of the distribution. The parameters of this distribution are indicative for the edge statistics of a scene [16]. In general, the Weibull parameter β encodes image (edge) contrast, and γ corresponds to the grain size and is related to the amount of texture. A higher value for β indicates more contrast, while a higher value for γ indicates a smaller grain size (more fine texture). For example, $\gamma = 2$ the Weibull distribution is equivalent to the normal distribution and for $\gamma = 1$ it is a double exponential. The exponent δ indicates the fractal dimension.

When the depth of a natural scene becomes larger, object surfaces will become smaller and will contain fewer details and therefore become smoother. Hence, scene elements appear increasingly fuzzier with depth. On the contrary, when scene depth becomes smaller, object surfaces will become larger and coarser showing more contrasting details. In this case, natural image statistics computed from the image follow a Weibull distribution with increasing β and γ [15], which is consistent with the observations in [16]. Hence, a relation exists between natural image statistics and depth patterns of a scene [14], [15].

Image statistics and color constancy: It has been shown in [17], that natural image statistics, represented by Weibull distributions, are useful to select the proper color constancy method. It is derived that if the image contains a limited number of edges, pixel-based color constancy is preferred. In case of sufficient edges (e.g., more than eight different edges), edge-based color constancy is preferred. The order of the best performance in terms of the number of edges (from low to high) is as follows: zeroth-order statistics first, followed by first-order and second-order statistics. If the image contains contrasted edges (i.e. β and γ are small), edge-based color constancy is used, otherwise pixel-based color constancy is preferred. Hence, a relation exists between image statistics (Weibull parameterization) and color constancy where contrast β and grain size (γ) are related to the number of edges, amount of texture, and signal-to-noise ratio to which the used color constancy methods are sensitive.

Depth and color constancy: In contrast to previous work, the contribution of this paper is to exploit the relation between depth and color constancy. Contrasted details (edges) are common for close-by objects (edge-based color constancy is preferred) than distant objects (pixel-based color constancy is preferred). This novelty is used to improve color constancy by inferring the scene geometry, and consequently the depth of each layer. Then, image statistics are exploited (per layer/depth) to select the proper color constancy method. Further, scene layers at different distances may be illuminated by different light sources. Color constancy will be applied per layer allowing multiple light sources per scene. The only assumption is that a single light source is illuminating the scene part at a certain distance (layer).

III. Preliminaries

In this section, we briefly discuss the computational methods to estimate the illumination and to determine scene geometries (stages).

A. Color Constancy

Assuming Lambertian reflection, the image color $\mathbf{f} = (R, G, B)^T$ depends on the color of the light source $e(\lambda)$, the surface reflection $s(\mathbf{x}, \lambda)$ and the camera sensitivity function $\mathbf{c}(\lambda)$:

$$\mathbf{f}(\mathbf{x}) = \int_{\omega} e(\lambda)\mathbf{c}(\lambda)s(\mathbf{x}, \lambda)d\lambda \ , \quad (2)$$

where ω is the visible spectrum, λ is the wavelength of the light and $\mathbf{x}$ is the spatial coordinate. Under the assumption that the recorded color of the light source $\mathbf{e}$ depends on the color of the light source $e(\lambda)$ and the camera sensitivity function $\mathbf{c}(\lambda)$, the color of the light source is estimated by

$$\mathbf{e} = \int_{\omega} e(\lambda)\, \mathbf{c}(\lambda)\, d\lambda \ . \quad (3)$$

Since both $e(\lambda)$ and $\mathbf{c}(\lambda)$ are unknown, color constancy is an under-constrained problem. Therefore, in order to solve the color constancy problem, a number of assumptions are made such as the Grey-World assumption (i.e. the average pixel value is grey) and the White-Patch assumption (i.e. the maximum pixel value is white) [1].

To incorporate both pixel values and higher-order derivative information, in this paper, the following color constancy framework is used [6],

$$\left(\int \left| \frac{\partial^n \mathbf{f}^{\sigma}(\mathbf{x})}{\partial \mathbf{x}^n} \right|^p d\mathbf{x} \right)^{\frac{1}{p}} = k\, \mathbf{e}^{n,p,\sigma}, \quad (4)$$

where k is a multiplicative constant chosen such that the illuminant color, $\mathbf{e} = (e_R, e_G, e_B)^T$, has unit length (the Euclidean norm of a vector is used), n is the order of the derivative, p is the Minkowski-norm and $\mathbf{f}^\sigma(\mathbf{x}) = \mathbf{f} \otimes G^\sigma$ is the convolution of the image with a Gaussian filter with scale parameter σ. Using Equation 4, different color constancy algorithms are generated by varying one or more of parameter values. For example,

1) when n=0, pixel-based color constancy algorithms are obtained, such as the Grey-World algorithm ($\mathbf{e}^{0,1,0}$), the White-Patch algorithm ($\mathbf{e}^{0,-1,0}$) and the general Grey-World ($\mathbf{e}^{0,13,2}$);
2) when n=1, color constancy algorithms are obtained using first-order derivative information, i.e. image edges information. The Minkowski-norm p and smoothing parameter σ depend on each specific data set. The instantiation $\mathbf{e}^{1,1,6}$ is applied in this paper;
3) when n=2, the framework provides color constancy methods based on second-order statistics. Similarly, the other two parameters p and σ vary with the data set. We use $\mathbf{e}^{2,1,5}$ in our experiments.

To this end, in order to exploit the relation between natural image statistics and color constancy, a set of color constancy methods is required [17] for learning the most appropriate method to use within each stage based on its natural image statistics. In this paper, we focus on the above instantiations which include pixel and derivative-based methods. However, other color constancy methods can be used as well.

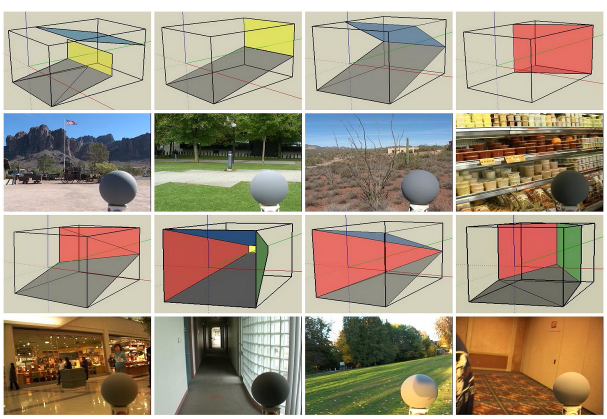


Fig. 1. Stage models [15] and their corresponding instantiations: top two rows, from left to right: sky+bkg+gnd, bkgGnd,skyGnd, gndDiagBkgLR; bottom two rows:diagBkgLR, box, 1side-wallLR, corner.

B. Depth from Stage Models

A number of methods have been proposed to estimate the rough scene geometry from single images [18], [19], [20]. However, these methods are restricted to a number of classes limiting their applicability. Therefore, *stages* [15] which correspond to generic categories, are taken. Stages are defined as a set of prototypes of common scene configurations. They can be seen as discrete classes of scene geometries. Typical classes of discrete $3D$ scene geometries (i.e. stage models) include single-side backgrounds (e.g. walls and buildings) or three sides (e.g. corridor and narrow streets). A number of stage models, together with corresponding images are shown in Figure 1. It has been shown in [15] that images can be classified into one of the different stages. Each stage model has a certain depth layout in terms of layers at a certain distance to the camera. In this way, color constancy can be applied per depth layer.

As shown in Figure 1, the depth structures of the stage models are shown in different colors. Each stage has a unique depth pattern. For instance, images of stage *"sky+bkg+gnd"* are divided in three layers: sky (in blue), background (in yellow) and ground (in brown). While, images of stage *"box"* will be divided in five layers: top (in blue), bottom (in brown), right (in green), left (in red), and middle (in yellow). On the other hand, images of stage *ground* will not be divided as it only contains one depth layer.

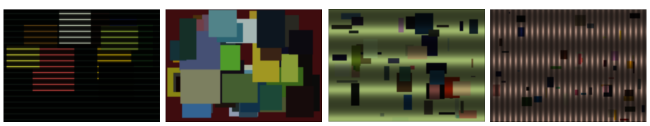
IV. Datasets and Similarity Metrics

Four independent data sets are used in the experiments. In order to obtain a classifier with proper generalization, an independent data set (denoted as *"stages data set"*) is used to train the stage classifiers [15]. The data set consists of 3589 images classified as 15 different categories representing the scene geometries (stages): 151 *sky+bkg+gnd*, 333 *bkgGnd*, 81 *skyGnd*, 212 *ground(gnd)*, 139 *gndDiagBkgLR*, 132 *gndDiagBkgRL*, 75 *diagBkgLR*, 71 *diagBkgRL*, 84 *box*, 57 *1side-wallLR*, 69 *1side-wallRL*, 266 *corner*, 960 *Person (persBkg)*, 833 *noDepth*, and 126 *tabPersBkg*. Images are taken under a large variety of lighting conditions.

As a second data set, the color constancy data set of Ciurea and Funt [21], referred to as *real-world data set*, is used for testing. Note that this data set contains the ground truth for color constancy while no ground truth is provided for *stages data set*. Therefore, the color constancy algorithms are evaluated on *real-world data set*. The *real-world data set* consists of more than 11,000 images, extracted from 2 hours of video for a wide variety of settings such as indoor, outdoor, desert,cityscape, etc. There are in total 15 different video clips taken at different places and with varying lighting conditions. As there exists a correlation among images of the same video clip, we test the color constancy algorithms on a subset of uncorrelated images composed of 711 images. These images are manually selected and annotated. A few example images are shown in figure 2(b). In each image, there is a grey ball at the right bottom, which is used to capture the ground truth of the light source. Note that the grey ball is masked when the illumination is estimated. In order to learn our models for this dataset, we use 15 folds cross-validation. The ground truth of the *real-world* dataset is obtained using the original images (color model is NTSC-RGB). Therefore, the ground truth is recomputed by [22] for converting the images from *NTSC-RGB* to linear *RGB* assuming gamma is equal to 2.2. This modified dataset is named linear *real-world*.

The third set is the *ColorChecker* dataset provided by Shi and Funt [23]. This dataset consists of 568 images, both indoor and outdoor. For this dataset, we used three-fold cross-validation to learn our models.

As a fourth set, the training data set that is created based on spectral reflectance data presented in [24]. This data set originally comprises only surface and illuminant spectra, which are first combined into (R, G, B)-values. Then, using these generated pixel colors, several Mondrian-like images are created, which all have different properties in the number of edges, the amount of texture and contrast. Since only material surfaces are present in the original data set, shadow gradients are added to several images to enlarge their photo-metrical variety. Note that the resulting images contain up to tens of different surfaces, hence, many different transitions, simulating the statistics of real-world images as close as possible. A few example images are shown in figure 2(a). This data set will be called the *Mondrian data set* in the remainder of the paper.



(a) Mondrian images.

(b) Real-world images.

Fig. 2. Examples of images that are in the two data sets that are used in this paper. The first data set consists of images that are generated using surface reflectance spectra combined with illuminant spectra [24]. The second data set consists of real-world images [21].

A. Metrics

Two performance measures are used in this paper: stage classification is evaluated using the average precision, while the angular error is used to validate the performance of the color constancy algorithms.

Average precision *(AP)*. The average precision is equivalent to the area under a precision-recall curve. It combines precision and recall in a single number. Mean average precision *(MAP)* is used to evaluate the performance of the features over all the stages, which is obtained by averaging the average precisions over all stages.

Angular error. In order to evaluate the performance of the color constancy algorithms, the angular error ε is used,

$$\varepsilon = \cos^{-1}(\hat{e_l}.\hat{e_e}), \tag{5}$$

where $\hat{e_l}$ is the normalized ground truth of the illumination, while $\hat{e_e}$ is the normalized estimation. Both mean and median angular errors are taken as performance indicator.

Name	**% in data set**	**AP**
sky+bkg+gnd	9.1%	0.65
bkgGnd	9.9%	0.34
skyGnd	2.7%	0.34
gnd	12.1%	0.67
gndDiagBkgLR	6.6%	0.16
gndDiagBkgRL	4.6%	0.16
diagBkgLR	4.6%	0.12
diagBkgRL	3.8%	0.15
box	8.0%	0.37
1side-wallLR	12.9%	0.46
1side-wallRL	15.6%	0.41
corner	6.5%	0.15
persBkg	3.5%	0.19
MAP	0.320	

TABLE I

Stage classification results for each stage using the RGB-SIFT feature. The last row gives mean average precision over all stages. stages data set is used for training the stage classifiers. real world data set used for testing the color constancy algorithms, see [10] for details.

V. Stage Classifier & Segmentation Masks

To build the stage classifier, we use the Bag-of-Words *(BoW)* classification framework proposed by Van de Sande et al. [25]. The RGB SIFT descriptors are used as it outperforms other variants of the SIFT-feature [26]. For the vocabulary construction, we use a standard k-means to build a vocabulary of size 4000. After the assignment stage, we use a two level Spatial Pyramid (SP) [27]. A compact spatial pyramid is obtained by compressing the original SP histograms based on the method proposed in [28]. For the classification stage, we use generic $1-vs-all-$-based classifiers with χ^2 kernel, see [29], [25]. The output of the classifier is a single stage label. In this paper, there are a total of 13 classifiers corresponding to the 13 stages used (excluding *noDepth* and *tabPersBkg*) as in [10]. These stages are specific characteristics of the data set used in [15].

The performance of the stage classification for each stage is shown in Table I. From this table, it can be derived that for some stages, such as *sky+bkg+gnd*, and *gnd*, the results are satisfying. For other stages, like *diagBkgLR* and *diagBkgRL*, the results still leave room for improvement. This is due to occlusion occuring in these categories, making it hard to classify them correctly.

A. Illumination Estimation from Stages

After stage classification, the proper color constancy algorithm is selected per stage. In [9], algorithm selection is based on the angular error of the five different color constancy algorithms discussed in section III. Algorithms are applied on the training images of a specific stage. The algorithm with the lowest angular error is assigned to the stage under consideration. Note that training the stage classifier and the selection of the proper color constancy algorithm per stage is processed off-line. Then, the on-line processing is to predict which stage an (unknown) image belongs to by using the trained classifier. Finally, the color constancy algorithm that has been assigned to that stage will be used to estimate the light source to correct the image. This method implies *global illumination*, as no depth knowledge is exploited here, see [9] for details.

B. Image Segmentation to Obtain Depth Layers

Different depth layers (image segments) of a stage geometry model correspond to a scene part at a certain depth. Each layer represents geometrical entities like walls, ground, and sky. We will use the image division provided by the stage model to learn which color constancy algorithm performs best for each layer. Both hard and soft segmentation is considered, the latter taking the uncertainty into account due to the rough outline of the stage geometry. Both of them are based on the occurrence probability in the training set. Ground truth is obtained by manual annotation, thereby dividing the training set according to the stage patterns, and fitting the parameters of each stage model (horizon, vanishing points) to visually best fit the underlying data. For this purpose, the *stages* dataset is used to obtain the hard and soft segmentation masks that represent each scene geometry category.

More precisely, suppose that an image j belongs to stage S, which is composed of N depth layers (partitions). Correspondingly, there will be N mask maps for stage S denoted by T. Hence, the mask map T_i for a specific stage partition i is obtained by taking the average of the mask maps, as in [10]:

$$T_i(x) = \frac{\sum_{j=1}^{n} M_{j,i}(x)}{n}, \tag{6}$$

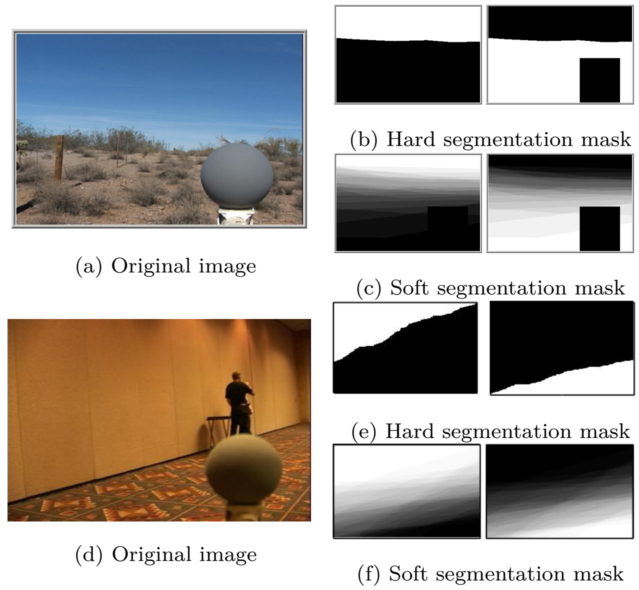
where n is the total number of images in the training data, and $M_{j,i}(x)$ is the mask map of the j^{th} image for i^{th} partition. Note that, $M_{j,i}(x)$ is an indicator function: $M_{j,i}(x) = 1$, if x belongs to the i^{th} partition and 0 otherwise, see [10] for details.

Hard Segmentation: Mask maps are used to automatically divide the images. Assuming that the images of stage S can be partitioned into N layers, then there exist N mask maps. The binary mask map is defined as follows:

$$T'_i(x) = \begin{cases} 1, & T_i(x) = \max_{j=1}^{N} T_j(x), \\ 0, & otherwise. \end{cases} \tag{7}$$

As a consequence, the values in the hard mask map are either 0 or 1, as shown in Figure 3(b). After the maximum mask map is obtained, the color of the light source is estimated using pixels from one layer, while, other pixels are ignored. In this way, multiple light sources are allowed per scene (i.e. one per depth layer).

Soft Segmentation: As stage classification is a rough estimation of the scene geometry, some locations (pixels) are more reliable than others to belong to a certain layer. To this end, we assign different confidence values to locations. We set the confidence values of pixels to $\mathrm{T_i}(\mathbf{x})$, which indicates the occurrence frequency of pixel positions appearing in the training data set. Hence, given a location (pixel position), the larger the confidence value, the more probable it belongs to that layer (image segment). Note that the values of the soft segmentation mask map are between 0 and 1, as shown in figure 3(c) of stage "*skyGnd*", and figure 3(f) of stage "*gndDiagBkgLR*".



(a) Original image (b) Hard segmentation mask (c) Soft segmentation mask (d) Original image (e) Hard segmentation mask (f) Soft segmentation mask

Fig. 3. An example of hard and soft segmentation mask maps. The original image (a) belongs to stage *"skyGnd"* and (d) belongs to stage *"gndDiagBkgLR"*. The mask maps are of the same size as the original image. The difference between hard segmentation mask map, shown in figures (b), (e), and the soft segmentation mask map, shown in figures (c), (f), is that values in figures (b), (e) are either 0 or 1. While values in figures (c), (f) are between 0 and 1.

After the mask map has been obtained by hard or soft segmentation, images in the training data set will be divided into several segments. The most suitable color constancy algorithm will be selected for each segment. This method takes advantage of *local* illumination in which the color of the light source is estimated using pixels from one layer while other pixels are ignored. In this way, multiple light sources are allowed per scene.

C. Illumination Estimation from Segmentation

The method using scene geometry for color constancy in [10] consists of the following steps: first, images are classified into stages. Then, according to the assigned stage model, images are divided into depth layers. Then, for each depth layer, the proper color constancy algorithm is selected. This is achieved by analyzing the angular errors for all color constancy algorithms applied on all images in the training data set of this stage. Hence, the stage model is labeled with depth layers and their corresponding color constancy methods, providing the highest color constancy accuracy (*i.e. lowest angular error*). Each depth layer is allowed to be illuminated by a different (single) light source. In the case of a single layer (i.e. the whole image), illumination estimation will be computed using the best color constancy algorithm for its assigned stage. The whole process is shown in Figure 4.

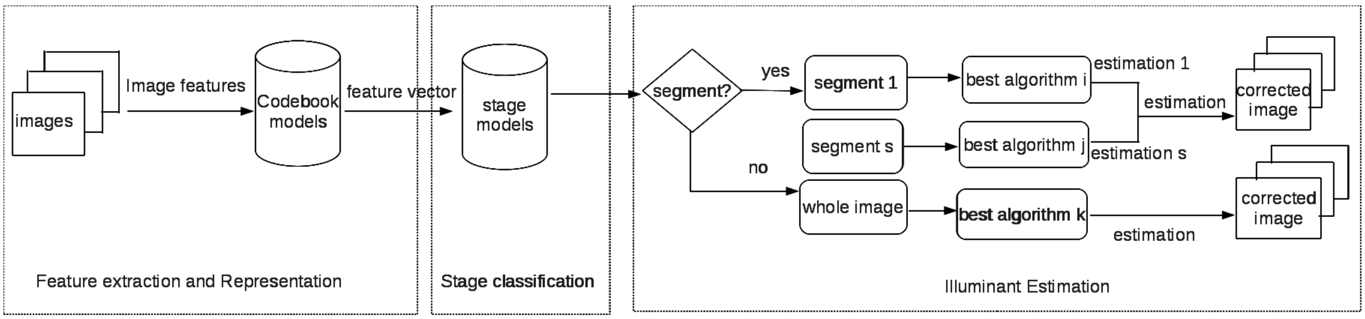


Fig. 4. Outline of color constancy using 3D scene geometry in [10]. Note that the codebook models and the stage models are obtained off-line. After the stage segmentation is obtained by hard or soft segmentation, the most suitable color constancy algorithm (with the lowest angular error) is selected for each layer of the stage. Stage segmentation and color constancy algorithm selection are trained on the data set beforehand.

For each unseen image in the test data set, the online process is as follows: first, the image is classified into stage S. Then, it is divided according to the mask map of stage S, which has been obtained by the training data set beforehand. Further, the color of the light source is estimated for each depth layer using its assigned color constancy algorithm. The final estimated illumination for the entire image is a weighted combination of the illuminations estimated for each depth layer. This weighting scheme provides the confidence degree between the estimated illumination (assigned based on the best performing color constancy algorithm) and the actual illumination. To be precise, the weight of the estimated illumination for each layer is inversely proportional to its angular error, see [10] for details.

D. Evaluation

In this section, we will first show experimental results for several state-of-the-art color constancy algorithms. Next, the results are compared to the performance for estimating the illumination by exploiting both the stages and the depth knowledge within the stages.

The algorithms that are evaluated here are the five instantiations discussed in Section III. The results for single algorithms are shown in Table II. These methods are applied to each image in *Real-World dataset*. Table II shows that edge-based methods (i.e. 1^{st}-order Grey-Edge and 2^{nd}-order Grey-Edge) outperform the pixel-based methods (i.e Grey-World, White-Patch, and general Grey-World).

Results of Illumination Estimation from Stages (Without Segmentation): Evaluation of the proposed methods is performed using the leave-one-out cross validation method. Illumination estimation is computed using the entire image. The median angular error of the proposed method without segmentation is 4.8° as shown in Table II. Compared with the best-performing algorithm, i.e., the 1^{st}-order Grey-Edge, a decrease of almost 8% on the median angular error is achieved. Results on individual stages reveal that most of the color constancy algorithms have a preference for specific stages. For instance, 0^{th}-order methods like the White-Patch and the general Grey-World prefer stages where depth is significant, like the stage *sky+background+ground*. Such stages with a large depth can contain haze, which causes a relatively low signal-to-noise ratio. It is known from [8] that methods that are based on higher-order statistics like the 2^{nd}-order Grey-Edge do not perform well on such images. On the other hand, the 2^{nd}-order Grey-Edge algorithm performs better on images with a high amount of texture, e.g. many edges. This is reflected in a preference for stages like *diagBkgLR* and *diagBkgRL* that generally contain images with much contrast and many edges.

Results of Illumination Estimation from Segmentation:

a) Hard Segmentation: The performance of the proposed method using hard segmentation on each stage is shown in Figure 5. The performance of the proposed method using hard segmentation on the entire data set *real-world data set* is given in Table II: the median angular error equals 4.5°. Compared to the baseline, the median angular error is reduced by almost 14%.

b) Soft Segmentation: The performance of the proposed method using soft segmentation on each stage is shown in Figure 5 while the result over the whole data set is shown in Table II: the median angular error is 4.6°, which is quite similar to the proposed method using hard segmentation. The proposed method using soft segmentation makes an improvement of 12% in median angular error over the baseline.

Discussion. Manual classification in Table II, shows how the stage classification influences the final results. Using this ideal classifier (i.e. the mean average precision is 1), the median angular errors of the method without segmentation is reduced to 4.6°. Using hard segmentation, the best-performance that can be obtained is 3.7° for the median angular error, while the median angular error can be further reduced to 3.6° by using soft segmentation. Hence, improving stage classification will further improve the color constancy results significantly.

Moreover, due to the stage classification, we do not only improve the overall illumination estimation accuracy, but we can also assess the illumination color of the various geometrical constellations of a scene. This is outlined in

Method	Mean		Median	
Grey-World	7.4°		7.0°	
White-Patch	7.3°		6.1°	
general Grey-World	6.4°		5.8°	
1^{st}-order Grey-Edge	**6.0°**		**5.2°**	
2^{nd}-order Grey-Edge	6.0°		5.4°	
Proposed (auto): without segmentation	5.7°	(− 5%)	4.8°	(− 8%)
Proposed (auto): hard segmentation	**5.4°**	**(−10%)**	**4.5°**	**(−14%)**
Proposed (auto): soft segmentation	5.4°	(−10%)	4.6°	(−12%)
Proposed (manual): without segmentation	5.5°	(− 8%)	4.6°	(−12%)
Proposed (manual): hard segmentation	4.7°	(−22%)	3.7°	(−29%)
Proposed (manual): soft segmentation	**4.7°**	**(−22%)**	**3.6°**	**(−31%)**

TABLE II

Performance of several color constancy algorithms on the *real-world data set*. Proposed (auto) means that the proposed methods are applied to automatically classified images, while proposed (manual) indicates that our methods are evaluated on manually classified images. Note that, results in bold indicate the best performing method within each category.

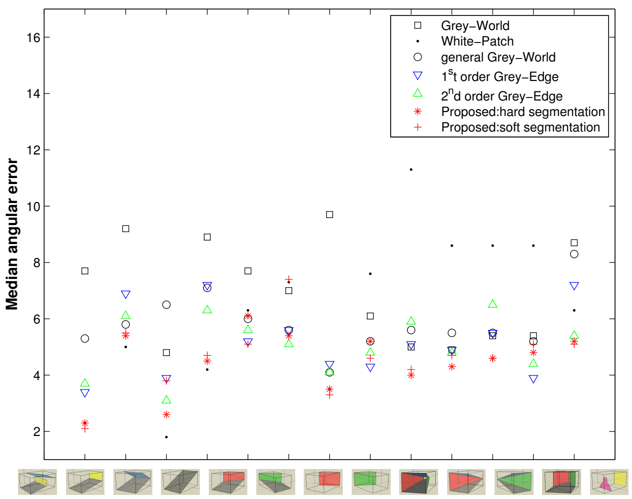


Fig. 5. Median angular errors of color constancy algorithms for each stage on the *real-world data set* from [10]. The stage models are shown on the horizontal axis: "sky+bkg+gnd", "bkgGnd", "skyGnd", "gnd", "gndDiagBkgLR", "gndDiagBkgRL", "diagBkgndLR", "diagBkgndRL", "box", "1sidewall-LR", "1side-wallRL", "corner", " persBkg".

Figure 5, where each stage is represented by its best estimation algorithm. Expanding from this, we propose to estimate the light source color at the various depth layers as indicated by the $3D$ stage model. This allows the estimation of a distant light source and to distinguish a nearby illumination (indoor, shadow) from a far away light source (outdoor, sunlight). In the next section, we propose to exploit scene depth (*i.e.* scene geometry) together with natural image statistics to achieve a proper selection of color constancy algorithms per depth.

E. Multiple Illumination

In this section, we explain the assumptions of our model for distinguishing nearby light sources (e.g. indoor illumination) from distant illuminations (e.g., outdoor illumination). Hence, the estimation of multiple light sources per scene geometry is achieved. For this purpose, a comparison between the estimated illumination per depth layer and its corresponding ground truth illumination, is to be evaluated.

Segmentation	Estimation	Groundtruth
sky (M)	(0.57, 0.58, 0.58)	
background (M)	(0.59, 0.58, 0.55)	(0.56, 0.58, 0.58)
ground (M)	((0.60, 0.58, 0.54)	

TABLE III

The mean of the Ground-truth Illumination (M) for stage *sky+bkgnd+gnd* along with the Euclidean distance between M and the standard deviation (std), Compared to those obtained using the illumination of the White patches within each specific segment.

However, such ground truth is not yet available in the current color constancy data sets. Consequently, we did not pursue evaluation of these extensions. Alternatively, in our experiments we use the available ground truth illumination for the whole image (*global illumination*) as a baseline to compare with the estimated illumination per depth.

Expanding from this, we propose an experiment based on the global image illumination and the well-known *White-Patch assumption*, which states that a surface with perfect reflectance properties will reflect the full range of light that it captures. Consequently, the color of this perfect reflectance is exactly the color of the light source. Accordingly, for stage S, we manually select the White-Patches for each of its constituent segments. Next, the mean *illumination of those manually selected patches* per segment, is then compared with the mean *ground truth illumination (available for each image)* of the given stage. Specific example for stage *sky+bkgnd+gnd* is shown in Table III.

Discussion: The quantitative results shown in table III, demonstrate that when the object is near the light source, its estimated illumination is closer to the illumination of the light source. However, when the object is far away from the light source then its estimated illumination is deviated from the light source illumination. For instance, for the *sky + background + ground* scene geometry category, which is an outdoor scene category, and, consequently, its light

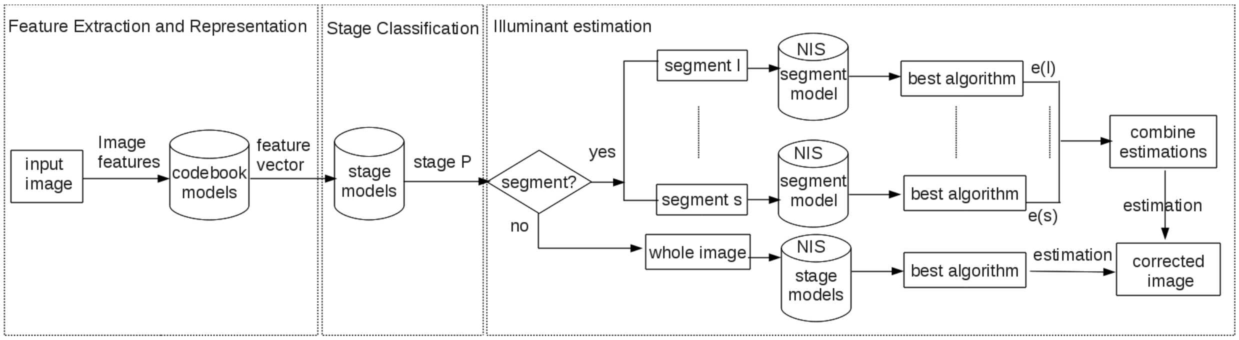


Fig. 6. Outline of the proposed method using $3D$ scene geometry, Natural Image Statistics features (NIS) and color constancy. First, an input image is classified into a stage, and then segmented into depth layers. Then, the method exploits the local image statistics per depth (image segment) for selecting the most appropriate color constancy algorithm. The final estimated illumination for the whole image is obtained by a weighted combination of the estimated illumination per depth. Note that, the NIS classifiers are trained (using cross-validation scheme) and tested on the *Real-World dataset*.

source illumination is the sunlight illumination: for the *sky* layer, which is the nearest to the light source, its estimated illumination is the closest to the ground-truth illumination. However, for the *ground* layer which is the farthest from the light source, it has the most deviated illumination estimation from the light source (ground-truth).

VI. Illumination Estimation using Segmentation and Natural Image Statistics

The proposed method falls into the learning-based color constancy category. An alternative approach is proposed in [17], that exploits the relation between natural image statistics and color constancy. The method is based on a *global selection mechanism*, where the Weibull distribution is used to select the proper color constancy method per image. In contrast, this paper proposes a novel strategy for selecting the most appropriate color constancy method per *image depth*, based on its *local natural image statistics* features.

The main distinction between this work and other learning based color constancy methods that estimate illumination for image regions, such as [11], is that they use training data to learn surfaces (regions) based on texture features. For a test image, they first segment it into regions. For each image region, similar surfaces are then found (in the training dataset), based on weak color features (i.e. comparing the statistics of pixels belonging to similar surfaces with the target surface). Further, they use the ground truth of corresponding surfaces for illumination estimation per segment. Meanwhile, we utilize spatial information based on the 3D scene geometry models and its constituent depths (segments) for the estimation of illumination per depth. We also exploit the Weibull features which are more efficient features (instead of the weak color features used in [11]) for estimating the illumination per segment. Finally, we studied and applied more complex methods for integrating the estimated illumination per depth (segment) into a unique illumination estimate.

To this end, the illumination estimation task is reduced to the following steps: (1) learning stage classifiers using the training images of the stages dataset; (2) Learning the Natural Image Statistics (NIS) classifiers [17] per *depth* from training images of the color constancy datasets; 3) Estimating the illumination for each segment of the target image based on its local natural image statistics and the pre-learned NIS model per depth; 4) Combining these weighted estimated illuminations per depth into a unique more accurate estimate. The whole process is shown in Figure 6.

A. Depth-NIS Model

We use the training images of each stage S to learn *Depth-NIS* models per stage. First, the training images of each stage S are partitioned into segments using hard or soft segmentation maps (described in section V-B). For each image k in the training set of S, the local image statistics features ω_{kj} belonging to segment j are extracted. Next, labels y_j of these stage images, belonging to segment j, are computed. More precisely, let M be the set of color constancy algorithms that are considered, where M_i denotes algorithm i. Further, the accuracy of the estimated illumination of algorithm i on layer j is denoted by $\epsilon_i(j)$. This is computed by analyzing the angular errors for all color constancy algorithms for depth layer j. The color constancy algorithm with the lowest angular error is assigned to this segment. Hence, label y_j is derived using the performance of the color constancy algorithms on segment j.

$$y_j = min_i(\epsilon_i(j)) \tag{8}$$

where i denotes the color constancy method and j is the depth layer (image segment) in stage S, respectively.

Finally, a *NIS classifier* is created for each depth layer $layer_j$. The labels y_j, determined in the previous step, are used as classes. For an unknown test image we first apply the stage classifier. After the stage assignment and segmentation steps, extract NIS features for image segment j. Then, apply $layer_j$ NIS trained classifier on these extracted features. The output of the classifier is a color constancy method which is assigned to image segment j

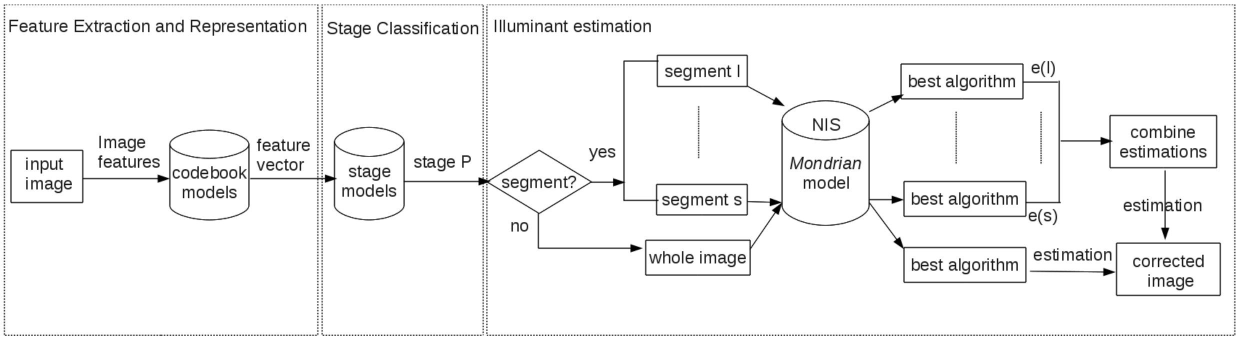


Fig. 7. Outline of color constancy using $3D$ scene geometry, Natural Image Statistics features (NIS) and color constancy. For a test image, stage classification and image segementation are done first. The selection of the proper color constancy method is done using NIS trained classifiers per stage depth. The method is trained on an independent dataset *Mondrian dataset* and the tested on the *Real-World dataset*.

for estimating its illumination. The final estimated illumination for a test image is a weighted combination of the estimated illuminations for each segment j belonging to its assigned stage.

For the purpose of evaluating our approach, the two different schemes proposed in [17], for training the required NIS classifiers are investigated. In the first scheme which is based on cross-validation we use the *real-world data set* for both training and evaluating our approach. The *real-world data set* will be divided into 15 parts, then the method will be trained on 14 parts of the data and tested on the remaining part. This procedure is repeated 15 times. Hence, every image exists exactly once in the test set. The whole process is shown in Figure 6. The second scheme consists of using an *independent data set* which we refer to as *Mondrian data set* for training the NIS classifiers. Meanwhile, the *real-world data set* will be used for the testing phase. In this way, the data sets for training and testing are completely different. This scenario reflects the case, when the data set (used for testing the method) is unknown which is the most general case. The whole process is shown in Figure 7.

B. Weighted illumination Estimation per Depth

In the previous section, we provided an efficient method for estimating the illumination per image segment S based on its local statistics. Here, we analyze how to optimally fuse the locally estimated illumination to generate a new illumination estimate. The simplest method is to take the average of the illumination estimates for each segment S. However, a straight forward extension is to take the weighted average of all these estimated illuminations. Hence, if n color constancy algorithms corresponding to the n segments of an image are to be combined, then the weighted average is defined as:

$$\bar{\mathrm{e}} = \sum_i^n w_i e_i,$$

where, i refers to an image segment, e_i is the estimated illumination for segment i and w_i is the learned weight assigned for the estimated illumination e_i.

In particular, the weighting of the estimated illumination for an image segment S is assigned based on the stage category it belongs to. Concretely, the ground-truth illuminations (of the training images) belonging to each stage, are used to learn a *Stage Illuminant* model. In this section, we analyze four approaches to model the ground-truth illumination per stage, and further to combine multiple illuminations, namely *Bayesian*, *Kernel density*, *Histogram smooth*, *Mixture of Gaussians (MOG)* and *Error Difference Confidence (EDC)* schemes. To be precise, the different approaches are as follows.

Classifiers: For the purpose of learning the *stage-illuminant models* used for weighting the estimated illumination per depth, well-known classifiers such as *Bayesian* and *MOG* are investigated. For each classifier, we learn its underlying parameters using the training data of each stage. Finally, trained classifiers (i.e., Bayes & MOG) are obtained per stage. For an unknown test image, after the stage classification and segmentation steps, the estimated illumination for each image segment is assigned. To this end, the learned *stage-illuminant* classifiers is then used to weight the locally estimated illumination. For instance, we use the *MOG stage-illuminant* classifier, to derive the weighting of the estimated illumination for each image segment. This implies using the underlying parameters (mean and variance of k Gaussian distributions learned beforehand) of the *MOG* model.

Histograms: This approach exploits frequency histograms generated from the ground-truth illuminations of the training images within each stage. These frequency histograms (i.e., grids) measure the occurrence frequency of both the normalized *red and green* values of the ground-truth illuminations for each stage. Concretely, the output is a stage frequency grid which reveals how frequent an input illumination occur for a certain stage. To this end, we introduce the *Kernel density* and the *Histogram smooth* methods which exploit the occurrence frequency of the ground-truth illuminations for weighting the estimated illumination. In *Kernel density*, the generated histograms usually contain large amount of empty bins, which leads to noisy weighting estimates. In contrast, in the *Histogram smooth* scheme, the obtained histograms are smoothed to reduce the noise effect due to the existence of the empty bins.

The testing procedure for an input test image is as follows: first, the image is classified to the correct stage, and further segmented. Then, the illumination is estimated and weighted for each segment. The weight of each estimated illumination is assigned by first extracting its normalized red and green illuminant values. These values are then mapped within the appropriate frequency grid (learned beforehand) for its assigned stage. Finally, the output weight is the corresponding occurrence value for the closest ground-truth illumination to the given input illumination.

EDC Criteria: Finally, we examine the weighting approach used in [17] for weighting the estimated illumination per segment. The angular error E between the estimated illumination for segment S and the image ground-truth illumination is exploited. In particular, weighting the estimated illumination of each segment is assigned with respect to the inverse of its angular error. Hence, the larger the value of the angular error E for the estimated illumination, the smaller its assigned weighting value. We refer to this scheme as *Error Difference Confidence (EDC)* criteria.

For the purpose of evaluating an unknown test image, the process works as follows: After classification and segmentation, the local image statistic features for each segment S are extracted. Then, a trained NIS-Classifier for segment S (learned beforehand) is used with these features to estimate its illumination. The angular error E between the estimated illumination of segment S and the ground truth illumination of the input image is obtained. The inverse of the angular error E is used for weighting the estimated illumination of segment S.

C. Color Correction

The final step, is to combine all these weighted estimates of each segment into a single estimated illumination. The single estimated illumination is then compared with the ground-truth illumination for the input image. To be precise, we first sum up all the weighted estimated illuminations into a single estimated illumination sum_{we}. Followed by, normalizing the single illumination sum_{we} by the total weights used (i.e., summation of the inverse of the angular errors for each segment). Finally, we calculate the angular error between the single estimated illumination and ground-truth illumination of the input image.

VII. Experiments

In this section, the proposed method is evaluated and compared with the state-of-the-art color constancy methods on data sets including hyper-spectral, ColorChecker, linear and non-linear real-world data sets. The main advantage of hyper-spectral data is that many different illuminants can be used to realistically render the same scene under various light sources. However, the simulation of illuminants generally does not include real-world effects like inter-reflections and non-uniformity. Consequently, the evaluation of *real-world* RGB-images and *ColorChecker* results in more realistic performance evaluations. Next, the performance of the proposed method for estimating image illuminant based on its assigned scene geometry, the local image statistics per depth together with the proposed fusion algorithms are given.

TABLE IV

Performance of color constancy algorithms applied on hard and soft segmented images, trained and tested over the *real-world data set* using cross-validation *(cv)*. The proposed methods are applied to automatically classified images. *NIS*, *GE*, *EDC* and *KD* refer to *Natural Image Statistics*, *Grey-Edge* algorithms, *Angular Difference Confidence* and *Kernel Density* weighting schemes (see text).

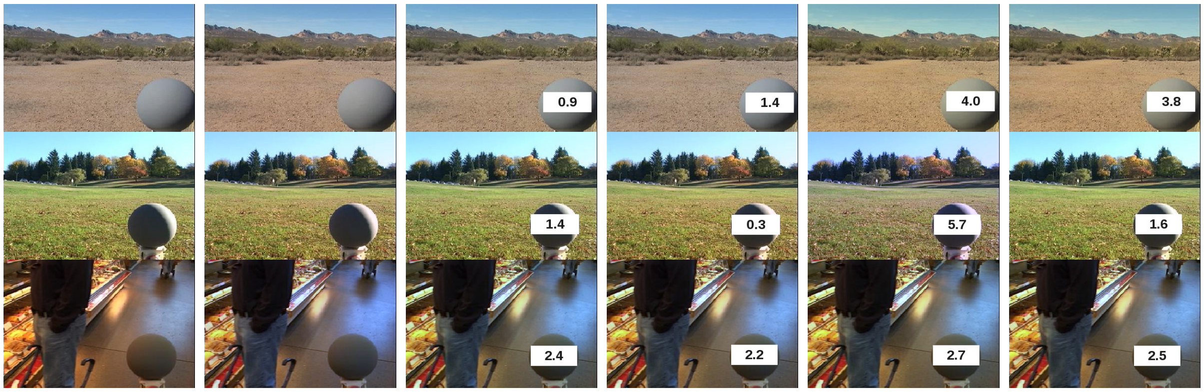
Method	**Mean**	**Median**
Baseline: 1^{st}-order GE	6.0°	5.2°
Baseline: Global NIS	5.7° (−5%)	4.7° (−10%)
indoor-outdoor classification	7.0° (+17%)	6.5° (+25%)
NIS hard-cv-Average	5.4° (−10%)	4.6° (−12%)
NIS hard-cv-Bayesian	4.9° (−18%)	3.7° (−29%)
NIS hard-cv-KD	4.7° (−22%)	3.7° (−29%)
NIS hard-cv-HistSmooth	4.7° (−22%)	3.6° (−31%)
NIS hard-cv-MOG	4.7° (−22%)	3.4° (−35%)
NIS hard-cv-EDC	**3.9**° (**−35**%)	**2.8**° (**−46**%)
NIS hard-Mondrian	4.7° (−22%)	3.3° (−37%)
NIS soft-cv-Average	5.3° (−12%)	4.6° (−12%)
NIS softcv-Bayesian	4.6° (−23%)	3.5° (−33%)
NIS softcv-KD	4.5° (−25%)	3.3° (−37%)
NIS softcv-HistSmooth	4.5° (−25%)	3.3° (−37%)
NIS softcv-MOG	4.5° (−25%)	3.2° (−38%)
NIS soft-cv-EDC	**3.9**° (−35%)	**2.6**° (−50%)
NIS soft-cv-Mondrian	5.4° (−10%)	4.4° (−15%)

A. *Illumination Estimation using NIS-real-world*

In this experiment, the performance of the proposed scheme for training our approach using part of the real-world data, and testing it using another independent part is examined. Hence, no direct relation exists, between the training data and the testing data (see Figure 6). This scheme corresponds to the situation where the circumstances under which the system is used are known apriori. The method is evaluated based on both hard and soft segmentations together with the different weighting strategies discussed in section VI-B for fusing the illumination estimates obtained for each image segment.

Hard Segmentation & NIS-*real-world:* Table IV shows the performance of the proposed method using hard segmentation (denoted as NIS-hard-*cv*) and a simple average *(denoted as "Average")* of the estimated illuminations. The obtained results show that simple averaging of the output illuminations improves the results compared to the baseline algorithms. Compared to the *global Natural Image Statistics (NIS)*, the median angular error is reduced by almost 5%. While, a reduction of 12% in the median angular error is obtained compared to $1^{st}-order$ *Grey Edge* baseline.

In addition to the use of only single algorithms, two combination algorithms are evaluated. The first method is proposed by [7] and distinguishes between indoor and



(a) Original image (b) Ground truth (c) Correction using NIS+hard segmentation (d) Correction using NIS+soft segmentation (e) Correction using 1^{st}-order Grey-edge (f) Correction using global-NIS

Fig. 8. Results of color constancy. The angular error is given on the grey ball, which is masked during illumination estimation.

outdoor images. They propose to use the shades-of-grey method for indoor images and the 2^{nd}-order Grey-Edge method for outdoor images. For convenience, we used manual annotation of indoor and outdoor images instead of the indoor-outdoor classifier proposed in [7]. As can be seen in Table IV, the accuracy of the illumination estimates does not improve with respect to the single baseline color constancy algorithm.

Another combination method is proposed by [8], where they use global image statistics for the selection of the most appropriate color constancy algorithm. Results indicate that the performance indeed improves with respect to the best-performing single algorithm, see table IV. However, the scene geometry is not taken into account as this method uses global selection of the most appropriate color constancy algorithm.

Next, we evaluate the performance of the *weighting schemes* proposed for combining the estimated illuminations, namely, Bayesian, Mixture-of-Gaussians *"MOG"*, Kernel-Density *"KD"*, Histogram-Smooth *"HistSmooth"*, and Error Difference confidence *"EDC"*, respectively. Using a weighted average instead of a simple average performs significantly better than the baseline. The median error decreased by around 29% based on the *Bayesian* weighting scheme with respect to the baseline. Compared to the *Bayesian* method, the *Kernel-density* decreased the mean error from 4.9° to 4.7°, while maintaining the same median error 3.7°. In addition, the use of the *Histogram-smooth* method results in a reduction of the median error to 3.6° due to the reduction of the noise level obtained, while smoothing the histogram empty bins. Further, the proposed method based on *MOG* scheme decreased the median error to 3.4° using two Gaussians. Finally, the *EDC* weighting scheme leads to a major drop on both the median and the mean angular errors up to 2.8° and 3.9°, respectively. These obtained results excel the state-of-art results significantly; a reduction of 35% in the mean angular error and 46% in the median error is obtained w.r.t. the baseline (see table IV).

When using a *perfect classifier* (i.e., all of the test images are correctly classified into stages), the performance of the proposed method achieve an improvement of 48% of median error compared to the best performing single color constancy algorithm.

Soft Segmentation & NIS-real-world: The performance of the proposed method using soft segmentation over the *real-world data set* is shown in Table IV. The proposed method based on the simple average of estimated illuminations (denoted as NIS-soft-cv-average), results in a performance improvement of almost 12% compared to the baseline. Results obtained using the weighted average of the estimated illuminations improve the performance over the simple averaging combination scheme. The *Bayesian* method decreased the median error to 3.5° compared to 4.6° using the simple averaging scheme. Additionally, the median errors obtained using the *MOG*, *kernel density*, and *Histogram smooth* weighting methods are reduced further over the *Bayesian* up to 3.3°, 3.3°, and 3.2°, respectively. Finally, the *EDC* method leads to a major reduction in the median angular error up to 2.6°. This results in an overall performance improvement of almost 50% with respect to the baseline (i.e., 5.2°).

When using a *perfect classifier* (i.e., all of the test images are correctly classified into stages), the performance of the proposed method achieve an improvement of 52% of median error compared to the best performing single color constancy algorithm.

B. Illumination Estimation using NIS-Mondrian

In this experiment, the performance of the second approach proposed for training our method is evaluated. In particular, the training is done using an independent dataset (*Mondrian* dataset), and tested on the real-world dataset. This scheme corresponds to the most generic approach of the proposed method, where the testing is unknown. The method is evaluated based on both hard and soft segmentations and the different weighting strategies for fusing the estimated illumination per segment.

Hard Segmentation & NIS-Mondrian: Results of the second experiment, learned on the Mondrian data set and tested on the real-world data set based on the hard segmentation masks (denoted by NIS hard-Mondrian in table IV) performs significantly better than the baseline. Compared to the *Global NIS* baseline, the median angular error is reduced by around 30%. Meanwhile, a 37% reduction on the median error is obtained (i.e., from 5.2° up to 3.3°) compared to the $1^{st}-$*order Grey Edge* baseline.

Soft Segmentation & NIS-Mondrian: The proposed algorithm learned on the Mondrian data set and tested on the *real-world data set* based on the soft segmentation masks (denoted by NIS soft-Mondrian in table IV) performs significantly better than the baseline. An improvement on the median angular error from 5.2° up to 4.4° is obtained (i.e., 15% improvement).

In addition to automatic classification, *manual classification* is used to determine how the stage classification performance influences the final results. Using this *ideal classifier*, the best-performance that can be obtained is 3.3° (i.e., 37% improvement) for the median angular error based on soft segmentation, while the median angular error can be further reduced to 3.2° (i.e., 38% improvement) by using hard segmentation.

Discussion: In the *Mondrian* experiment: the test-data is unknown, and corresponds to the most generic approach of the proposed method. No learning step is required for a new data set, since the results of the classifier are independent of the test data. On the other hand, the proposed algorithm learned for the *real-world data set*, requires data with enough variety to train the method. In this experiment, test-data is known making the proposed method less generic. Hence, a learning step is required for each new data set.

Results show significant improvement of the *real-world data set* case over the *Mondrian* case (with no prior knowledge): the median angular error drops to 2.8° and 2.6° based on hard and soft segmentations, respectively. However, from the experiments, it can be concluded that the proposed method can be trained using completely independent training set, and still performs significantly better than the baseline algorithm (i.e., 5.2°).

Figure 8 presents three images, which are correctly classified based on our proposed approach. The proposed method using soft segmentation is more effective in the presence of shadow or shading edges.

C. Linear Real-World and ColorChecker Datasets Evaluation

Finally, we evaluate our method on the linear *real-world* and the *ColorChecker* datasets. Note that the training is done using a) a portion of the datasets used for evaluation using cross-validation, and b) an independent *Mondrian* dataset. The results in table V and table VI show the obtained results based on our proposed method for the *real-world* and the *ColorChecker* datasets, respectively. As expected, the various combination schemes applied on the estimated illumination of each segment improve the results over the simple averaging combination scheme. In addition, the *EDC* method results in achieving the minimum angular error between the estimated illumination and the ground-truth illumination (i.e., the best performance). Finally, in table VI we compare with the method proposed in [11]. The proposed method resulted in improving the performance. We attribute this to the important spatial knowledge that we exploit based on the proposed 3D scene geometry, together with the efficient natural image statistics features for estimating illumination per depth and integrating these weighted estimated illuminations efficiently into a unique single illumination. These are all important for obtaining proper performance.

TABLE V

PERFORMANCE OF COLOR CONSTANCY ALGORITHMS APPLIED ON *hard* AND *soft* SEGMENTED IMAGES, TRAINED OVER LINEAR *real-world data set* USING CROSS-VALIDATION *(cv)* OR USING THE INDEPENDENT MONDRIAN DATA SET. THE PROPOSED METHODS ARE APPLIED TO AUTOMATICALLY CLASSIFIED IMAGES. *GE*, *EDC* AND *KD* REFER TO *Grey-Edge* ALGORITHM, *Angular Difference Confidence* AND *Kernel Density* WEIGHTING SCHEMES, RESPECTIVELY (SEE TEXT).

Method	**Mean**	**Median**
Baseline: 1^{st}order GE	14.5°	13.0°
Baseline: Global NIS	12.6° (−13%)	12.2° (−6%)
NIS hard-cv Average	11.4° (−21%)	9.5° (−27%)
NIS hard-cv-Bayesian	11.1° (−23%)	9.4° (−28%)
NIS hard-cv-KD	11.1° (−23%)	9.4° (−28%)
NIS hard-cv-HistSmooth	10.9° (−25%)	9.4° (−28%)
NIS hard-cv-MOG	10.9° (−25%)	9.0° (−31%)
NIS hard-cv-EDC	**10.6°** (−27%)	**9.2°** (−29%)
NIS hard-Mondrian	10.8° (−26%)	8.7° (−33%)
NIS soft-cv-Average	11.4° (−21%)	10.0° (−23%)
NIS soft-cv-Bayesian	11.0° (−24%)	9.5° (−27%)
NIS soft-cv-KD	11.0° (−24%)	9.5° (−27%)
NIS soft-cv-HistSmooth	11.0° (−24%)	9.5° (−27%)
NIS soft-cv-MOG	11.0° (−24%)	9.4° (−28%)
NIS soft-cv-EDC	**10.6°** (−27%)	**9.4°** (−28%)
NIS soft-Mondrian	11.0° (−24%)	8.6° (−34%)

VIII. CONCLUSION

We have investigated the relation between scene depth, local image statistics and color constancy. The scene geometry has been computed first to obtain image depth layers. Then, image statistics are exploited (per depth) to select the proper color constancy method. Our approach enables the estimation of multiple illuminations by distinguishing nearby light source from distant illuminations.

Experiments on various benchmark data sets show that the proposed algorithm outperforms state-of-the-art single color constancy algorithms with an improvement of almost 50% of median angular error. When using a perfect classifier (i.e, all of the test images are correctly classified into stages), the performance of the proposed method improves the median angular error as much as 52%. Using Linear data set, leads to an improvement of 29% compared to the state-of-the-art single color constancy algorithms. This gain in performance can largely be explained by the

TABLE VII
SUMMARY OF THE EXPERIMENTS. NOTE THAT: NIS REFERS TO NATURAL IMAGE STATISTICS.

Experiment	Results	Train set	Test set
- Illumination Estimation from stages vs segmentation. - Besides comparisons with several standard state-of-the-art algorithms.	- Table II.	- Stages dataset. - Real-World dataset.	- Real-World dataset.
- Illumination Estimation using Segmentation (hard & soft) & NIS. - Weighted illumination Estimation per Depth	- Table IV.	- a) Real-World dataset. - b) Mondrian dataset.	- Real-World dataset.
- Illumination Estimation using Segmentation (hard & soft) & NIS on Linear datasets.	- Table V & VI.	- a) Real-World dataset. - b) Color Checker dataset. - c) Mondrian dataset.	- a) Real-World dataset. - b) Color Checker dataset.

TABLE VI
PERFORMANCE OF COLOR CONSTANCY ALGORITHMS APPLIED ON *hard* AND *soft* SEGMENTED IMAGES, TRAINED OVER LINEAR *Color-Checker data set* USING CROSS-VALIDATION *(cv)* OR USING THE INDEPENDENT MONDRIAN DATA SET. *GE* REFERS TO *Grey-Edge* ALGORITHM. *EDC* AND *KD* REFER TO *Error Difference Confidence* AND *Kernel Density* WEIGHTING SCHEMES, RESPECTIVELY (SEE TEXT).

Method	Mean	Median
Baseline: Grey World	6.4°	6.3°
Baseline: White Patch	7.6°	5.7°
Baseline: General Grey World	4.7°	3.5°
Baseline: 1^{st}order GE	5.3°	4.5°
Baseline: 2^{nd}order GE	5.1°	4.4°
Baseline: Global NIS	4.2°	3.1°
Baseline: Exemplar-based	3.1°	2.3°
NIS hard-cv Average	3.4°	2.6°
NIS hard-cv-Bayesian	3.1°	2.4°
NIS hard-cv-KD	3.1°	2.4°
NIS hard-cv-HistSmooth	2.9°	2.4°
NIS hard-cv-MOG	2.9°	2.2°
NIS hard-cv-EDC	**3.0°**	**2.1°**
NIS hard-Mondrian	3.8°	2.9°
NIS soft-cv-Average	3.4°	2.5°
NIS soft-cv-Bayesian	3.0°	2.4°
NIS soft-cv-KD	3.0°	2.3°
NIS soft-cv-HistSmooth	2.9°	2.3°
NIS soft-cv-MOG	2.9°	2.2°
NIS soft-cv-EDC	**2.8°**	**2.2°**
NIS soft-Mondrian	3.6°	2.8°

fact that most color constancy algorithms are specifically suited for images with certain image statistics, like a high (or low) signal-to-noise ratio. Further, it is shown that extracting local geometry features is more efficient than applying a global selection or combination algorithm. A summary of the discussed methods is presented in Table VII.

ACKNOWLODEGMENTS

This work was partly supported by the projects TIN2009- 14501- C02- 02 and TIN2012- 39051 of the Spanish Ministry of Science and Innovation.

REFERENCES

[1] M. Ebner., "Color constancy." in *Wiley*, 2007.
[2] G. Hordley., "Scene illuminant estimation:past, present, and future." *Color Res. and App.*, vol. 31, no. 4, pp. 303–314, 2006.
[3] A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: Overview and experiments." *IEEE Trans. on Image Processing (TIP).*, vol. 20, no. 9, pp. 2475 – 2489, 2011.
[4] A. Chakrabarti, K. Hirakawa, and T. Zickler., "Color constancy beyond bags of pixels." in *CVPR*, 2008.
[5] G. Finlayson, S. Hordley, and C. Lu., "On the removal of shadows from images." *TPAMI*, vol. 28, no. 1, pp. 59–68, 2006.
[6] J. van de Weijer, T. Gevers, and A. Gijsenij., "Edge-based color constancy." *TIP*, vol. 16, no. 9, pp. 2207–2214, 2007.
[7] S. Bianco, G.Ciocca, C. Cusano, and R. Schettini., "Improving color constancy using indoor-outdoor image classification." *TIP*, vol. 17, no. 12, pp. 2381–92, 2008.
[8] A. Gijsenij and T. Gevers., "Color constancy using natural image statistics." in *CVPR*, 2007.
[9] R. Lu, A. Gijsenij, T. Gevers, K. van de Sande, J.M., Geusebroek, and D. Xu., "Color constancy using stage classification." in *ICIP*, 2009.
[10] R. Lu, A. Gijsenij, D. Xu, V. Nedovic, J.M., Geusebroek, and T. Gevers., "Color constancy using 3d scene geometry." in *ICCV*, 2009.
[11] H. Jose and M. Drew., "Exemplar-based colour constancy and multiple illumination." *PAMI.*, 2013.
[12] V. Nedovic, A. Smeulders, A. Redert, and J.-M. Geusebroek., "Depth information by stage classification." in *ICCV*, 2007.
[13] R. Henry, S. Mahadev, S. Urquijo, and D. Chitwood., "Color perception through atmospheric haze." *J. Opt. Soc. Am. A*, vol. 17, no. 5, pp. 831–835, 2000.
[14] A. Torralba and A. Oliva., "Depth estimation from image structure." *TPAMI*, vol. 24, no. 9, pp. 1226–1238, 2002.
[15] A. R. Vladimir Nedovic, Arnold W. M. Smeulders and J.-M. Geusebroek., "Stages as models of scene geometry." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1673–1687, 2010.
[16] J. Geusebroek and A. Smeulders., "A six-stimulus theory for stochastic texture." *IJCV.*, vol. 62, no. 1-2, pp. 7–16, 2005.
[17] 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.
[18] E. Delage, H. Lee, and A. Y. Ng., "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image." in *CVPR*, 2006, pp. 2418–2428.
[19] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky., "Depth from familiar objects: A hierarchical model for 3d scenes." in *CVPR*, 2006., pp. 2410–2417.
[20] D. Hoiem, A. A. Efros, , and M. Hebert., "Geometric context from a single image." in *ICCV*, 2005, pp. 654–661.
[21] F. Ciurea and B. Funt., "A large image database for color constancy research." in *CIC*, 2003, pp. 160–164.
[22] A. Gijsenij, T. Gevers, and J. van de Weijer., "Computational color constancy: Survery and experiments." *Trans. on Im. Proc.*, vol. 20, pp. 2475–2489, 2012.
[23] L. Shi and B. Funt.
[24] K. Barnard, L. Martin, B. V. Funt, and A. Coath., "A data set for color research." *Color Research & Application.*, vol. 27, no. 3, pp. 147–151, 2002.
[25] K. van de Sande, T. Gevers, and C. Snoek., "Evaluation of color descriptors for object and scene recognition." in *CVPR*, 2008.
[26] D. G. Lowe., "Distinctive image features from scale-invariant keypoints." *IJCV*, vol. 60, pp. 91–110, 2004.
[27] S. Lazebnik, C. Schmid, and J. Ponce., "Beyond bags of features: Spatial pyramid matching for recognizing natural scene

categories." in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[28] N. Elfiky, F. S. Khan, J. van de Weijer, and J. Gonzaležz., "Discriminative compact pyramids for object and scene recognition." *Pattern Recognition.*, vol. 45, no. 4, pp. 1627–1636, 2012.

[29] C.-C. Chang and C.-J. Lin., "Libsvm: a library for support vector machines." 2001.